Agile for Non-Software Teams

Ch 1: How Your Way of Working Matters

You Can Reach Your Goal Multiple Ways

- Different options optimize for different (maybe even conflicting) goals.
- There may not be an obvious best choice.

The Potential of an Agile Way of Thinking

- Work has changed in the last 100 years – science/tech advances, business model evolution, societal norm shifts.
- In the mid-1990s, there was a shift toward cross-functional teams, less process, closer ties to customers, delivering results and adapting to change more frequently.
- Reasons Agile adoption typically fails
    - Retaining the same strategic and managerial choices as before
    - Top-down mandate of rigid process without considering the context
    - Lack of leadership support
- Companies think Agile is something you install to make people work faster.

Articulate Your Motivation Around Agile

- Agile is a means to an end. What is that? Why do you want to change? Why do you think Agile will help?
- You’re probably in for a bad time if your motivations are… “We can always improve” or “Everyone else is doing Agile, so we should too.”

Ch 2: Consider How Agile Differs from Your Current Way of Working

---

Tactics

- How does your team produce results?
    - **Process** – sequence of steps
    - **Procedures** – actions or sequences done at certain times
    - **Roles** – who does what
    - **Meetings** – who interacts and when
    - **Artifacts** – things produced because work was done
    - **Tools** – things that help the team function

- Most Agile implementations focus too much on tactics, and spend less time on the mindset and purpose.

Choice-Making

- Principles give rise to tactics.
- Teams are the primary working unit in Agile.
- Geoff's opinion:
  - "In terms of process, Agile teams obsess over finishing what they start so they may deliver value without delay."
  - This doesn't sound right. Finish what makes sense, and stop doing what *doesn't make sense*.
- Things Agile teams do…
  - Congregate to plan, review, and reflect
  - Have meetings that are collaborative and psychologically safe
  - Be transparent about their work, visualizing it in ways that let them control the flow
  - Execute work with a balance of collaboration, individual work, and hand-offs
  - Have autonomy over how work gets done, self-organizing to maximize delivery, learning, and team growth
  - Be lead by people who are supportive enablers (not taskmasters)
  - Connect their work to the org through leaders and stakeholders

Values

- Why does your team exist? What difference will you make?
- Common topics to discuss: objectives, success criteria, constraints, other parameters (i.e., a team charter)
- **Typical** business values, which can be important, but aren't the only set of values…
  - Getting deliverables right the first time
  - Following industry standards every time
  - Delivering results by a certain time and/or within a certain budget
- A different set of choices…
  - Delivering early and often – don't wait for the entire thing to be done
  - Adaptation – start small and iterate to respond to change/feedback
  - Customer collaboration – teams and customers work together
    - Geoff's note: Note the power dynamic here. There's a shift from *we know what's best for the customer* to *let's work with the customer to learn more about what they want and, while doing so, give them ideas to help them ideate.*
  - Putting people first – not everyone needs/wants the same rigid process; think "Platinum Rule"
- "This holistic value system replaces contractual, transactional, vendor-buyer, and producer-consumer relationships between two mistrusting sides with relationships that yield outcomes of value to all partners."

Beliefs

- Underlying this way of thinking is *choice*. You can choose to **work adaptively** or choose to **maximize predictability**. Neither is inherently incorrect.
- People
  - "Competent, motivated, trusted, and supported people will do good work."
  - However, people make mistakes, which is why we work together.
- The customer
  - Sometimes they're not always right.
  - Be effective first, and efficient second.
- The work – "If the work is complex or complicated, emergence (or evolution) is a more appropriate response than planning."
- "For any group of people to succeed together, their actions must align with a shared set of beliefs and values, whether it's their natural mindset or chosen for the situation at hand."

Forms of Agility

- **agile** – ability to change what's delivered and how it's done
- **agile culture** – an org that tolerates or welcomes ambiguity
- **Agile** – philosophies from the Manifesto for Agile Software Development
- **Scrum** – an common instance of an Agile framework
- **Lean** – ancestor to Agile that focuses on producing a well-understood solution while optimizing for quality, speed, and minimal waste

- **Kanban** --related to Lean; focuses on flow of knowledge and service work, minimizing WIP, and visualizing work

