Disciplined Agile Scrum Master

Study Guide

Disciplined Agile

Disciplined Agile Scrum Master (DASM) Study Guide

About the DASM Training

Is your team treading water using waterfall? Do you feel trapped in an agile framework? Would you like to find solutions to the problems you've been wrestling with? Are you looking for ways to enhance your team's agility?

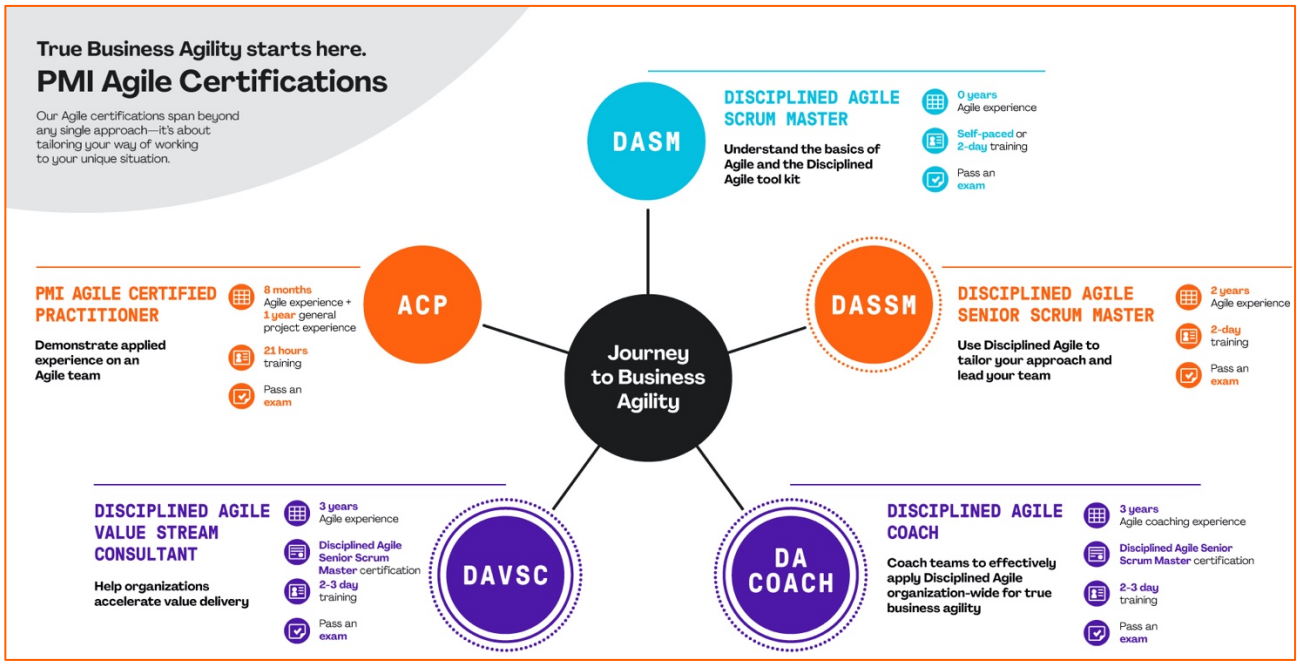
Break free from your old ways by choosing a way of working that fits your team's context. Find strategies to improve your processes and strengthen your team with the Disciplined Agile tool kit.

Disciplined Agile Scrum Master is a nine-lesson, instructor-led course that shows you how to use Disciplined Agile (DA) to improve your team's way of working. In just two days, you'll become familiar with foundational agile and lean practices that DA supports, practice using the tool kit to solve problems, and learn how to build high-performance teams.

This course is also appropriate for teams that wish to work together to learn Disciplined Agile and customize their way of working.

Filled with activities, animations, supplemental reading, and more, this course will prepare you to take the Disciplined Agile Scrum Master (DASM) exam and, equally important, start using Disciplined Agile immediately.

The PMI Agile Certification Journey



Materials for this Course

- *Choose Your WoW! A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working (WoW) (PDF Version)*
- DASM Activity Workbook PDF
- DASM Participant Handout PDF
- DASM Study Guide PDF
- DASM Bibliography/References/Glossary PDF

About this Study Guide

- This Study Guide is designed to aid you in reviewing the contents of the DASM training and preparing for the DASM Certification exam.
- This study guide does NOT contain everything you need to know to pass the exam.
- To adequately prepare, read this guide, along with relevant chapters of *Choose Your WoW!* and the *Participant Handout PDF*.
- **Use the Objectives, listed under each lesson in this guide, to determine what you need to study. The exam measures your ability to satisfy each of these objectives. While we encourage you to learn more about DA, you do not have to worry about information not contained in the objectives to pass the exam.**

Contents

About the Disciplined Agile Scrum Master (DASM)

Who Is a DASM?

A DASM is simply a DA Team Lead that practices the Scrum methodology.

The team lead plays an important role interacting with and serving the team. They also interact with stakeholders and neighboring teams, which means DA Scrum masters need superior people skills to be effective.

Throughout most of this training, references to the DA term "Team Lead" and the course term "DASM" are interchangeable.

What Is a DASM?

The responsibilities of a DASM fall below a DA senior Scrum master (DASSM) and a functional manager. A DASM area of focus is with a team.

Functional Manager	DASSM	
Responsible for a function (i.e., marketing or finance)	Responsible for implementation or development team	
Makes high-level business decisions	Guides their team to make joint decisions	**DASM**
	Qualified to work in variety of situations, with more complex scaling factors and variations	Qualified to work in more straightforward, less complex situations
	Needs skills in planning, metrics, and reporting	Has basic team leadership skills. Does not take an active role in planning, metrics, reporting or team development
	Takes an active role in team development	
	Knows the organization's various layers and departments	Knows how to work at the team level
	Can identify allies and coordinate with other teams to improve the value stream	
	Removes barriers between stakeholders and teams	

How Does a DASM Serve?

Team

First of all, a DASM serves the team, helping them be their best:

- Coaches team members in self-management
- Helps the team focus on creating high-value increments that meet the definition of done.
- Helps remove impediments to the team's progress
- Ensures that necessary events take place and are:
    - Positive
    - Productive
    - Within timebox

Product Owner

A DASM also serves the product owner in these ways:

- Helps find techniques for effective product goal definition and backlog management
- Supports in backlog grooming and planning
- Keeps informed of project status
- Helps the team understand the need for clear and concise backlog items
- Facilitates stakeholder collaboration as requested or needed

The Organization

Finally, a DASM serves the organization in these ways:

- Supports the product owner and team in achieving customer satisfaction
- Helps the team identify and address risks
- Helps with training and coaching in agile adoption
- Helps employees and stakeholders understand and work in their environment
- Removes barriers between stakeholders and teams.

Lesson 1: All About Agile

Description

In this lesson, you'll learn agile concepts and how to work with agile as a Disciplined Agile Scrum Master. After all, Scrum is an agile methodology. Functionally, a DASM coordinated and facilitates agile ceremonies or critical team events involved in developing a solution. A DASM also helps improve their processes by implementing Disciplined Agile, which is based on agile and lean.

Objectives

Describe the foundations of Agile

- Compare and contrast agile and waterfall
- List the benefits of being agile
- Outline the agile iterative way of working
- List and define the artifacts and ceremonies of agile

Describe Agile techniques and ceremonies relevant to Inception

- Define user stories
- Describe how to write and estimate a user story using different techniques
- Identify acceptance criteria and the definition of done
- Indicate how to effectively plan iterations

Describe Agile techniques and ceremonies that take place during Construction

- Describe how to demonstrate an iteration
- State how to obtain and receive feedback

Agenda

1. What is Agile?
2. The Agile Manifesto
3. How Does Agile Work
    a. The Iterative Process
    b. Planning an Iteration
    c. Agile Ceremonies and Artifacts
    d. User Stories
    e. Iteration Demonstration
4. Information Radiators

Lesson Notes

What Is Agile?

Agile is an iterative approach to project management and software development that helps teams deliver value faster and with fewer headaches.

Instead of betting everything on a big launch, agile teams deliver work in small, consumable increments.

There are several widely used agile methodologies, including Scrum, Extreme Programming, and the Dynamic Systems Development Method.

The Agile Manifesto

In February 2001, 17 people met at the Snowbird Ski Resort in the mountains above Salt Lake City, Utah, USA, to talk, ski, relax and try to find common ground—and of course, eat.

The problem, they agreed, was the document-driven, heavyweight approach most companies used for software development. These companies were so focused on planning and documentation that they lost sight of what really matters—pleasing customers.

What emerged was the Agile Manifesto, a 68-word document that ushered in a revolution—and not just in software development.

The Agile Manifesto lists four core values. It reads, in its entirety:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The Principles Behind the Agile Manifesto

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

There Is No Standard for Agile Terminology

Disciplined Agile strives to be agnostic in its terminology.

There will never be a standardization of terminology.

Feel free to choose whatever terms you like when you're on your teams but recognize that doing so may might hamper your ability to think outside the methodology box.

DA Term	XP Term	Scrum Term	Spotify Team
Iteration	Iteration	Sprint	Sprint
Team Lead	Coach	ScrumMaster*	Agile Coach
Daily Coordination Meeting	Daily Meeting	(Daily) Scrum Meeting	Huddle
Retrospective	Retrospective	Sprint Retrospective	Retrospective
Team	Team	Team	Squad, Tribe
Architecture Owner	Coach*		
Domain Expert	Customer*	Customer*	Customer*

* Means "not an exact match"

Agile Breaks Project into Iterations

At its core, agile is a way to get organized and plow your way through a complex project.

- **Make a list.** Sit down with your customer and make a list of features they'd like to see. This becomes your to-do list for the project.
- **Size things up.** Size up your tasks, relative to each other, and come up with a guess as to how long each one will take.
- **Set some priorities.** Ask your customer to prioritize their list so you get the most important stuff done first.
- **Start working.** Start at the top of your list and start delivering value, building, iterating, and getting feedback from your customer as you go.

This is oversimplified, of course, but it should provide a basic understanding. In the real world, the iterative process is a bit more complex.

Planning an Iteration

How do teams plan iterations?

- The product backlog is where the team collects all the work flowing to the team. Work is continuously placed in the product backlog from minimum business increments and from the release roadmap.
- The product owner prioritizes the product backlog, signaling to the team what work is the most important.
- At the start of each iteration, the team pulls the work they plan to do from the product backlog into the iteration backlog. This is a key point: the *team* pulls the amount of work it thinks it can get done during the iteration—*not* the product owner, team manager or stakeholders.
- Iteration planning ends when the team determines they have moved the right amount of work—that is, when they've pulled an amount that matches their capacity for the iteration). At this point, they start the iteration.

Agile Ceremonies

One of the things that sets agile apart from other management approaches is its ceremonies—sometimes referred to as rituals. These provide the framework for team to get work done in a structured manner, help to set expectations, empower the team to collaborate effectively, and ultimately drive results.

- **Iteration planning** is the activity to prioritize and identify the tasks for the next iteration.
- The **coordination meeting** is a regular, short meeting of the team where status is exchanged, progress is observed and impediments are noted and removed. This meeting is also known as a standup meeting or scrum.
- The **iteration demonstration** showcases what the team accomplished in the iteration.
- The **iteration retrospective** is a structured reflection designed to let the team learn and improve based on what's already been done.

Agile Ceremony Summary

These four ceremonies are only as effective as the amount of people they contain. Invite too many, and you may find that too many voices are in the conversation. Invite too few people, and you will find that you are not getting enough internal feedback. The goal is to find the happy medium.

- **Iteration Planning**: These should include the team, the team lead, and the PO. They are held at the beginning of each iteration and their length should be one hour for every week of the iteration. For example, a two-week iteration should have a two-week planning meeting. The PO comes to the meeting with a groomed product backlog, which they discuss with the team. They decide what work can completed in the iteration, and

place the rest into the backlog, which contains the work that can be pulled into the backlog, time permitting.

- **Coordination Meeting:** These should include the team, the team lead, and the PO. This is a short daily meeting, usually about 15 minutes, and held at a time convenient for the team, usually in the morning. But the time can be adjusted to the needs of your team. Participants in this meeting discuss the following: What did I complete yesterday? What am I working on today? Do I have anything blocking my work? Doing this provides accountability to your peers, because no one wants to be the blocker all the time.
- **Iteration Demonstration:** These should include the team, the team lead, and the PO. Stakeholders can also attend but are not required. Held at the end of an iteration, these meetings showcase work that has been done during the iteration. Work showed here should be demonstrable and meeting the quality standard of the team. In other words, don't show something half-baked here. They should be anywhere from 30 to 60 minutes.
- **Iteration Retrospective:** These should include the team, the team lead, and the PO. They are held at the end of each iteration and should last about 60 minutes. These meetings discuss what has been accomplished during the iteration, and also what worked, and what didn't work for the team. Discussing what works allows the team to better focus on those areas. Discussing what doesn't work allows the team to figure out solutions and develop a plan for dealing with things that don't work for the team.

These ceremonies should be scheduled in a way that makes the best use of a team's time. For example, some teams hold the iteration retrospectives and demonstrations at the same time, since the team may have agreed that doing so is an efficient use of their time.