Ch 3: Listen to Your Team's Concerns

- Isn’t Agile Just for IT?
- We Do Operational Work, Not Product Development
- We Already Follow Industry Best Practices
- Has Agile Been Tried Before for My Kind of Work?
- We Can’t Change Our Methods Because of Regulatory Compliance Requirements
- What If Agile Applies to Only a Portion of Our Work?
- Some of the Agile Values Will Not Be Welcome Here
- We Are Not Empowered to Make Decisions
- If We Feel We’re Already Working in an Agile Way, Will We Have to Change Anything?
- What If There Are Practices We Can’t Do, or Can’t Find Equivalents For?
- Should We Use the Same Process as Everybody in the Company?
- We’re Already Too Busy; We Don’t Need More Meetings
- Agile Seems to Have a High Administration Overhead
- What If It Doesn’t Work Out?

Change can be difficult. Focus on the Agile approach to choice-making, rather than the specific processes you’ll use. This chapter lists common concerns about changing to something more Agile.

Isn’t Agile Just for IT?

It gained popularity in IT, but the values, beliefs, and principles of Agility are applicable elsewhere.

We Do Operational Work, Not Product Development

“Agile is ideal for non-routine work that involves complexity, change, and ambiguity and that it benefits from team collaboration.”

We Already Follow Industry Best Practices

As the world evolves, business needs to as well. If the best practices aren’t getting you the results you need, perhaps you should change.

Has Agile Been Tried Before for My Kind of Work?

Most likely yes (e.g., HR, design, marketing). Resist the temptation to copy/paste what others have done, though; learn why it was successful first.

We Can’t Change Our Methods Because of Regulatory Compliance Requirements

“Compliance requirements rarely force organizations to work a single way. Their purpose is to maximize safety and confidence and also to reduce certain risks; organizations accomplish that by proving that their decision-making execution protocols are reliable and safe, and by demonstrating that they actually follow them. There is nothing inherent to Agile that makes it non-compliant.”

What If Agile Applies to Only a Portion of Our Work?

Apply it where it makes sense; start with small improvements first.

Some of the Agile Values Will Not Be Welcome Here

“If you feel some of the Agile values would not be welcome in your organization, try and figure out why.”

We Are Not Empowered to Make Decisions

These changes take time; look for several small wins to build momentum toward change.

If We Feel We're Already Working in an Agile Way, Will We Have to Change Anything?

If you're already adapting to changing requests, regularly finishing deliverables together, and engaging in healthy communication with stakeholders, you may be well on your way.

What If There Are Practices We Can't Do, or Can't Find Equivalents For?

Don't adopt things that don't help you. Meet people where they are. Think JENTM (just enough, not too much).

Should We Use the Same Process as Everybody in the Company?

This isn't likely to be successful unless everybody works in the same way already. An Agile org is about having "a culture of customer and outcome orientation, cross-team collaboration, value delivery, and adaptation."

"Many people worry that if teams don't use similar methods, it will be hard to manage the overall organization to move people between teams." The main focus of transformation is empowerment and habit-building; don't create rigid silos.

We're Already Too Busy; We Don't Need More Meetings

Agile isn't about having meetings; it's about being stronger together and collaborating to drive value and reduce waste. Don't meet just to meet; there should be value in the meetings.

Agile Seems to Have a High Administration Overhead

Every way of working has pros and cons. Agile focuses on **flexibility, reliability, transparency, and collaboration**.

What If It Doesn't Work Out?

Maybe it won't, but you should learn something from the effort.

"It's okay to decide that Agile is not for you if you are really convinced that its mindset is not what you need to succeed. It's *not okay* to write it off because of concerns with practices such as 'we don't have anyone who could be a product owner,' 'we can't do automated testing,' or 'it's too many meetings.'"

Ch 4: Determine Where You'll Try Agile for the First Time

- Work on adopting Agile gradually, iteratively, in a series of **experiments**. Framing things as experiments allows you to abandon the idea of presupposed success (you can't really know if it will succeed).
- Questions for understanding the work
  - What products, services, and solutions do you provide?
  - What are the other major responsibilities or activities?
  - What work goes into the answers to the first two questions?
- "If your people only do a piece of a bigger product, and the piece doesn't deliver customer or business value on its own, doing Agile strictly within your team will have very limited business effect."
- Typical streams of work
  - **Development** – new offerings, enhancements to existing offerings
  - **Production** – making things in a repeatable manner
  - **Business as Usual** – operational or mission-critical work required for ongoing operation
  - **Support** – dealing with issues when using your product
- If some of these come up as values, Agile may not be right for you.
  - Get it right the first time.
  - Deliver on time and on budget.
  - Adopt/establish standard processes.
  - Nail down choices early, because changing them later is prohibitive.
  - Minimize errors by supervising staff.