Agile Artifacts

One of the things that sets agile apart from other management approaches is its ceremonies—sometimes referred to as rituals. These provide the framework for team to get work done in a structured manner, help to set expectations, empower the team to collaborate effectively, and ultimately drive results.

- The **product backlog** is the list of work required to create a product. This is the artifact that collects all the work flowing into the team. The product owner prioritizes the product backlog, signaling to the team which work is most important.
- The **iteration backlog** is the list of work to be completed in the current iteration, in the order determined by the team.
- A **burndown chart** is a graphic representation of how quickly the team is working through work items. The burndown chart shows the total effort against the amount of work for each iteration.
- A **user story** is a tool used in agile to capture a description of a feature from the user's perspective. A user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

User Stories

A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability—usually a user or customer of the system.

User stories shift the focus of the team from writing about requirements to talking about them. They contain a sentence or two and, more importantly, a series of conversations about the desired functionality.

Once a user story has been written, we need to estimate how much effort it will take. How easy or difficult will it be to implement?

User story estimates help teams plan their iterations.

There are many different estimation strategies, which fall under the Plan the Release process goal, under Choose Estimation.

When the Story Is Done

The agile way of working has two components built in.

1. The first is acceptance criteria. Acceptance criteria are unique to each user story.
2. The second is a definition of done—a checklist of what makes a user story "done."

The definition of done is an agreed upon set of items that must be satisfied before a user story can be considered complete.

How a Demo Works

1. Determine which tasks (based on user stories) meet the definition of "done."
2. Demonstrate new features (finished stories) to stakeholders.
3. Solicit feedback:
    a) Is this story ready for release? Does the feature or functionality meet the needs of the customer?
    b) Are there issues that require more work? Should this story be returned to the iteration backlog?

Lesson 2: Agile and Beyond

Description

One of the most important aspects of being a DASM is leading your team through the agile journey, embracing the DA mindset, understanding DA's central concepts, and acquainting yourself with the DA tool kit to customize your team's way of working and optimize your processes. This lesson provides an overview. You'll learn more about DA as you proceed through the training.

Objectives

Describe the significance of the Disciplined Agile Mindset

- Describe what Disciplined Agile is
- Define the eight principles of DA
- Describe the "promises" of DA
- Describe the "guidelines" of DA
- Describe how Disciplined Agile is an agnostic hybrid of approaches that leverages strategies from a variety of sources

Define the eight DA principles and how they are core to what sets Disciplined Agile apart from other agile frameworks

- Recognize the importance of making Delight Customers a priority
- Describe how Being Awesome is important for building a great agile team
- List the 5 levels of awareness (Enterprise Awareness)
- Identify how different contexts require different strategies – teams need to be able to own their own process and to experiment to discover what works in practice for them given the situation that they face. (Choice is Good)
- Identify how DA provides guardrails helping you to make better process choices, not strict rules that may not even be applicable given the context that you face. (Pragmatism Over Purism)
- Identify the potential factors to consider regarding the context of a given situation faced by a team. (Context Counts)
- Identify that the large number of strategies the DA toolkit supports to Optimize Flow.
- Explain the importance of organizing around products/services

Explain the elements of the process blade diagram

Discuss how to use the DA tool kit to tailor your way of working within a select phase according to context

- Explain what it means to be goal driven
- Define process blade and how process blades are used inside DA
- Describe the purpose of a goal diagram
- Describe how to read a goal diagram

- Describe the process goals of DAD

Agenda

1. Agile Is Showing Its Age
2. What Is Disciplined Agile?
3. The Disciplined Agile Mindset
4. What is Guided Continuous Improvement?
5. Disciplined Agile People
6. Disciplined Agile Flow
7. Disciplined Agile Practices
8. How DA Works

Lesson Notes

Agile Is Showing Its Age

Agile frameworks are being routinely imposed upon teams—as well as on entire organizations— whether they make sense for specific teams or not, presumably to provide management with some degree of control.

Often, leadership's decision-making process boils down to "ask an industry analyst what's popular" or "what are my competitors doing?" rather than what is best for our situation.

With the development of what Martin Fowler referred to as the "agile industrial complex," agile seems to have lost much of its agility.

As a DASSM, you should be aware of the emotional intelligence of your team members, especially when introducing a new Way of Working (WoW). Some may embrace the change while others may find it stressful. Effectively managing your emotions as well as the emotions of others will positively impact your project.

The Disciplined Agile Mindset

The Disciplined Agile mindset is informed by principles, promises and guidelines. Disciplined agilists believe in these principles, promise to adopt these behaviors, and follow these guidelines. There is a purpose for each aspect of the mindset.

- **Principles**. The principles provide a philosophical foundation for business agility. They are based on both lean and flow concepts.
- **Promises**. The promises are agreements that we make with our fellow teammates, our stakeholders, and other people within our organization whom we interact with. The promises define a collection of disciplined behaviors that enable us to collaborate effectively and professionally.
- **Guidelines**. These guidelines help us to be more effective in our way of working (WoW) and in improving our WoW over time.

The Disciplined Agile Mindset: Principles

- **Delight Customers**. Customers are delighted when our products and services not only fulfill their needs and expectations, but surpass them. Successful organizations delight their customers.
- **Be awesome** There are several things that we, as individuals, can do to be awesome.
    - Act in such a way that we earn the respect and trust of our colleagues: Be reliable, be honest, be open, be ethical, and treat them with respect.
    - Be willing to collaborate with others. Share information with them when asked, even if it is a work in progress. Offer help when it's needed and, just as important, reach out for help yourself.
    - Be an active learner. We should seek to master our craft, always being on the lookout for opportunities to experiment and learn. Go beyond our specialty and learn about the broader environment.
    - Seek to never let the team down. Yes, it will happen sometimes, and good teams understand and forgive that.
    - Simon Powers points out that we need to be willing to improve and manage our emotional responses to difficult situations. Innovation requires diversity, and by their very nature diverse opinions may cause emotional reactions. We must all work on making our workplace psychologically safe.
- **Context counts**. Every person is unique, with their own set of skills, preferences for work style, career goals and learning styles. Every team is unique, not only because it is composed of unique people, but also because it faces a unique situation. Our organization is also unique, even when there are other organizations that operate in the same marketplace that we do.
- **Be pragmatic**. Instead of requiring "best practices," Disciplined Agile provides strategies for maximizing the benefits of agile despite certain necessary compromises that are being made. As such, it's pragmatic—not purist—in its guidance.
- **Choice is good**. In order to provide people with choices from which they can choose their way of working. Disciplined Agile has gathered strategies from a wide variety of courses and put them into context. It combines strategies from methods, frameworks, bodies of knowledge, books, our practical experiences helping organizations to improve, and many other sources. Better choices lead to better outcomes, earlier—because choice is good.
- **Optimize flow**. Looking at the flow of value enables teams to collaborate in a way so as to effectively implement our organization's value streams. Although each team may be but one part of the value stream, they can see how they might align with others to maximize the realization of value.
- **Organize around products/services**. In Disciplined Agile, we don't organize around job function—such as having a sales group, a business analysis group, a data analytics group, and so on. We organize around products and services. This enables us to accomplish several important things:
    - Organizing around products and services enables us to identify and optimize the flows that count, which are value streams.

    - Organizing around products and services enables us to be laser-focused on delighting customers.
- **Enterprise awareness.** When people are enterprise aware, they're motivated to consider the overall needs of their organization, to ensure that what they're doing contributes to the goals of the organization and not just to the goals of their team. Enterprise awareness positively changes people's behaviors in several important ways, making them more likely to:
    - Work closely with enterprise professionals to seek their guidance.
    - Leverage and evolve existing assets within the organization, collaborating with the people responsible for those assets to do so.
    - Adopt and follow common guidance, tailoring it where needed, thereby increasing overall consistency and quality.
    - Share their learnings across teams, thereby speeding up their organization's overall improvement efforts.

The Disciplined Agile Mindset: Promises

- Create psychological safety and embrace diversity
- Accelerate value realization
- Collaborate proactively
- Make all work and workflow visible
- Improve predictability
- Keep workloads within capacity
- Improve continuously.

The Disciplined Agile Mindset: Guidelines

- Validate our learning
- Apply design thinking
- Attend to relationships through the value stream
- Create effective environments that foster joy
- Change culture by improving the system
- Create semi-autonomous, self-organizing teams
- Adopt measure to improve outcomes
- Leverage and enhance organizational assets

What is Guided Continuous Improvement

A Kaizen loop is an approach where a team experiments with a small change in their way of working, adopting the change if it works in their given context and abandoning it if it doesn't.

Continuous improvement is the act of applying a series of Kaizen loops to improve your way of working over time.

Some students have heard this referred to as PDCA—Plan-Do-Check-Act. Deming went back and forth between PDCA and PDSA and finally settled on PDSA.

That approach is a bit too simple for more organizations. They need something a bit more prescriptive—more step by step.

The key idea is that your team should be always experimenting with new WoWs.

Guided continuous improvement extends the Kaizen loop strategy to use proven guidance to help teams identify techniques that are likely to work in their context. This increases the percentage of successful experiments and thereby increases the overall rate of process improvement.

And that's where Disciplined Agile comes in. It enables you to increase your rate of process improvement by helping you to identify strategies that are more likely to succeed given your situation.

Disciplined Agile enables you to increase your rate of process improvement by helping you to identify strategies that are more likely to succeed given your situation.

If we get better at this, we will succeed more often, and we will improve faster. We can do this if:

- We have access to an experienced agile coach, but they're expensive and hard to find.
- We have access to a process knowledgebase, like the Disciplined Agile tool kit.

Some experiments fail. You'll learn something, but it's still a failure.

Failing fast is fine but succeeding early is better.

Disciplined Agile Life Cycles

A life cycle is a process for planning, creating, testing and deploying a product or service. Disciplined Agile life cycles provide teams with the flexibility of choosing an approach that makes sense for them. They are an essential tool for teams choosing their own way of working.

- Disciplined Agile Delivery supports several life cycles, which are shown here.
- Disciplined Agile describes some additional life cycles, especially DA FLEX, which is a life cycle at the value stream.
- Disciplined Agile also has life cycles for business teams.
- The flow aspects of process blades are described via either a life cycle or workflow diagram(s).

Disciplined Agile Practices: How Does Disciplined Agile Work?

Knowing where you want to end up is having a goal.

- Goals give us focus.
- Goals allow us to measure progress.
- Goals help us remain committed and undistracted.
- Goals help us overcome procrastination.
- Goals give us motivation.

That's why Disciplined Agile takes a goal-driven approach.

Most teams struggle to truly own their process—mostly because they don't have the process expertise within the team to do so.

A few key points:

- Every team faces a unique situation and therefore should tailor their approach to best address that situation and evolve their way of working (WoW) as the situation evolves. In other words, context counts.
- You need to have choices and know what those choices are—you can't own your process if you don't know what your options are.
- We want to be awesome at what we do, so we need the flexibility to experiment with ways of working so that we can discover how to be the most awesome team we can be.

There are several fundamental advantages to taking a goal-driven approach to agile solution delivery:

- **It enables teams to focus on process outcomes**, not on process compliance.
- **It provides a concise, shared pathway to leaner, less wasteful process decisions**
- **It makes process decisions explicit**, helping your team choose its way of working.
- **It makes your process options very clear** and thereby makes it easier to identify the appropriate strategy for the situation you find yourself in.
- **It enables effective scaling** by providing you with strategies that are sophisticated enough to address the complexities that you face at scale.
- **It takes the guesswork out of extending agile methods** and thereby enables you to focus on your actual job, which is to provide value to your stakeholders.
- **It makes it clear what risks you're taking on** and thus enables you to increase the likelihood of success.
- **It hints at an agile maturity model** (this is important for any organization struggling to move away from traditional maturity models).

Complex Adaptive Systems

A complex adaptive system is a system in which a perfect understanding of the individual parts does not automatically convey a perfect understanding of the whole system's behavior.

Organizations are a collection of interacting teams and groups, each of which evolves continuously.

As teams evolve their ways of working, they motivate changes in the teams with whom they interact.

Because of this constant process evolution—hopefully for the better—and because people are unique, it's impossible to predict how people are going to work together or what the results of that work will be.

In short, your organization is a complex adaptive system.

So, How Does Disciplined Agile Work?

And, unfortunately, there are no one-size-fits-all solutions.

This may sound obvious, but you would be surprised at the number of organizations that overlook it. They seek to impose a single solution on all teams, regardless of whether the solution is appropriate or not.

Since your organization is a complex adaptive system, no single approach will work for all the groups in it.

That's where Disciplined Agile comes in.

Disciplined Agile is an agnostic hybrid that leverages strategies from a variety of sources.

- Disciplined Agile does much of the heavy lifting when it comes to answering the question "How do all of the various agile techniques fit together?"
- In many ways, Disciplined Agile is the process glue that connects the various agile practices.
- Disciplined Agile leverages ideas from a wide variety of agile, lean, iterative and even traditional sources.

To accommodate the wide variety of needs in a complex organization, the Disciplined Agile tool kit captures team-level strategies in a series of process blades.

A process blade addresses a specific organizational capability, such as finance, people management, data management, agile solution delivery or vendor management.

A process blade encompasses a cohesive collection of process options—including practices, strategies and workflows—that should be chosen and then applied in a context-sensitive manner

Why is it called a blade?

A process blade is called a blade to imply that it can be updated or even replaced, just like a blade server in your IT infrastructure, where components can be independently replaced. As the situation a team faces evolves, a team needs to be able to update their configuration of a process blade—or even replace it entirely—with little or no impact to the team around them. Hence, a process blade is the process equivalent of a server blade.

If you don't have data center experience, think of a blade as you would think of the different blades in your kitchen's knife block. Each knife serves a different purpose: a paring knife for fruit, a serrated knife for cutting bread, a cleaver for chopping through bones and so on. In the same way, each process blade services a different function: data management, portfolio management and so on.

Disciplined Agile Tool Kit Process Blades

To help you to navigate the wealth of advice contained in the DA tool kit, the process blades have been organized into four layers:

- **Foundation:** Provides the conceptual underpinnings of the Disciplined Agile tool kit. This includes the principles, promises and guidelines of the Disciplined Agile mindset; fundamental concepts from both agile and lean; fundamental concepts from serial/traditional approaches; roles and team structures; and the fundamentals of choosing your way of working.
- **Disciplined DevOps:** Streamlining of solution development and operations, and Disciplined DevOps is an enterprise-class approach to DevOps. This layer includes Disciplined Agile Delivery and other enterprise aspects of DevOps.
- **Value Stream:** based on Al Shalloway's Flow for Enterprise Transformation, or "FLEX." It's not enough to be innovative in ideas if these ideas can't be realized in the marketplace or in the company. FLEX is the glue that ties an organization's strategies in that it visualizes what an effective value stream looks like, enabling the team to make decisions for improving each part of the organization within the context of the whole.
- **Disciplined Agile Enterprise:** Senses and responds swiftly to changes in the marketplace. It does this through an organizational culture and structure that facilitates change within the context of the situation that it faces. Such organizations require a learning mindset in the mainstream business and underlying lean and agile processes to drive innovation. The Disciplined Agile Enterprise layer focuses on the rest of the enterprise activities that support an organization's value streams.

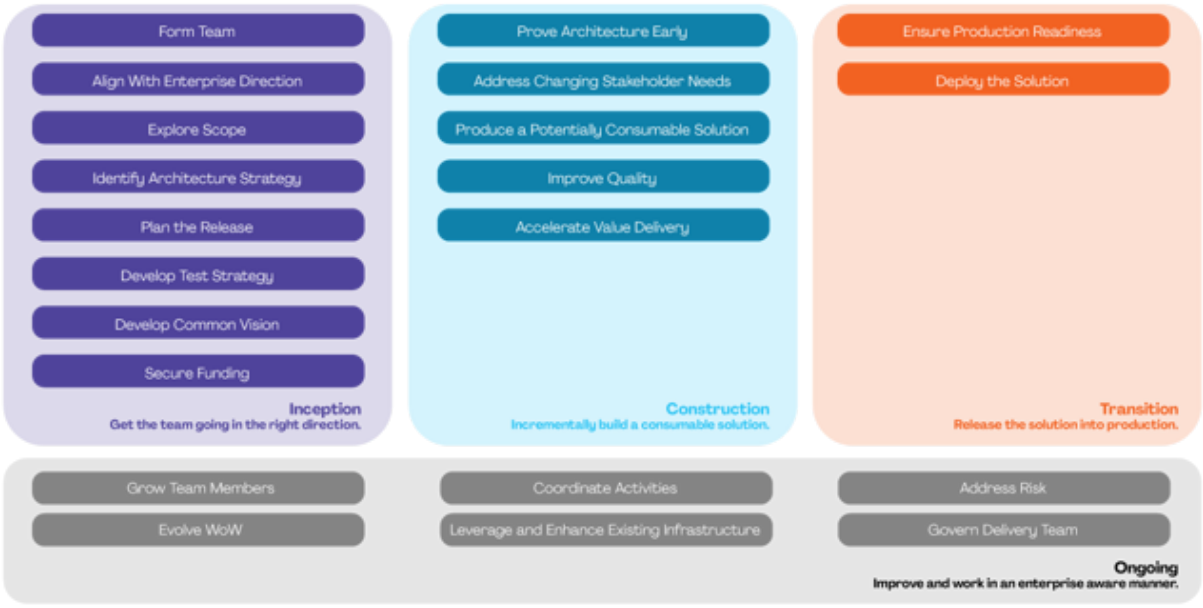
How Does Disciplined Agile Work?

Disciplined Agile lets you drill through four layers of detail:

Process Goals

A process goal captures the detailed, process-related decisions and options for a cohesive subset of a team's way of working (WoW). They provide guidance so that a team can tailor and scale agile strategies given the context of the situation they face. Sometimes called a process capability.

Process Goal Diagrams



Each of these process goals is elaborated by a process goal diagram.

- These are examples of goal diagrams.
- We apply goal diagrams at all levels of the tool kit.

Process Goal Diagrams are visual depictions of the aspects you need to think through about the goal—which we call “decision points”—and several options for each decision point to choose from. These don't identify every possible technique available. However, they provide you with a good range of options and to make it clear that you do, in fact, have choices.

It has three main components.

1. The process goal itself is the beginning of the diagram.
2. Each goal is tied to a collection of decision points. These are issues that your team needs to determine whether they need to address and, if so, how they will do so.
3. Each decision point is tied to a list of potential practices and strategies that address it. In many cases, these can be combined.

Because there may be many techniques to choose from, “default” techniques are listed in boldface italic type.

Tied to each decision point is a list of potential practices and strategies. Each has its advantages and disadvantages:

- An **ordered option list** is depicted with an arrow to the left of the list of techniques. In an ordered list, the most desirable, most effective techniques appear at the top of the list and the less desirable techniques are at the bottom of the list.
- An **unordered option list** is depicted without an arrow—each option has advantages and disadvantages, but it isn’t clear how to rank the options fairly.

- **Potential starting points** are shown in bold italics. These are good starting points for small teams new to agile that are taking on a fairly straightforward problem.

Often you will find that you need to examine the trade-offs associated with each of the options you're considering.

That's where the option descriptions and trade-offs tables come in.

These list each of the options associated with a given decision point in the left column, then explain some of the trade-offs associated with that option in the right.

Lesson 3: Building and Supporting a Discipline Agile Team

Description

As a DASM, your primary focus is your team. In this lesson, you'll learn about Disciplined Agile teams, the characteristics of a leader, and how you can support your team so that they can be their best. This lesson will also introduce you to aspects of a team's context that can impact how you form a team and evolve your way of working (WoW).

Objectives

Explain how people are organized into DA teams

- Compare and contrast leaders to managers
- Identify roles that can be leaders
- Describe potential, primary and secondary roles on DA teams

Define the primary DA roles and how they each are key to the success of a self-organizing agile team

- Describe the 5 Primary DA roles
- Describe the responsibilities of the 5 primary DA roles
- Describe why each of the 5 primary DA roles is important
- Explain how to help your team work well together

Agenda

1. Disciplined Agile People
   - Roles on a DA team
2. Team Lead
   - The DASM
   - Leader Qualities
   - Supporting the Team
3. Types of Team
4. Tactical scaling factors
5. Team Context and Scaling Factors

Lesson Notes

Be Awesome – DA Principle

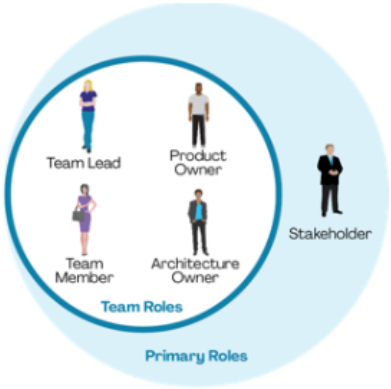
DA teams are awesome and foster joy. Let's review the ways teams can use the DA principle "Be Awesome."

- Act in such a way that we earn the respect and trust of our colleagues. Be reliable, be honest, be open, be ethical and treat them with respect. Second, willingly collaborate with others.
- Share information with them when asked, even if it is a work in progress. Offer help when it is needed and, just as important, reach out for help yourself.
- Be an active learner. We should seek to master our craft, always being on the lookout for opportunities to experiment and learn. Go beyond our specialty and learn about the broader software process and business environment.
- Seek to never let the team down. Yes, it will happen sometimes, and good teams understand and forgive that.
- Be willing to improve and manage our emotional responses to difficult situations. Innovation requires diversity, and by their very nature, diverse opinions may cause emotional reactions. We must all work on making our workplace psychologically safe.

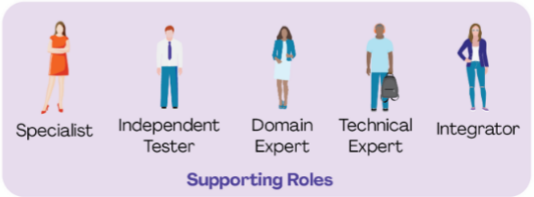
Building a DA Team

DA Delivery teams can be made of different combinations of roles, depending on the skills the team needs. These teams include **primary** and **secondary** roles.

- **Primary roles** are the key roles on a delivery team. Every team needs these roles.



- **Secondary roles** are additional roles. Any one or more of these roles might be included on a team. There may be other secondary roles not included here.



There Is No Standard for Agile Terminology

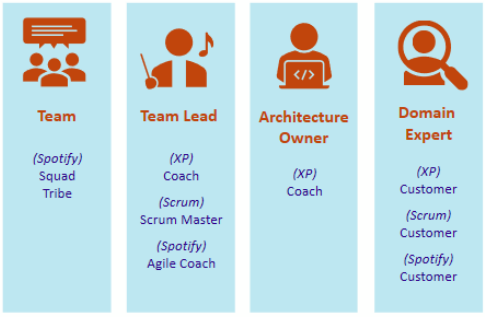
Disciplined Agile strives to be agnostic in its terminology.

It does not favor any methodology.

DA's Generic Terms

DA uses generic terms for roles you likely have different names for.

Here are four common **DA roles** and their corresponding names from XP, Scrum and Spotify.



We have only listed the instances where a title different from DA's is used. You don't need to know these terms or be familiar with their methodologies. The point is, what you are familiar with may be different from the DA term.

To work with DA roles, your team may need to change their mindset and skill to transition into DA roles.

Summary of Key Roles

A delivery team is made up of four roles:

- The **team members** main job is to build the product
- The **architecture owner**'s main job is to ensure that the team builds the product right
- The **team lead's** main job is to coordinate the building of the product
- The **product owner's** main job is to ensure that the team builds the right product

Primary DA Roles

Primary roles are ones that we typically see on all teams regardless of the situation.

Team Lead	• Agile process expert • Keeps team focused on achievement of goals • Removes impediments
Product Owner	• Owns the product vision, scope and priorities of the solution
Architecture Owner	• Owns the architecture decisions and technical priorities • Mitigates key technical risks
Team Member	• People able to work in cross-functional roles to deliver the solution
Stakeholder	• Includes the customer but also other stakeholders such as the sponsor, operations engineers, support staff, architecture, database group and finance

A few key points to keep in mind:

- DA explicitly brings in the role of architecture owner.
- The Product Owner (PO) should not be the team lead or the Architect Owner (AO).
- The PO is part of the team. Scrum typically does not include the PO as part of the team.
- Stakeholder is more robust term than customer, although we really love the word customer.

Note: *Chapter 4 of the Choose Your WoW book describes the rights and responsibilities of the primary and supporting roles.*

Supporting DA Roles

DA supporting roles and their responsibilities are listed below.

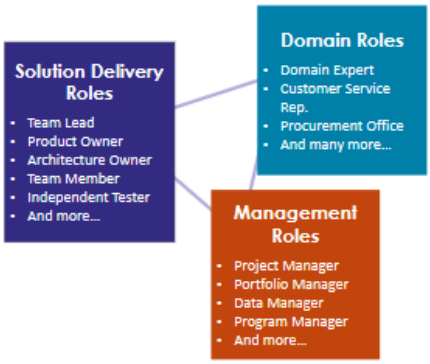
Independent Tester	• A test/quality professional outside of the team who validates their work
Specialist	• Someone in a specialist role, such as business analyst, program manager or enterprise architect
Domain Expert (SME)	• Someone with deep knowledge of the domain, such as a legal expert or marketing expert who is brought in as needed to share their expertise
Technical Expert	• Someone with deep technical knowledge, such as a security engineer or user experience (UX) professional, whose help is needed for a short period
Integrator	• Someone responsible for the operation of the overall team build

Note: *Disciplined Agile addresses many of the roles that are common in modern organizations. But, because every organization is unique, it isn't possible to cover all potential roles.*

People Can Fulfill More Than One Role

There are many different types of roles, and people can hold more than one role.

Below are different roles based on the functional area.



What is a Disciplined Agile Scrum Master (DASM)?

A DASM is simply a DA Team Lead that practices the Scrum methodology.