Chartering

Here are some things to consider when getting alignment around the work the team will be doing.

Question	Description	Example
Who are your customers?	Who benefits from your work? Who do you support? Be specific or use personas.	Hiring managers, candidates we speak to
What is the value of your products?	How do you make a difference? Why is it worth the effort? This is like a mission statement.	We help hiring managers invest the minimal amount of time filling a position with someone great.
What does success look like?	Look for the human effects (not products or features) for your team, your customers, your company.	We don't lose good candidates through our own actions.
How are you constrained?	What limits your work? How do external factors affect your ability to plan, execute, and finish work?	Our understanding of the departmental work is not enough to answer candidates' questions.
What values maximize the chance of success?	What are the top values that guide choices? What's non-negotiable? Beware of things handed down from management (e.g., maximizing efficiency) that likely won't change your way of working.	Close collaboration with hiring managers and candidates
What are your assumptions?	Ideas: characteristics of the people doing the work, complexity of the work, expectations from others, business landscape, kinds of changes you'll have to respond to	• We've been trying to hire the best people, but we really need to hire the *rig ht* people. • The first 90 days really matters.

Ch 5: Understand That It Will Be a Journey, and It Can Go Wrong

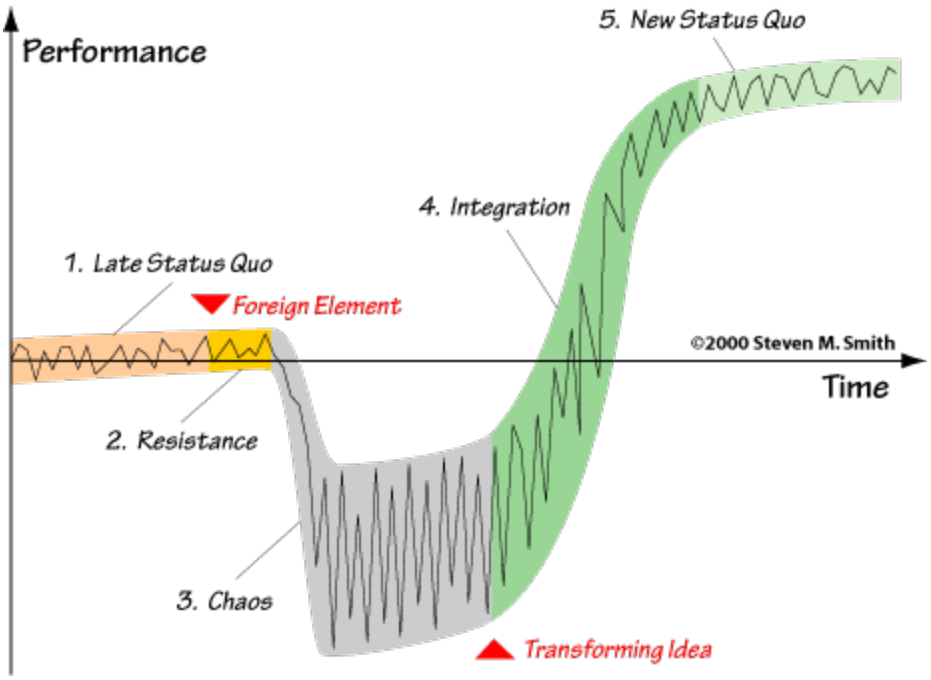
- Voluntary participation
- The experience of change
- Don't start with practices, frameworks, or tools
- The risk of misalignment

"*Becoming Agile* requires making deeper changes to engaging people, managing work, defining success, communicating, behaving, and so on."

Voluntary participation

- **Bad idea**: people == resources that need utilizing and controlling
- You want people to be their best selves; you get this through…
    - Psychological safety
    - Respect
    - Trust
    - Transparency
- If you think any of the above four are issues for your team, you'd do well to address those before trying to introduce Agile. People don't like change when it's not aligned with reality.

The experience of change



- Image: Satir change model
- Most people feel the **chaos** phase the deepest. This is time of struggle. “Competent people who successfully followed the old process now find themselves in unfamiliar territory.” The default reaction is to reject the change.
- What this looks like…
  - *Iteratively* doesn’t mean *quick and dirty*; it means *start small and deliver it cleanly*
  - Collaboration isn’t just about being helpful or working with others; it’s intentional
  - How leaders talk about missed expectations really matters
- **Idea**: replace the term “Agile” with something that has less baggage

Don’t start with practices, frameworks, or tools

- Even though these things (e.g., Scrum) are popular, installing them won’t bring about the culture and mindset to make Agile successful.
- Work with your team to design the methods. “If you go in with fully baked methods, they will see them as prescription or imposition, and few teams appreciate that.”

The risk of misalignment

- Mindset and tactics misaligned – saying you value things, but acting otherwise
- People that depend on each other operating with different mindsets – for example, some collaborate and others prefer solo time, team goes Agile but stakeholders don’t

Ch 6: Prepare the Journey

- Choose a good time to start
- Get your manager and stakeholders on board
- Prepare the team
- Prepare yourself

Choose a good time to start

- If the team is already drowning in work, they’ll have no time to learn new ways of working.
- Try starting after a big project wraps up.
- If you try to introduce Agile in the middle of a project (where most of the high-impact decisions have been made and now it’s time to execute), your approach will neither make much difference nor teach the team about Agile.

Get your manager and stakeholders on board

- Actively collaborate with your manager about where you want to go; they need to be your partner.
- Managers need to make it safe to fail; you will be learning things, which involves making mistakes.
- "…castigating the team and demanding accountability – will make people retreat to the perceived safety of their traditional methods and thus stop the journey dead in its tracks."
- Err on the side of overcommunicating.

Prepare the team

- Complications
    - Motivation – some people are more comfortable with the status quo
    - Interest – many think Agile is a process/methodology
    - Empowerment – mandated Agile is rarely successful
    - Language – terms can sometimes come with baggage (e.g., "ceremonies", "process")
- "Remember though: don't force Agile on people. Your goal is to get everyone ready, willing, and able to partake in an experiment that reflects some real change."

Prepare yourself

- "It calls for coaching instead of telling, advising, and rewarding."
- "As an Agile leader, you will experience a constant balancing act of guiding, empowering, offering solutions, and coaching. Just remember to *lead* more than you *manage*."
- If you don't have an internal champion, here are some attributes (see the PDF for more context)
    - People-oriented
    - Empathetic
    - Credible and trustworthy
    - Flexible
    - Versed in systems thinking
    - Demonstrable results
    - Tailoring
    - Aimed toward getting the team self-sufficient

Ch 7: Learn Enough about the Agile Principles to Get Started

- Organize People Around Value Creation
- Collaborate on a Product, Service, or Solution
- Produce Outcomes of Value
- Always Work on What's Most Important
- Get Feedback Frequently
- Keep the Cost of Change Low
- Constrain the Intake of Work

- Visualize the Work
- Break Work Down
- Bounded Team Autonomy
- Self-organization
- Collaboration

Organize People Around Value Creation

- Most companies organize people **by function or role**
  - Advantages…
    - People become experts
    - Colleagues with the same skills can back each other up and help each other develop professionally
    - The org can treat them uniformly in terms of career growth and performance management
    - This arrangement is ideal for hand-offs from one specialty to the next
  - Disadvantages…
    - Specialists only focus on their part, losing sight of the ultimate value/use of their work
    - Questions don't get raised, and answers aren't shared among them
    - There are delays because of hand-offs; you also need more coordination because of multiple touch-points for work
    - Having to pick up work, put it down, then resume often leads to lost context
- Agile orients people around **value delivery** instead of task completion
- Team members aren't clones, but their responsibility lies in producing the intended outcome

Collaborate on a Product, Service, or Solution

- Projects have…
  - Clear beginning and end
  - Stakeholders
  - Resources (a.k.a. staff)
  - Budgets
  - Timelines
  - Deliverables
- Agile favors treating the object of the work as a **product**…
  - Has life before and after making it
  - Has customers with emotional dimensions
  - Engages workers because it's generative instead of ticking boxes
  - Less of an us-versus-them mindset between the doers and the beneficiaries
- You can think of workflows as services (e.g., onboarding a new employee, which has customers such as the employee, the team they join, and the org at large)
- The Agile mindset is more about partnership – a two-way street where customers seek value and the empowered team that figures out how to accomplish those outcomes

Produce Outcomes of Value

- **Outcomes**
  - Address problems, needs, or goals
  - Mitigate an important risk
  - Learn (or get useful feedback) about something important
  - Make it possible for the business to do something significant
  - Enable or facilitate delivering value later
- **Deliverables** – ways you'll solve those problems, fulfill those needs, or accomplish those goals

Always Work on What's Most Important

- Finish work on one outcome at a time; also known as **limit work in progress (WIP)**
- Some problems (e.g., constructing a house) require up-front coordination because the cost of getting the single-pass effort wrong is too high.