Throughout most of this training, references to the DA term “Team Lead” and the course term “DASM” are interchangeable.

Who does the DASM interact with?

They play an important role interacting and serving the team.

They also interact with stakeholders and neighboring teams. In addition, to effectively interact with others, they must have people skills.

What does a DASM do?

A DASM …

- will lead and guide their team through their agile journey
- is qualified to work with one team on straightforward situations, scaling on a team level and utilizing basic team coaching skills
- practices team-oriented agile and lean techniques
- applies the Disciplined Agile tool kit at a beginner level to solve problems and improve processes.

How does the DASM serve the team?

The **DASM serves the team** by:

- Coaching team members in self-management
- Helping the team focus on creating high-value increments that meet the definition of done.
- Ensuring that necessary events take place and are

- positive
- productive
- within timebox

How does the DASM serve the product owner?

A **DASM serves the product owner** by:

- Helping find techniques for effective product goal definition and backlog management
- Supporting in backlog grooming and planning
- Keeping informed of project status
- Helping the team understand the need for clear and concise backlog items
- Facilitating stakeholder collaboration as requested or needed

How does the DASM serve the organization?

A **DASM serves the organization** by:

- Supporting product owner and team in achieving customer satisfaction
- Helping the team identify and address risks
- Helping with training and coaching in agile adoption
- Helping employees and stakeholders understand and work in their environment
- Removing barriers to progress

Leaders vs. Managers

Key concepts

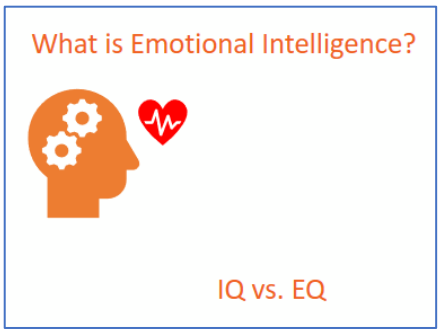
- Teams need leaders more than they need managers.
- Managers still are important; they add value.
- There is some overlap between managers and leaders, but there are some important differences too.
- Most great managers are also good leaders.
- **Agile teams are led, not managed.**
- Develop leadership skills so that you can guide, coach and support your team.

What is Emotional Intelligence and why is it important?

Emotional Intelligence is the ability to understand your emotions and manage them in positive and productive way. It's also the ability to be able to understand, relate and work effectively with others.

It's important because it plays an important part in your success in leading a team.

While you may have the technical skills, or IQ, to do your job, you'll need emotional intelligence or EQ, to motivate, support, and guide your team to high performance.



Leaders with emotional intelligence

- Acknowledge and manage your emotions in a positive way.
- Redirect negative thoughts and behaviors to positive ones.
- Think before you speak when emotions are high.
- Relate to and work effectively with others.
- Show empathy and concern for others.
- Understand that it's not just technical skills (IQ) that make you a good leader. You'll also need Emotional Intelligence (or EQ).
- Motivate, support, and guide your team toward high performance.

Here are some emotional intelligence skills you can build on:

Be Self Aware and Self Manage

Pay attention to what you are feeling (self-aware).

Manage your emotions in a constructive way so can build positive relationships (self-manage)

Promote Psychological Safety

Create an environment of psychological safety where the team feel safe to express themselves and feel that they belong.

Show empathy for others' emotions and concerns to establish healthy working relationships.

Embrace Diversity

The team is a mix of unique individuals with differing communication styles and personalities.

Help the team appreciate their differences.

Foster Joy

Recognize that a joyful team is a productive one.

Resolve Conflict

Recognize that some conflict can be constructive.

Help dissenting parties reach mutually agreeable solutions.

You are the mediator who steps in, resolves the issue, and keeps the project on course.

Middle-up-Down-Management

The DASM supports the team by helping employees and stakeholders understand and work in their environment. One way to do this is by following the lean management concept of "middle-up-down management," which can apply to the team lead, as well as to higher levels of middle management.

Business stakeholders can clearly set the vision.

Mid-level managers create an ecosystem within which people work to implement that vision:

- They engage with business shareholders.
- They create an environment to facilitate the manifestation of management's vision.
- They work with their teams to ensure the environment supports them.
- At the team level people self-organize to implement the vision.

To learn more, see Nonaka, Ikujiro (1988). Toward Middle-Up-Down Management: Accelerating Information Creation. *MIT Sloan Management Review*, Spring 1988.

Team Working Agreements

Internal

A team working agreement defines their internal way of working and how they are willing to interact with other teams.

External

External working agreements are sometimes defined in terms of service level agreements.

Types of Teams – Project vs. Product/Long-Standing team

The project vs. product/long-standing team issue is critical to organizations.

Most organizations will have both project and long-standing teams.

A long-standing team can take on short-term projects (it's merely a big chunk of work).

Project Teams:	Long-Standing Teams:
Bring the people to the work	Bring the work to the team
Potential for significant budgeting and tracking overhead	Budgeting is straightforward
Motivates building teams based on who is available at the time	Motivates thoughtful team building and evolution
Significant overhead in the team gelling	Motivates desire to learn and improve the team's way of working
Short-term focus motivates quality shortcuts	Engenders a focus on long-term issues such as advertisement, quality and evolution

Shared Team Services

Collaborative Enterprise Teams

Individuals are members of both a work team and an enterprise team.

In some cases, the work teams will nominate who will be in the enterprise role for them.

In other cases, the enterprise team will assign one of their people to support a work team.

This requires a lot of people to be part of the enterprise team.

Shared Services Enterprise Teams

These teams fulfill requests for work from other teams.

Team Context and Scaling Factors

Every team is in a unique situation.

There is more to scaling than just team size.

The spider diagram below shows each scaling factor for tactical scaling. The further out you are on each leg, the more complex your team's context is. This means you will face more challenges in scaling and may need to look at tailoring in more areas than if your context was near the center of the diagram.

Team size – Scaling Factor

Key Concepts

- Large teams are organized differently than small teams.
- Globally distributed teams are organized differently than collocated teams.
- Organizational distribution, some people working for an outsourcer, can also affect things.
- This lesson works through these issues.
- Sometimes we need to bring in technical or domain experts to help us out for a short period.
- Sometimes we're supported by an independent tester due to regulatory compliance concerns.
- The line with the diamond at the end is the UML notation for composition.
- Coordination within medium-sized teams can likely be accomplished via a "Scrum of Scrums."
- Each subteam builds a portion of the overall solution (features, components, …).
- Large teams: The leadership teams will self-organize and meet and coordinate with each other as they see fit.
- Large teams: The program manager/coordinator coordinates the overall program and leadership subteams (called an RTE in SAFe®).

Geographic Distribution – Scaling Factor

As team members are more geographically dispersed, coordination becomes more difficult and more sophisticated communication and coordination methods are required. Tools like conferencing or messaging apps are vital here.

Organizational Distribution – Scaling Factor

Compliance – Scaling Factor

Technical Complexity – Scaling Factor

Domain Complexity – Scaling Factor

Tactical Agility at Scale

Using a tool like the spider diagram supports tactical agility at scale: scaling agile at the team level. The aim is to apply agile deeply to address all the complexities/scaling factors appropriately.

Lesson 4: Choosing Your WoW!

Description

To be an effective DASM for a delivery team, you'll want to have a firm understanding of the Disciplined DevOps layer so that you can effectively help your team meet complex challenges in producing high quality solutions on time. With your understanding of the DA landscape and the DA tool kit, you can optimize how the team works with the Disciplined DevOps layer. You are also equipped to identify and help resolve impediments in this layer that the team faces.

Objectives

Describe what business agility is and how it is core to the value proposition of Disciplined Agile

- Define business agility
- Identify the full range of business agility

Determine which situations each of the DA life cycles is best applied

- Describe how DA supports a variety of lifecycles
- Identify the 3 phases of the DAD delivery cycle
- Describe the Agile life cycle and identify when to use
- Describe the Lean life cycle and identify when to use
- Describe the continuous delivery Agile life cycle and identify when to use
- Describe the continuous delivery Lean life cycle and identify when to use
- Describe the exploratory life cycle and identify when to use
- Describe the program life cycle and identify when to use
- Describe the business agile and business lean life cycles
- Identify how to choose a life cycle and who chooses

Apply the DA practice of choosing a team's way of working (WoW)

- List the 5 steps for choosing your WoW
- Analyze the context using the spider chart
- List factors impacting context when choosing a team's WoW
- Select best-fit life cycle using the decision tree

Agenda

1. What is business agility?
2. What is a complex adaptive system?
3. Why do we want to be able to choose our team's way of working?
4. What are the Disciplined Agile life cycles?
5. How do you choose your way of working?

Lesson Notes

What is Business Agility?

Business agility is an organization's ability to rapidly adapt to market and environmental changes in productive and cost-effective ways.

Business agility focuses on value realized by having stakeholders identify, prioritize and sequence the work to be done and allocate it appropriately to the product/service teams.

This is sometimes referred to as enterprise or organizational agility.

Business agility enables the realization of the highest value in a shorter amount of time, predictably, sustainably and with high quality.

By working in small delivery increments, we continuously adjust to what is needed, enabling the organization to change direction at low cost.

Why Do We Want to Be Able to Choose Our Team's Way of Working?

We want to be able to choose our team's way of working so we give those teams the flexibility to adapt to changing circumstances, which is at the heart of agility. It gives the pieces of the organization—and, by extension, the organization itself—the ability to adapt.

And **because** your organization is a complex adaptive system. What works for some teams may not work for others. And even if a specific way of working does work for another team, there may be unintended consequences outside that team.

Things to consider when choosing your way of working:

- Every team has a different way of working.
- We evolve our way of working to reflect what we learn when we work with other teams.
- We accomplish our goals by working with other teams.

What Are the Discipline Agile Life Cycles?

A life cycle is a process for planning, creating, testing and deploying a product or service.

Disciplined Agile life cycles provide teams with the flexibility of choosing an approach that makes sense for them. They are an essential tool for teams choosing their own way of working.

- Disciplined Agile Delivery supports several life cycles, which are shown here.
- Disciplined Agile also has life cycles for business teams.
- The flow aspects of process blades are described via either a life cycle or workflow diagram(s)

Disciplined Agile Life Cycles—Agile

It extends Scrum's construction life cycle. In addition to being a more detailed view of the life cycle, we consider several other interesting aspects:

- It is iteration based.
- It uses non-Scrum terminology. Remember, Disciplined Agile is agnostic!
- It shows inputs from outside the delivery life cycle.
- There is a work item list, not a product backlog.
- It includes explicit milestones. Along the bottom of the life cycle diagram, see the suggested lightweight milestones that teams should strive to meet. Such milestones are an important aspect of agile governance.

When to use the Agile life cycle:

The work:

- is primarily enhancements or new features
- can be identified, prioritized, and estimated early in the project

The team:

- is new to agile practices
- is familiar with Scrum and Extreme Programming (XP)
- is typically working on a project

Disciplined Agile Life Cycles—Lean

There are several interesting features to this life cycle:

- **It supports a continuous flow of work**. In this life cycle, the solution is deployed as often, and whenever, it makes sense to do so. Work is pulled into the team when there is capacity to do it, not at the regular cadence of an iteration.
- **Practices follow their own cadences**. With iterations/sprints, many practices (detailed planning, retrospectives, demos, detailed modeling and so on) effectively follow the same cadence, which is that of the iteration. In a lean approach, the conventional wisdom is to do something when it makes sense to do it, and not when the calendar indicates that you're scheduled to do it
- **It has a work item pool**. All work items are not created equal, although you may choose to prioritize some work by business value. Some work, particularly work resulting from legislation, is date driven. Some work must be expedited, such as fixing a "severity one" production problem. So, a work item pool instead of a prioritized stack makes a bit more sense when you recognize these realities.

When to use the Lean life cycle:

The work is:

- broken down into very small work items of roughly the same size
- difficult to predict in advance

The team:

- favors the lean approach of minimizing batch and any planning in advance of doing the work
- is typically working on a project

Disciplined Agile Life Cycles—Continuous Delivery: Agile

There are several interesting features to this life cycle:

- Common for stable/long-lived teams doing agile
- Iterations are typically 1-2 weeks, although 1 day is also common
- At the end of the iteration, you release into production
- More like a "very regular delivery" life cycle than a continuous delivery life cycle

When to apply Continuous Delivery: Agile…

- The work:
  - remains relatively stable within an iteration
  - consists of a series of releases over time
- The organization develops streamlined deployment practices and procedures
- The team:
  - is long-lived and stable
  - can deliver solutions to stakeholders on a frequent and incremental basis
  - can show value to stakeholders rapidly, especially before the completion of the entire solution
  - form mature DevOps practices including continuous integration, continuous deployment, and automated regression testing

Disciplined Agile Life Cycles—Continuous Delivery: Lean

There are several interesting features to this life cycle:

- A good end goal for your improvement efforts
- The product is shipped into production or the marketplace on a very regular basis. This could be as often as daily, although weekly or monthly is quite common too.

When to choose Continuous Delivery: Lean…

- The work consists of a series of releases over time
- The organization develops streamlined deployment practices and procedures
- The team
  - is long-lived and stable
  - can deliver solutions to stakeholders on a frequent and incremental basis
  - can show project value to stakeholders rapidly, especially before the completion of the entire solution
  - forms mature DevOps practices including continuous integration, continuous deployment, and automated regression testing

Disciplined Agile Life Cycles—Exploratory

It's also referred to as the "Lean Startup" life cycle.

Sometimes it takes time to identify what your stakeholders need.

This life cycle is used by teams that find themselves in startup or research situations where their stakeholders have an idea for a new product, but they do not yet understand what is needed by

their user base. As a result, they need to quickly explore what the market wants. This is best done using a series of short learning experiments. When the team is ready to productize, they move to another life cycle.

Apply the Exploratory Life Cycle when…

- The solution addresses high-incertitude cases, such as a new, unexplored market, or a new product.
- The stakeholders and delivery team are very flexible in adapting the solution as it is being developed.
- You have one or more valid hypotheses/strategies to test with clear go/no-go criteria for when the test is over.
- You are willing to experiment and evolve your idea based on your learnings.

Disciplined Agile Life Cycles—Continuous Delivery: Exploratory

It's also referred to as the "Lean Startup" life cycle.

Sometimes it takes time to identify what your stakeholders need.

This life cycle is used by teams that find themselves in startup or research situations where their stakeholders have an idea for a new product, but they do not yet understand what is needed by their user base. As a result, they need to quickly explore what the market wants. This is best done using a series of short learning experiments. When the team is ready to productize, they move to another life cycle.

Apply the Exploratory Life Cycle when…

- The solution addresses high-incertitude cases, such as a new, unexplored market, or a new product.
- The stakeholders and delivery team are very flexible in adapting the solution as it is being developed.
- You have one or more valid hypotheses/strategies to test with clear go/no-go criteria for when the test is over.
- You are willing to experiment and evolve your idea based on your learnings.

Disciplined Agile Life Cycles—Program

There are several interesting features to this life cycle:

- Best-Fit Life Cycle for a team of teams
- The subteams/squads may have their own ways of working
- You need to coordinate the work (backlog), the architecture and people across subteams
- The subteams may be on their own cadence, although having a common cadence makes work easier

The Program life cycle should be applied when…

- You need a large team of teams. Some problems require a large team, and in some cases, you may even decide to organize that large team into a team of teams.

- You have the skills to implement agile at scale.

Common Project Phases

You may have noticed that every Disciplined Agile life cycle has phases delineated, although not every life cycle includes every phase.

That's because Project-based life cycles—even agile and lean ones—go through phases.

It all starts with **Inception** when the team envisions and plans the project, doing just enough work to get organized and get going in the right direction. The team will initially form itself, then invest some time in initial requirements and architecture exploration, initial planning, aligning itself with the rest of the organization and securing funding for the rest of the project.

The process continues with **Construction**. The team produces a consumable solution with enough functionality to be valuable to stakeholders. During this phase, the team will be performing analysis, design, programming, testing and management activities every single day.

And finally, the process concludes with **Transition**. The team releases its solution into production. This includes both determining whether the solution is ready to be deployed and then actually deploying it.

In addition to these three phases, many projects include a number of **Ongoing** activities that occur continuously through the other three phases.

Disciplined Agile Milestones

Milestone	Fundamental Question Asked
Stakeholder vision	Do stakeholders agree with your strategy?
Proven architecture	Can you actually build this?
Continued viability	Does the effort still make sense?
Sufficient functionality	Has the team produced (at least) a Minimum Business Increment (MBI)?
Production ready	Will the solution work in production?
Delighted stakeholders	Are stakeholders happy with the deployed solution?

This is what it looks like when you combine these critical views into a single diagram.

Ideally, we want to streamline the overall flow between all of these activities.

And that is the essence of Disciplined DevOps.

How Do You Choose Your Way of Working?

There are five steps that will help you choose your team's Way of Working:

1. **Analyze the context**: What context does your team face? Factor in your team's size, geographic distribution, organizational distribution, compliance requirements, technical complexity, and domain complexity.
2. **Select the Best-Fit Life Cycle**: Given the team's context, which life cycle is the best fit? Remember: the life cycle is a starting point that can be changed at a later time when it makes sense.
3. **Connect the Dots**: Given your context and life cycle, which process goals and diagrams should you consider first? Which process goals are the least relevant given your team's situation.
4. **Make Some Choices**: Within the relevant goal diagrams, make some choices for the team's way of working. Refer to the options table in *Choose Your Wow!* To learn more and to review pros and cons of each option.
5. **Practice Continuous Improvement**: What context does your team face? Factor in your team's size, geographic distribution, organizational distribution, compliance requirements, technical complexity, and domain complexity.

Analyze the Context

We have a powerful tool we use to visualize a team's context; we call it a spider diagram.

Each of the arms of this diagram represents a different factor that affects the team's context:

- Geographic distribution
- Team size
- Domain complexity
- Technical complexity
- Compliance
- Organizational distribution

To analyze a team's context, the rate of these factors is rated on a linear scale; the farther away from the center of the diagram it is, the riskier—or more complex—that factor is.

Select the Best-Fit Life Cycle

We have a have another powerful tool we use to help select the most appropriate life cycle. It's a life cycle decision tree.

To use it, simply work through each of the decision points, answering "yes" or "no" depending on your context.

In the end, it will *recommend a life cycle most appropriate to your team.*

Lesson 5: Tailoring Your Practices: Inception Phase

Description

As DASSMs, we want to contribute to our organization's business agility. Understanding the value stream layer will help you position your team to achieve this. With your understanding of the DA Landscape and the DA tool kit, you can optimize how the team works with the value stream layer. You will also be equipped to identify and help resolve impediments the teams face in the value stream layer.

Objectives

Describe the inception phase and why it is important.

- Define Inception
- Identify the process goals associated with the Inception phase

Discuss how to use the DA tool kit to tailor your way of working within a select phase according to context

- Rank and select process goals according to their relevance to the phase and the team's context
- Identify key practices for the team try using goal diagrams

Agenda

1. DA Phases
2. DA Process Goals
3. Agile Practices: Plan the Release (Agile life cycle)
4. Choices in the Inception Phase

Lesson Notes

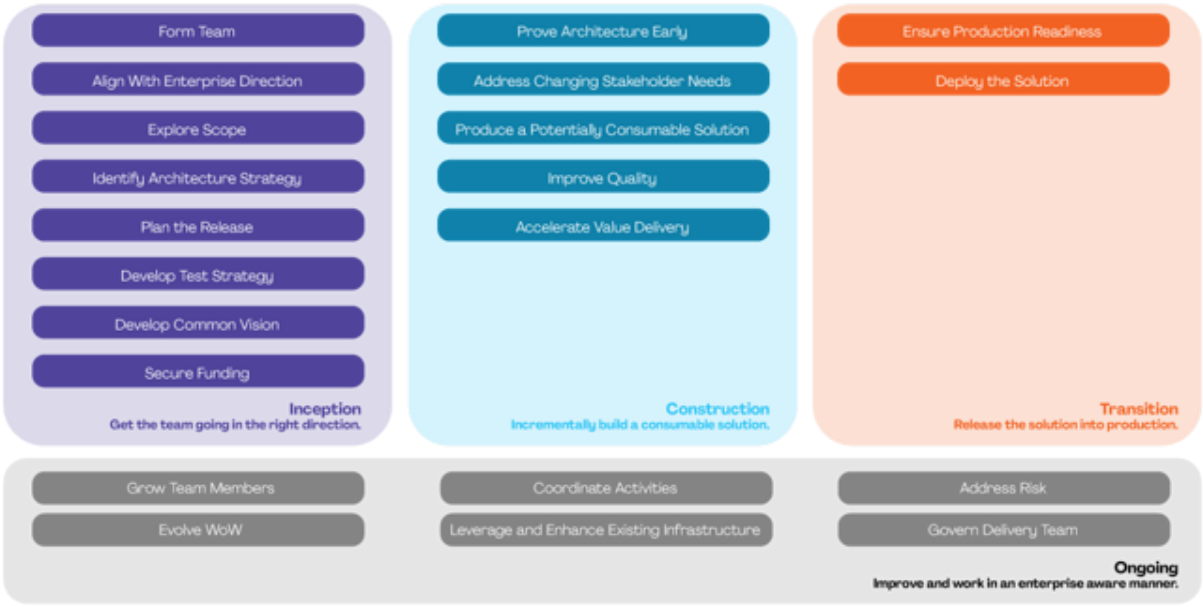
DA Phases and Process Goals

The Disciplined Agile phases are: Inception, Construction and Transition. Long-standing teams also have ongoing considerations.

- **Inception**: Get the team going in the right direction. Initiate the endeavor in a streamlined manner, including initial planning and modeling, and obtain agreement regarding the vision with the primary stakeholders.
- **Construction**: Develop a consumable solution in a collaborative and incremental manner. The team requests feedback on a regular basis, which they then act on accordingly.
- **Transition**: Release/deploy the solution to stakeholders, ideally at the point where a new minimum business increment (MBI) has been developed.

Ongoing is not a phase but a category that spans the phases.

DA Process Goals in Each Phase



- Disciplined Agile is **goal driven**.
- The DA tool kit guides people through **process-related decisions**.
- Types of decisions are organized under **process goals.**
- There are **21 process goals** under Disciplined Agile Delivery (DAD).
- If a team is having a problem in a certain **phase**, they can find a list of areas where they can look for options to try.

Agile Practices

Writing User Story

During the Inception phase, teams practicing the Agile life cycle plan their release.

Part of planning the release means exploring the scope of the upcoming work.

One tool teams often utilize is the user story.

How do you choose an option if you don't know what each of them means?

Your *Choose Your WoW!* book tells you about each of them when you look up Explore Usage in Chapter 9 "Explore Scope" and review the Options Table.

Writing User Stories

Explain that once a story is written, we need to estimate how much effort it will take. How easy or difficult will it be to implement?

User story estimates help teams plan their iterations.

There are many different estimation strategies, which fall under the Plan the Release process goal under Choose Estimation.

Read about each alternative in the options table in Chapter 11 of your *Choose Your WoW!* book.

Knowing When a Story Is Done

Another important practice for Agile life cycle teams is determining how they will know when each piece of work is done. The item should achieve a level of quality according to parameters set by the team. One way to accomplish this with acceptance criteria, which are developed for each item during the Inception phase.

Determining when a story is done falls under the Explore Scope process goal. under Explore Quality Requirements. You can learn more about these options and the risks each entails by reading the options table in the Explore Quality Requirements section of Chapter 9 in your *Choose Your WoW! book*

Choices in the Inception Phase

Here is an example you can use or adapt, if you wish.

The Tiger team works for a large retailer. Their context has changed over the years.

They have grown to 20 people, including some contractors.

Their work on a retail website is more complex and now includes mobile apps.

The complexity includes not only architecture, but also regulatory requirements as they deal with customer and financial data.

The team is now distributed across the continent.

They are expected to create an environment conducive to collaboration.

Connect the Dots

Given the team's context and life cycle, which goals are most relevant?

Make Some Choices

Review the steps for the Develop Common Vision goal diagram using your example or the Tiger team:

1. Choose a decision point.
2. Review the options.
3. For more information, look up the options in the options table in your *Choose Your WoW!* book. If necessary, look up any options you need to know more about on the internet, in blogs or in books.
4. Select an option you believe will work best, given a team's context and life cycle choice.

Lesson 6: Tailoring Your Practices: Construction Phase

Description

As a DASM, you'll want to help your team tailor their way of working in every phase of your life cycle. This lesson covers the Construction phase, including lean concepts and tools that can help your team excel. You'll use the DA tool kit to improve a team's Construction phase processes.

Objectives

Describe the Construction phase and why it is important.

- Define Construction
- Identify process goals associated with the Construction phase

Discuss how to use the DA tool kit to tailor your way of working within a select phase according to context

- Rank and select process goals according to their relevance to the phase and the team's context
- Identify key practices for the team try using goal diagrams

Explain how to Eliminate Waste and Build Quality In (Lean principles)

- Identify the causes of waste and delays
- Describe how to minimize waste through value stream mapping
- Describe the push and pull methods of moving work
- Describe the Kanban approach to managing work in process
- Explain how to build and validate quality into the delivery process

Explain how to Deliver Value Quickly (Lean principle)

- Explain cost of delay
- Describe how to realize value
- Explain the importance of delivering incrementally
- Contrast MBI with MVP

Agenda

1. DA Construction Phase
2. DA Construction Process Goals
3. Agile Practices (Agile life cycle)
4. Lean Tips
    a) Deliver Value Quickly
    b) Visualize the Value Stream
    c) Deliver Incrementally
    d) Ensure Value by Building Quality In

e) Use Pull Systems and Kanban Boards

5. Choices in the Construction Phase

Lesson Notes

DA Construction Phases and Process Goals

The Construction phase is when we incrementally build a consumable solution.

This phase begins as soon as we complete the Inception phase. We have done the planning to get our team moving in the right direction. Now we're ready to start work.

The Construction phase ends when the solution is ready for release and advances to the Transition phase.

Agile Practices

Planning an Iteration

In agile, the main part of Construction is the iteration.

It all starts when the team gets together to plan the iteration.

Planning the iteration falls under the process goal Produce a Consumable Solution under Plan the Work.

Chapter 17 in your *Choose Your WoW!* book provides more information about each option and the associated risks.

Defining "Done"

During the Construction phase, the team collaborates to create a definition of "done", or DoD, which they will apply to every work item to determine when they are finished.

DoD falls under the process goal Accelerate Value Delivery under Verify Quality of Work.

You can look up the various strategies listed and the risks they entail in Chapter 19 of your *Choose Your WoW!* book.

Demonstrating an Iteration

At the end of an iteration, the team conducts an Iteration Review, which includes a Demo. The team demonstrates each work item to stakeholders to obtain feedback and determine whether it is ready to move forward to the Transition phase, where it will be deployed.

Demonstrating the iteration falls under the process goal Produce a Consumable Solution under Ensure Consumability.

You can look up what the options mean and the risks they entail in Chapter 17 in your *Choose Your WoW!* book.

Lean Tips: Deliver Value Quickly

Lean is an approach that produces value for customers quickly through a focus on reducing delays and eliminating waste, which results in increased quality and lower cost.

To deliver value quickly, deliver the smallest unit possible that will create value for the customer.

The key is to:

- Focus on time instead of cost.
- Focus on removing delays rather than on going fast.
- Achieve flow by working on smaller things and with people fully allocated to the work.
- Limit batch size.

Lean Value

In Lean, value is:

- What the customer considers important
- What the business invests in
- What is most useful to the customer upon release

Realizing Value

What does it mean to realize value? When is value realized?

“A product, feature or service has value if the customer considers it important at the time it is delivered. Value is realized when it is used.”

Lean Tip: Visualizing the Value Stream

To realize value, we need to look at the value stream the path from ideation of a product or feature to delivery to the customer. The construction phase is an important part of the value stream, during which we focus on building features that will have value for the customer.

Cost of Delay

What is the cost of delay?

The cost of delay is the lost revenue or opportunity caused by the delay between conceiving the idea and having customers realize value from it.

The longer the period between the two, the greater the loss in revenue."

Every day without realizing the value is a day without revenue from it. So the delay costs money, and the cost increases with each day. There is also the cost of more time and effort expended during this period.

Minimum Business Increment vs. Minimum Viable Product

Minimum Business Increments:

- Build the **smallest enhancement** to an **existing product**
- Investment for **revenue**

Minimum Viable Product:

- Take the smallest step to determine **viability** of a **new product** without a customer base
- Investment for **discovery**

Job Sequencing

Sequence jobs, not in order of priority, but in order of which MBI can realize the most value more quickly than the others.

MBIs are perfect for this because they are, by nature, deployable and consumable.

Lean Tip: Ensure Value by Building Quality In

Building quality in means continuously checking quality because delays in doing so will cost more in the short and long term.

"Build quality in" means:

- Continuous validation: test the work being done
- Continuous integration: test the dependencies
- Continuous deployment: test value of the work being done

Lean Tip: Eliminating Waste

Eliminating waste starts with looking at areas where waste tends to occur. For delivery teams, waste is most often manifested as delays. Let's look more closely at how to eliminate waste.

Sources of Waste

These are some of the most common sources of waste in our work:

- Miscommunication
- Building the wrong thing

- Building items of less importance
- Lost realization due to delays
- Aging information
- Relearning
- Handoffs and hand-backs
- Defects

Value Stream Mapping to Find the Cause

Value stream mapping helps teams look at the path followed to realize value and then to pinpoint problem areas, where there is waste or delay.

Note that you need to look beyond the construction phase and beyond your own part of the value stream.

Five Whys" analysis is a variant of "who, what, when, where, why" by Taiichi Ohno, creator of the Toyota Production System.

He suggested that continuing to ask "why" several times would get you insights into the root cause of things.

The "Five" is a "rule of thumb" to encourage you to keep digging. It doesn't have to be five questions exactly.

Lesson 7: Tailoring Your Practices: Transition Phase

Description

As DASMs, we deal with conflict in our jobs and among team members on an almost daily basis. In this lesson you'll learn how to use conflict to solve problems and to minimize conflict when it is unhealthy. You can lead your teams on their way to high performance using your conflict resolution skills.

Objectives

Describe the Transition phase and why it is important.

- Define Transition
- Identify process goals associated with the Transition phase

Discuss how to use the DA tool kit to tailor your way of working within a select phase according to context

- Rank and select process goals according to their relevance to the phase and the team's context
- Identify key practices for the team try using goal diagrams

Agenda

1. The Transition phase
2. Transition phase process goals
3. Choosing a process goal, a decision point and an option

Lesson Notes

Project Phases

Project-based life cycles—even agile and lean ones—go through phases.

It all starts with Inception when the team envisions and plans the project, doing just enough work to get organized and get going in the right direction. The team will initially form itself, then invest some time in initial requirements and architecture exploration, initial planning, aligning itself with the rest of the organization and securing funding for the rest of the project.

The process continues with Construction. The team produces a consumable solution with enough functionality to be valuable to stakeholders. During this phase, the team will be performing analysis, design, programming, testing and management activities every single day.

And finally, the process concludes with Transition. The team releases its solution into production. This includes both determining whether the solution is ready to be deployed and then actually deploying it.

Transition is sometimes referred to as a "release sprint (or iteration)" or a "deployment sprint," and if the team is struggling with quality, a "hardening sprint."

The aim of the Transition phase is to successfully release your solution into production. This includes determining whether you are ready to deploy the solution and then actually deploying it.

Transition Phase Process Goals

The Transition phase has two process goals:

- Ensure product readiness
- Deploy the solution

Ensure Production Readiness

The aim of the **Ensure Production Readiness** process goal is to determine whether you can safely deploy your solution into production. Remember, your team is producing a consumable solution—that is, usable **and** desirable **and** functional. Something that actually gets the job done.

Although your team should have produced a potentially consumable solution all the way through the Construction phase, this is your last chance to ensure the solution is consumable before deploying it to stakeholders. This goal is important because it reduces the risks associated with deployment by ensuring that the team is technically ready to deliver and that stakeholders are prepared to receive new functionality.

It's important to note that this goal reflects the realities faced by teams that are following project-based life cycles:

- Scrum-based Agile life cycle
- Kanban-based Lean life cycle

Teams following these life cycles tend to release into production every few months (or more) and have not yet completely automated their regression tests nor adopted the continuous integration/continuous deployment pipeline required to evolve into one of the two continuous delivery life cycles.

Deploy the Solutions

The aim of the **Deploy the Solution** process goal is to provide options for how to successfully release your solution into production. A typical disciplined agilist may react with "Well, why don't we just completely automate this?"

And they're right, they **should** fully automate deployment.

This process goal is important because:

- It captures several strategies for automating deployment.
- It provides several strategies for releasing your solution into production.
- It describes what needs to be performed to successfully release into production.

- It describes options for how you can ensure your release was, in fact, successful.

To effectively deploy your solution, you should consider several important questions:

- To what extent will you automate the deployment process?
- What strategy will you follow to release into production (this time)?
- What activities must you perform to release your solution?
- How will you validate that the release was successful

Lesson 8. Tailoring Your Practices: Ongoing

Description

As DASMs, we know the value of planning in helping our teams to be effective. This lesson will help you gain a better understanding of planning and show you effective methods for doing so. With planning knowledge and tools, you can deal with unexpected problems your teams may encounter on the journey toward Disciplined Agility.

Objectives

Describe the Ongoing phase and why it is important.

- Define Ongoing phase
- Identify process goals associated with the Ongoing phase

Discuss how to use the DA tool kit to tailor your way of working within a select phase according to context

- Rank and select process goals according to their relevance to the phase and the team's context
- Identify key practices for the team try using goal diagrams

Explain how to Learn Pragmatically (Lean principle)

- Define "standard work" and its use as a baseline for continuous improvement
- Explain the benefits of explicit workflow
- Describe how to use Kaizen loops and PDSA techniques for continuous improvement
- Define the options for cross-team learning: "community of practice" and "center of excellence"

Agenda

1. Understanding Ongoing Process Goals
2. Ongoing Agile practices
   a. Standard Work
   b. Explicit Workflow Policies
   c. Guided Continuous Improvement
3. How does an agile organization support cross-team learning?
   a. Communities of Practice
   b. Centers of Excellence

Lesson Notes

Ongoing Phase

The Ongoing phase includes those activities that occur continuously through the other three phases.

Ongoing Process Phase Goals

There are six process goals associated with the Ongoing phase:

- Grow team members
- Coordinate activities
- Evolve WoW
- Address risk
- Leverage and enhance existing infrastructure
- Govern delivery team

Grow Team Members

The Grow Team Members process goal captures options for providing opportunities for people to improve. This process goal is highly related to the People Management and Continuous Improvement process blades which focus on helping people at the organization level. There are several reasons why this goal is important:

1. People—and the way we work together—are key to success. Remember the agile value of "Individuals and interactions over processes and tools?"
2. Motivated people are effective people. In *Drive: The Surprising Truth About What Motivates Us* (2011), Daniel Pink argues people are motivated by autonomy, mastery and purpose. This process goal focuses on providing opportunities for people to master their craft.
3. Solution delivery is a team sport. Great teams are composed of people who want to work and improve together.

This ongoing process goal describes how we will support our team members in their personal and professional growth. To be effective, we need to consider three important questions:

- How will we help people improve their skill set?
- How will we provide feedback to team members to help them grow?
- How will we sustain the team over time to enable people to grow?

Coordinate Activities

The Coordinate Activities process goal provides options for coordinating both within a team and with other teams within our organization. There are several reasons why this goal is important:

- **Support effective collaboration**. It is rare to be completely autonomous because we often need to collaborate with others, hence the need to coordinate with one another.

This will help to reduce and hopefully eliminate several sources of waste, particularly wait time and rework.

- **Support autonomy**. In *Drive: The Surprising Truth About What Motivates Us (2011*), Daniel Pink argues that autonomy, mastery and purpose are what motivates people. One aim of this process goal is to suggest ways of working that enable both people and teams to work as autonomously as possible, yet still collaborate effectively with others as needed. Note that the Develop Common Vision process goal promotes the idea of teams with purpose and the Grow Team Members process goal provides opportunities for gaining mastery.
- **Working agreement within the team**. A team's working agreement describes how it will work together as well as with others. An important aspect of our team's working agreement is how we intend to coordinate our activities internally within our team.
- **Working agreement with other teams.** Similarly, indicating how others may interact with our team is also an important part of our team's working agreement. Having effective coordination strategies in place enables our team to collaborate effectively with others.

Evolve WoW

The Evolve Way of Working (WoW) process goal provides options for identifying and evolving how we will work together as a team. The focus of this goal is on the WoW for a team, the focus of Continuous Improvement is to support and enable teams to choose their WoW and to share learnings across the organization. There are several reasons why this goal is important:

- **Every team is unique and faces a unique situation**. Because people are unique, teams are therefore also unique. Every team faces a unique configuration of complexity factors including team size, geographic distribution, technical complexity, regulatory compliance, and other issues. The implication is that a team needs to tailor their WoW to address the situation that it faces.
- **We are constantly learning**. As individuals we learn every day - maybe we learn a new skill, something about the problem we face, something about how our colleagues work, something about our technical or organizational environment, or something else. These learnings will often motivate us to evolve the way that we work.
- **The other teams we collaborate with are evolving**. Very few agile teams are "whole" in practice. They must collaborate with others to achieve their mission. Because these other teams are evolving their WoW over time the implication is that the way that they interact with us will evolve too, something that we may be able to learn from.
- **Our environment is constantly evolving**. Our external environment is constantly changing, with our competitors evolving their offerings, the various levels of government introducing new legislation (including regulations that we need to comply with), new and evolving technical offerings in the marketplace, and world events in general. Our internal environment also evolves, with people joining and leaving our organization, our organizational structure evolving, and our IT ecosystem evolving as other teams release their solutions into production. Needless to say, we may need to evolve our WoW to reflect these changes.

- **The team needs somewhere to work**. On some teams, everyone is dispersed and working from home; the rest will need space for some or all team members.
- **The team needs sufficient tooling**. The team needs access to physical and digital tools so we can do our work.
- **These strategies are applicable to a wide range of teams, not just solution delivery teams**. We've applied these strategies with leadership teams, marketing teams, finance teams, enterprise architecture teams, data management teams, and many others. Having said that, the focus is on how solution delivery teams can choose their WoW. Although this process goal applies to all of those teams the rest of the goals may not. Each of these domains (marketing, leadership, etc.) requires domain-specific advice.

Leverage and Enhance Existing Infrastructure

The Leverage and Enhance Existing Infrastructure process goal provides options for reusing and hopefully improving existing assets within our organization. These assets may include guidance, functionality, data and even process-related materials. This process goal is related to the Improve Quality process goal, which focuses on strategies to pay down technical debt in such assets and the Reuse Engineering process blade, which focuses on the reuse of existing assets. There are several reasons why this goal is important:

- A lot of good work has occurred before us. There is a wide range of assets within our organization that our team can leverage. Sometimes we will discover that we need to first evolve the existing asset so that it meets our needs—which often proves faster and less expensive than building it from scratch.
- We can reduce overall technical debt. The unfortunate reality is that many organizations struggle under significant technical debt loads. By choosing to reuse existing assets, and investing in paying down some of the technical debt that we run into when doing so, we'll slowly dig our way out of the technical debt trap that we find ourselves in.
- We can provide greater value quicker. Increased reuse enables us to focus on implementing new functionality to delight our customers instead of just reinventing what we're already offering them.

Addressing Risk

Disciplined Agile Delivery (DAD) has several risk mitigation strategies built in:

- The Address Risk process goal. Originally DAD had two risk-focused process goals, this one and Identify Initial Risks, but due to the significant overlap between the two we decided to simplify the framework by combining them into a single process goal.
- Support for a risk-value life cycle. DAD promotes a risk-value life cycle approach where we recommend that risk be considered when prioritizing work in addition to stakeholder value---many agile methods focus just on value to their detriment. The risk-value profile for a DAD team shows how DAD teams address a lot of risk very early in the life cycle via addressing the Stakeholder Vision and Proven Architecture milestones.
- Support for ordered ways of working (WoW). As you've seen here, within each process goal diagram many of the decision points have ordered option/choice lists. This makes

the lower-risk ways of working explicit because the more effective options tend to be towards the top of the lists.

- The Address Risk process goal provides options for how we will approach risk within our team. Although the project management community prefers the term "manage risk" rather than "address risk," not surprisingly, we find that the word manage comes with too much baggage---managing risk leaves the door open to needless bureaucracy, whereas addressing risk motivates us to focus on dealing with the challenges that we face.

There are several reasons why the Address Risk goal is important:

- We face many risks. Many risks are addressed within the team, but some risks we'll need help from outside the team to address. Disciplined teams make risks transparent, making it easier for them to garner the help they need.
- Understanding the level of risk is a critical decision factor for moving forward. Two of the questions that we should ask at the Stakeholder Vision milestone is whether the team understands the risks that it faces and if so, does it have a viable strategy to respond to them? Similarly, any go-forward decision made during Construction should take the current level of risk faced by the team into account.
- Reducing risk increases our chance of success. Enough said.
- It's usually better to deal with risks early (in other words, shift risk mitigation left). Risks tend to grow (but not always). If a risk proves to be a problem, it's better to know that early when we still have time and budget to fix it, or if the risk proves insurmountable, it's better to cancel or go in a different direction and thereby not waste time and money.

Process Goal: Govern Delivery Team

The Govern Delivery Team process goal provides options for governing agile and lean delivery teams. Governance establishes chains of responsibility, authority and communication in support of the overall enterprise's goals and strategy. It also establishes measurements, policies, standards and control mechanisms to enable people to carry out their roles and responsibilities effectively. You do this by balancing risk versus return on investment (ROI), setting in place effective processes and practices, defining the direction and goals for a team, and defining the roles that people play within a team.

The Govern Delivery Team process goal is supported by both the IT Governance and the Control process blades. There are several reasons why this goal is important:

- We are going to be governed. Many in the agile community believe that governance is a swear word, likely because they've had negative experiences when traditional governance strategies [COBIT] were applied to agile teams. Although we understand this attitude, we find it to be counterproductive because someone is going to govern our teams, like it or not. Someone will govern the finances, they will govern the quality, and they will govern what we produce---just to name a few issues.
- We deserve to be governed well. Our team is made up of intellectual workers, people who are smart and skilled at their jobs. They respond well to leadership—deciding for

themselves what to do—and not very well to management—being told what to do. As a result, effective governance is based on motivation and enablement, not command and control.

- Governance is context sensitive. A traditional waterfall team is governed in a very different way than an agile project team, which in turn is governed in a different way than a team following the Continuous Delivery: Lean life cycle. Teams that are less experienced or facing significant risk will require more governance than those that are not.
- Our team is part of a larger organization, and we need to leverage that. Our organization is a complex adaptive system (CAS), a collection of teams working together in an adaptable and constantly changing manner. And we've been doing this for a very long time, in some cases decades and even centuries. We have a wealth of experience, skills, intellectual property and physical assets available to us that we can use in new ways to delight our customers. The point is that we don't need to work on our own, and in fact we likely can't—given the complexity that we face, and we certainly don't need to build everything from scratch.
- Effective governance enables collaboration. Given that our organization is a CAS, the leaders who are governing us must focus on helping our teams to be successful. This includes ensuring that we have the resources we require to accomplish our mission and ensure that we're collaborating effectively with the other teams that we need help from.
- We have responsibilities to external stakeholders. Our team has stakeholders to whom we are beholden, and one aspect of governance is to ensure that our team meets their needs. These stakeholders include auditors who need to ensure that we're compliant to any appropriate regulations or internal processes, legal professionals who help us to address appropriate legal issues, and company shareholders (citizens when we work for a government agency or nonprofit) whom we effectively work for.

Our focus in this process goal is on delivery/development governance, but as you can imagine other governance categories have an effect on it. For example, solution delivery teams will still need to be governed in their use of data, guided by user experience (UX) standards, and funded in accordance to finance guidelines, while fulfilling roles supported by people (management) governance.

In this process goal we use several terms that we want to define now:

- *Leadership. People within our organization, often senior management, who are leaders.*
- *Enterprise groups. Teams responsible for information technology (IT) or enterprise-level activities such as enterprise architects, finance, security and procurement.*
- *Enterprise professionals. People such as enterprise architects, finance professionals, security engineers and procurement specialists.*

Lean Practices for the Ongoing Phase

Lean Practice: Standard Work

The first of these practices is standard work.

Standard work is the work process agreed to by the team doing it. This is the standard that the team agrees is the best way to do something.

It is defined by explicitly expressing the team's workflow.

Standard work acts as a backdrop for doing our job and immediately seeing how well we are doing. It gives us an ideal to strive for.

Standard work is not static. When a better way is found, the standard should be updated. That's one of the purposes of a standard: to set a baseline from which to improve, while adhering to its original purpose.

Test out new ideas right away. Any that prove their worth in the real world should become part of the new standard.

Standard work should create tension. It provides the means for doing our job and immediately seeing how well we are doing. One of its purposes is to create tension between what we're *currently* doing and what we're *supposed* to be doing. This tension promotes learning. And innovation

Standard work reinforces innovation and makes improvement possible. It is essential for continuous improvement—moving from one standard to a better standard without slipping back.

Standard work articulates who, what, when and where work is done. It focuses on content, sequence, timing and outcomes needed. As mentioned previously, it is intended as a basis for improvement.

It is not a prescription or record of what's to be done. Rather, it is an identification of steps or activities of the best, currently known approach to achieving a solution, within the boundaries established by the organization; it entails visibility (visual controls) and discipline.

Standard work is *not* static, and when a better way is found the procedure is updated. To continually improve, one must understand the purpose of the standard, and improve the standard, while adhering to its purpose. As you perform standard work, you will find things you don't like and you will think of one improvement after another. You should implement these ideas right away and make this improved description the new standard. Embrace those practices that prove themselves in real life

Closely related to standard work is explicit workflow.

Explicit workflow policies act as guidelines as to how a team should perform their work. In their simplest form, they're checklists used to define when the team can pull work from one process step to another. They govern the team's process and serve as boundaries for the execution of their work.

Explicit workflow policies:

- Go beyond enabling everyone to understand what other team members are doing
- Make visible any gaps between what we've agreed to and what actually happen

When we can't follow the explicit workflow, the difference between the two assists learning.

Lean Practice: Continuous Improvement

Kaizen is a Japanese term meaning "improvement."

A Kaizen loop is an approach where a team experiments with a small change in their way of working, adopting the change if it works in their given context and abandoning it if it doesn't.

Some students heard this referred to as PDCA—Plan-Do-Check-Act. Deming went back and forth between PDCA and PDSA. He finally settled on PDSA.

Continuous improvement is the act of applying a series of Kaizen loops to improve your way of working over time.

Guided continuous improvement extends the Kaizen loop strategy to use proven guidance to help teams identify techniques that are likely to work in their context. This increases the percentage of successful experiments and thereby increases the overall rate of process improvement.

Here's how that works.

First, an organization identifies a problem that needs fixing. It does some root cause analysis to figure out what the problem actually is.

Next, the organization identifies a potential improvement—such as a new practice or strategy—that they want to experiment with to see how well it works in their situation.

Then they try it out. They implement the change.

After a brief period, they'll assess whether the change has been effective. They'll measure it against clear outcomes.

If the new way of working is effective, they adopt it; if it isn't, they abandon it, for now at least.

But they're not done. At that point they share what they've learned with others—other members of their team and other teams throughout their organization.

Then start the process over. It is continuous improvement, after all.

Disciplined Agile enables you to increase your rate of process improvement by helping you to identify strategies that are more likely to succeed given your situation.

If we get better at this, we will succeed more often, and we will improve faster. We can do this if:

- We have access to an experienced agile coach, but they're expensive and hard to find.
- We have access to a process knowledgebase, like the Disciplined Agile tool kit.

Some experiments fail. You'll learn something, but it's still a failure.

Failing fast is fine but succeeding early is better.

Communities of Practice

One approach to remedying this situation is to establish communities of practice.

In many ways a community of practice is a collection of people who share a craft or profession who have banded together to "learn" from each other to develop themselves and often even the organization. Communities of practice are also known as communities of excellence, interest leagues—not to be confused with The Justice League—or even guilds. We've seen communities of practice focused on agile software development, testing, architecture, management, coaching, business analysis, DevOps and many more.

Communities of practice form on a volunteer basis, although in some organizations the community of practice lead may be a budgeted position.

Establishing and participating in communities of practice provides a conduit to leverage knowledge across the organization. A community of practice forms when people recognize the need to help one another learn a topic. This falls into three major areas:

- To share techniques with one another. Community of practice members will often share techniques with one another through face-to-face chatting in discussion forums, and practitioner presentations—such as a lunch-time presentation.
- To support one another's learning. The goal diagram also captures several strategies that community of practice members may choose to employ to support one another—in particular coaching and mentoring. Although a Center of Excellence within your organization may officially be responsible for coaching and mentoring—and for most of the potential activities captured by the goal diagram—you will often see members of a community of practice doing this as well in an informal manner. This is particularly true when no center of excellence exists for a given topic.
- To capture techniques. Some communities of practice will choose to start capturing the techniques and strategies that they learn and share with one another, typically in an informal manner using either a wiki or documentation repository such as Microsoft SharePoint.

Communities of practice are typically formed through one of two strategies:

The first, and most common, is in an ad hoc manner when practitioners realize that they have a common interest in learning and decide to support one another in doing so. The group will typically start meeting physically, perhaps in your cafeteria or cafe, or perhaps one of your meeting rooms. When it becomes clear a community of practice is needed, the next step is to start putting internal support mechanisms in place such as discussion forums.

The second strategy is when an existing Center of Excellence around a topic decides to support a community of practice around the topic that they are tasked to support within the organization.

Structures of communities of practice tend to be fairly fluid, with a leadership team initially composed of the most experienced members who work together to organize and support the

overall effort. The community of practice leader—if there is one—typically emerges from within, although sometimes that person is assigned at first if the team has been created by a community of excellence

Membership in a community of practice is voluntary. Members will come and go as they see fit and participate as much as they are willing to or have the time to

Centers of Excellence

Closely related to communities of practice are centers of excellence.

A center of excellence—sometimes called a center for excellence—is a group of people with specialized skills and expertise whose job is to provide leadership and purposely disseminate that knowledge within an organization. Centers of excellence should not be confused with communities of practice.

In the last few years we've seen agile centers of excellence, testing centers of excellence, DevOps centers of excellence, and architecture centers of excellence created within organizations to help their continuous improvement efforts.

A Center of Excellence is typically formed to address a skills or knowledge deficit within an organization. The members of a center of excellence are typically coaches. Center or excellence coaches will be involved with many of the activities of Continuous Improvement

- **Identify techniques**. Coaches will work with one another, and with the people they are coaching, to identify potential techniques (practices, strategies, principles) that they can help people to adopt to improve the way that they work.
- **Share techniques**. Coaches will help practitioners to share techniques that they find effective with one another. Helping to build a learning organization is the primary way for center of excellence coaches to scale their efforts—better yet, work their way out of a job.
- **Capture techniques**. Center or excellence coaches work with practitioners to capture viable techniques to build organizational memory around their processes and strategies.
- **Support teams**. The primary mission for center of excellence coaches is to support individual and team learning.
- **Organize communities of practice**. Very often a center of excellence will initiate—or at least support—the initiation of one or more communities of practice to aid their educational efforts.
- **Governing improvement**. A center of excellence will often collect and track a collection of metrics to help them both govern and justify your organization's investment in the center of excellence.

A center of excellence will be formed by identifying people who have the skills and knowledge, the ability to coach and likely train people, and who have the drive to continue learning on their own.

Review the information from the following table with your learners:

Strategy: Hire from within

Advantages:

- The person is known and respected within the organization
- The person knows how to navigate the environment

Disadvantages:

- May not have experienced the topic outside your organization; may struggle with "that's the way it's done here" issues
- May be wrapped up in the activities of their existing position
- Often new to being a coach

Strategy: **Hire new employees**

Advantages:

- Brings a new viewpoint and fresh experiences into the organization
- Often has coaching experience

Disadvantages:

- Can be very difficult to find
- Can be difficult to get rid of a full-time employee if they don't work out

Strategy: **Hire consultants/contractors**

Advantages:

- Easier to find candidates because many experts choose to become consultants
- There may be an opportunity to hire the person as a full-time employee once you've tried them out

Disadvantages:

- Difficult to identify this type of person if you don't already have people experienced in the center of excellence to help you identify viable candidates

Lesson 9. Influence Outside the Team

Description

Since Disciplined Agile is based on both agile and lean, a DASM should have a firm grasp of lean principles and how they function at a systems level. In this lesson, you'll gain a deeper understanding of how lean principles and tools impact the organization, which will inform your work with your team.

Objectives

Describe how Lean takes a system view rather than a team view

- Contrast Lean aspects of knowledge work with work in the real world, including sources of waste and delay
- Describe aspects of regular work that affect quality and efficiency, including sources of waste and ways to improve

Recognize when to be resilient

- Describe how resiliency supports lean thinking
- Explain when to build workflow according to resiliency outcomes

Agenda

1. What is Lean?
   a. Lean principles
   b. Lean Incorporates Systems Thinking
2. Lean Knowledge Work vs. Lean Factory Work
   a. Lean Resiliency

Lesson Notes

Looking Beyond the Team

Key concept

Sooner or later, anyone who's serious about improving their team's performance will come up against the realization that they'll have to look beyond their team.

What is Lean?

As we move toward organization-level thinking, we move into the realm of lean. **The term "lean"** was coined to describe Toyota's business during the late 1980s by a research team from MIT's International Motor Vehicle Program. Three members of that team—James P. Womack,

Daniel T. Jones and Daniel Roos—first defined the term in a book about their project, *The Machine that Changed the World*.

Some key points: 

- Henry Ford revolutionized mass production, making it possible for minimally trained workers to assemble cars quickly and efficiently.
- Toyota eliminated much of the waste inherent in Ford's system by making smaller batches of parts to be used as needed instead of stockpiling larger quantities.
- Toyota also empowered its workers to improve the process and stop the line when issues and errors occurred.

Key Lean Concepts

- Lean has been expanded to virtually all industries.
- Lean is based on systems thinking
- Lean involves continuous learning.
- Lean is not about building cars but about building great organizations via learning.

Wastes in Lean

The virtual world is not the same as the physical (or real) world, but many of the same principles apply.

- **Defects**. Defects are defects and considered waste because they provide no value to the customer.
- **Overproduction**. In manufacturing, this is making more things than you need. In knowledge work, this is building features that are not needed. It adds complexity and delays value delivery.
- **Inventory**. In manufacturing, this is stocks of goods and raw materials. In knowledge work, this is unfinished work. Unfinished work has at least two significant wastes/risks to it. First, when it just sits there, dependencies between it and other work may be forgotten. When other things change, these changes may not be reflected in this unfinished work. Also, bugs may not be detected until later, making them harder to fix.
- **Waiting**. In software, it's worth including all types of delays: workflow and feedback. These delays cause unplanned work.
- **Transportation**. In manufacturing, transportation means moving work-in-process around. In knowledge work, it's moving work to different people. Handoffs always lose information, hand-backs lose time.
- **Motion**. In manufacturing, motion is moving people. Transportation is about moving inventory. In knowledge work, it's about switching projects, which can happen easily (just a few strokes of a keyboard) but can switch the mind of the developer (multitasking) and become very costly.
- **Excess Processing**. With big batches, we do a lot of analysis before we need to—or a lot of development. Incremental delivery can reduce excess processing.

- **Nonutilized Talent**. Using less-skilled people than those available lowers work quality. This also includes people knowing things that others don't, but not having a way to share that knowledge, which forces relearning.

Manufacturing vs. Knowledge Work

Manufacturing	Knowledge Work
Defects	Defects
Overproduction	Creating unnecessary features *(adds complexity, delays value)*
Inventory	Partially done work *(hides risk of errors)*
Waiting	Delays *(causes unplanned work)*
Transportation	Handoffs *(loss of information)* and hand backs *(loss of time)*
Motion	Motion *(shifting from value stream to value stream)*
Excess Processing	Doing more work than needed
Nonutilized Talent	Lower quality work *(requiring relearning)*

Lean Principles – Software Development and Knowledge Work

These are the principles of Lean as they apply to software development and knowledge work:

- Build Quality in
- Eliminate Waste
- Learn Pragmatically
- Keep Options Open
- Deliver Value Quickly
- Respect People
- Optimize the Whole
- Build in Resilience

Lean Principle– Build Quality In

Key Concepts:

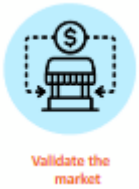
- Lean teams build quality into everything they do
- Lean teams adopt practices that ensure that each element of their solution, at every increment, meets appropriate quality standards.

Building quality in means:

- Continuous validation (test work being done)
- Continuous integration (test dependencies)
- Continuous deployment (test value)

Four Stages of Lean Validation Process

Four distinct stages make up the **lean validation process**. Only once you've passed all four, can you be confident that your product idea is worth developing.

Validate the problem. Is this a problem worth solving? If users don't think this is a major problem, your solution won't be appealing.	
Validate the market. Some users might agree that this is a problem worth solving. But are there enough of them to make up a market for your product?	
Validate the product. The problem might exist, but does your product actually solve it?	
Validate willingness to pay. There might be market demand and a great product. But will people be willing to reach into their wallets and pay for it?	

Lean Principle – Eliminate Waste

In lean thinking, any activity that doesn't directly add value to the finished product is waste.

To reduce waste, it's critical to allow teams to self-organize and operate in a way that reflects the work they're trying to accomplish.

The most common causes of waste within a team are:

- Multitasking
- Rework
- Building work of lesser value
- Relearning
- Miscommunication
- Errors and poor quality

Lean Principle – Learn Pragmatically

Key Concepts:

Lean teams continuously strive for perfection. Experimentation and reflection are the driving forces; they continuously run practical experiments and pragmatically apply what they learn—and freely share this new knowledge with other teams.

Lean Principle – Keep Options Open

Key Concepts:

It's not necessary to start projects by defining a complete specification.

Delay detailed discussions of future features—and decisions about them—until the last responsible moment, when more information will enable them to make a better decision.

Lean Principle – Deliver Value Quickly

Limiting the work of a team to its capacity enables a reliable and repeatable flow of work. Constraining a team to regularly delivering potentially shippable solutions motivates them to stay focused on continuously adding value.

Keys to Delivering Value Quickly

- Focus on time instead of cost.
- Focus on removing delays rather than going fast.
- Achieve flow by working on smaller things and with people fully allocated to the work.
- Limit batch size.

Lean Principle – Importance of Incremental Delivery

Key Concepts:

Solution development has specific timing issues. It is not realistic to build an entire product and then release it. Instead, build small, functional features so that they can be tested and released. In that way, you can deliver value quickly.

Lean Principle – Respect People

Key Concept:

Lean teams respond to people promptly, listen attentively, hear their opinions, and do not dismiss them even when they are different to their own. Lean recognizes that teams work best when they are empowered to make decisions where the work is.

Keys to Respecting People:

- Culture is the context in which all change must happen.
- Recognize all who consume your work as customers.
- Attend to people's needs and wants.
- Focus on understanding others' views and perspectives.

- Build partnerships based on trust.
- Create an environment of mutual influence.

Lean Principle – Optimizing the Whole

Key Concepts:

A lean organization seeks to optimize the whole value stream, not just individual functions or teams.

It understands that high-level business processes often cross multiple systems and teams and seeks to optimize the entire process, not just the work of a single team.

Which Part of an Airplane is Responsible for Flight?

Key Concepts

You need an engine for power, wings for lift, controls to direct it, a place for the pilot, a navigation system and many other things—no single part is responsible for flight. They're all responsible. Improving just one of them may actually reduce the plane's ability to fly.

You have to think of an airplane as a system.

This illustrates a couple of key points:

- Systems work as a whole; improving one part may actually degrade the performance of the whole.
- Local optimizations don't always make system-wide improvements
- It's more important to see how parts of a system fit together than to focus on individual parts.
- It's the same with any complex system. If you're trying to improve—or even influence—the system, it's much more important to see how all the parts work together than it is to focus on the individual parts.

The System is the Source of Most Problems

Key Concepts:

Poor systems will defeat almost all but the greatest of people.

If we trust people, we don't need to work on them; we need to improve the system.

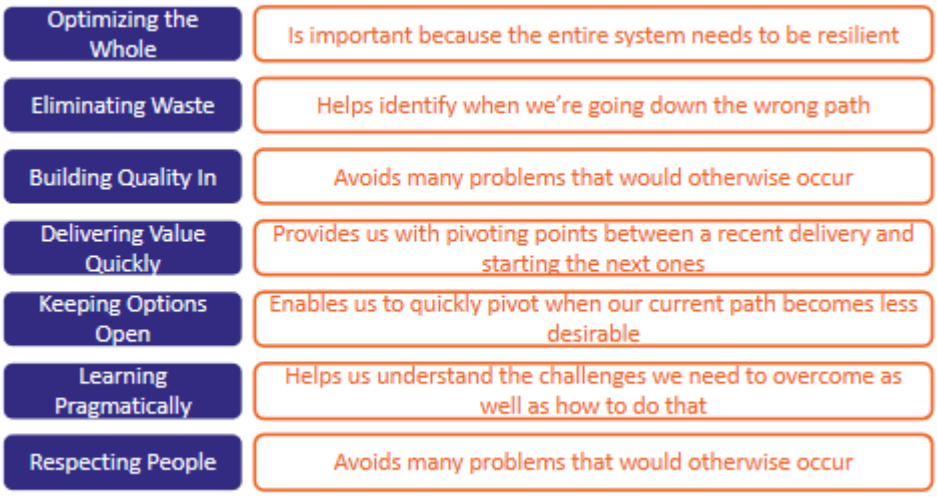
The role of management is to create great systems so that people can work autonomously to achieve the vision of leadership.

Lean Principle – Build in Resilience

Key Concept:

Lean teams constantly strive to build in the capability to survive, adapt and sustain the business in the face of change.

Lean Principles Enhance Resilience



Lean – Focuses on Realizing Value

Key Concepts:

Lean focuses on realizing value—both for the organization and for the customer.

This simple mantra can be used to drive the realization of value:

- Work on fewer things.
- Work on smaller things.
- Work more efficiently.
- Create better workflow.

How Do You Define Value?

Key Concepts:

- Value must be defined by the customer, but business stakeholders decide which value to go after.
- We must also attend to who our customers are. We're not going after all of them—we're going after those that our company is going after.

An example:

Southwest Airlines had a customer who always wrote complaint letters after a flight. She didn't like the boarding process, she didn't like that no meals were served, and the litany went on. She ended her letters by saying that she could no longer fly Southwest Airlines unless they made changes. Responding to her eventually got bumped up to the late Herb Kelleher—then the CEO of Southwest Airlines.

His response: "Dear X: We will miss you."

When Do Customers Know What They Want?

The best answer I've received is: "When I show them what they don't want."

Then the question becomes, "When do I want to show them what they don't want?"

The answer: As soon as possible.

Tell a story that makes this point.

Consider a time you were talking to a client or overheard someone talking to someone conveying requirements. When did clear requirements evidence themselves? Also, how many times do customers talk about their preconceived solutions instead of what their real problem is. When is getting quick feedback about what customers need a good thing? Quick delivery is not just about value being delivered quickly, but about enabling quick feedback to make sure what you are delivering is of good value.

An example from Al Shalloway from PMI:

> *During one of my first consulting gigs in the 1980s, I was creating sales-order software for a client. I got the requirements, implemented them, and was asked "What about deleting orders?"*
>
> *That wasn't in the requirements, so I responded that I wasn't told it needed to do that. The person said everybody knew it needed that.*
>
> *So, I put it in.*

This, of course, happened again; I learned to ask, "Is there anything else?" This would always get the response "no," which of course, didn't work either.

I started asking "Are you sure?" This was, of course, always answered with "yes."

Only when I gave up hope of getting all the requirements did I start acting differently—embracing change.

Types of Business Value

Key Concepts:

It's important to realize that we can also create value for the business even if that value is not released to the customer.

In knowledge work, creating value for the business includes the following:

- Value delivered to the customer
- Discovering what is of value
- Discovering how to build it
- Building it
- Preparing for consumption
- Improving our own internal methods
- Mitigating risk
- Learning something new

What is Role Management?

Key Concepts:

We have teams that are self-organizing and that choose their own ways of working. Across the organization, these can be dramatically different.

What, then, is the role of management?

Managers are no less committed to the work than any member of the team.

Their role is to create the environment for teams to be awesome.

Part of this entails:

- Listening and watching for the needs of the team and remove problems.
- Communicating with other managers to provide the bigger picture.