- If you optimize for single-pass, you're not optimizing for adaptation. (Not that there's anything wrong with that; just be aware of what you're solving for.)
- In Agile teams, the backlog is not a commitment but a set of options.

Get Feedback Frequently

- Two terms that get mixed up
    - **Effective** – delivering value
    - **Efficient** – minimizing waste
- Either approach can be correct, but efficiency assumes you already know how to produce **the right thing**
- Find ways to get feedback on what you do.
- When you are learning, there will be mistakes; allow the safety to make those mistakes and grow from the process.

Keep the Cost of Change Low

- Traditional project management is about early big decisions and preventing change.
- Agile considers the **cost of change** (e.g., if a guess turns out to be wrong, what will it take to course-correct).
- Something that helps here is working short cycles (i.e., fast feedback) and delivering small amounts of value (i.e., **small batch sizes**).

Constrain the Intake of Work

- This gets back to reducing WIP. I've loved the phrase from Kanban: "Stop starting and start finishing."
- Starting several things at once can end up yielding false productivity because it can be faster to finish one thing and deliver it than to start three things that all deliver at the same (later) time.
- You need to stay just enough ahead of the work so you can do reasonable just-in-time planning
    - Wait to long to refine/groom your backlog = people get starved for work
    - Refine/groom for too long = team spends too much time planning on work that may not even be needed

Visualize the Work

- You can't understand the work if you can't see it all laid out in some form.
    - Everything's in one place
    - Easy to see when adding or re-prioritizing will impact something else
    - Creates a focal point for the team to congregate around
    - Everyone can see the state of everything
- There are numerous ways to do this (e.g., physical board with sticky notes, work item trackers like Azure DevOps).

Break Work Down

- If something's too big, it's difficult to get it started and understand when it will end.
- Find the smallest mini-outcome you can deliver; make that your unit of measure. (Sometimes these are referred to as *tasks* or *user stories*.)
- You need to have some sense of when the work unit ends.
- When large items sit "in progress" it's difficult to understand what's going on with it.

Bounded Team Autonomy

- Traditional project management is about control and predictability. This is all well and good, but not when favoring adaptability.
- "The boundaries include organizational culture, policies, guardrails, strategic choices, and other system constraints. The degree of freedom is high, but cowboy behaviors are not welcome."
- Why autonomy?
    - The people doing the work often know more about it than those leading them
    - Having approvals makes for delays
    - No set of policies and procedures can cover every reality
    - People with some control over their work are more engaged and motivated

Self-organization

- Most of the people on the team are specialists, with some generalists.
- Having a skill lie with only one person makes that person a bottleneck.
- A team with only generalists isn't always possible or desirable.

Collaboration

- Processing tasks
  - Solo – one person working
  - Hand-off – multiple people with individual ownership of parts
  - Collaboration – multiple people owning a task
- Minimize mistakes by having collaborative, self-organizing teams.
  - Ideas can grow in novel ways as people think together
  - Social fabric is knit
  - Mini hand-offs are immediate
  - Quality increases because more eyes are on the work
- A balanced Agile team will have a mixture of solo, hand-off, and collaborative work.

Ch 8: Design Your Initial Way of Working

- Choose operating principles
- Design the workflow
  - Step 1: Who Will Manage the List of Outcomes, and How?
  - Step 2: Who Will Determine and Sequence the Deliverables, and How?
  - Step 3: How Small Can You Make Work Items?
  - Step 4: What Does "Done" Mean?
  - Step 5: What Should the Workflow of a Typical Work Item Be?
  - Step 6: How Will You Handle Sensitive or Confidential Work?
  - Step 7: How Will You Visualize the Work?
  - Step 8: How Will You Constrain the Team's Work Intake?
  - Step 9: Which Feedback Loops Should You Have for Work Content?
  - Step 10: What Sort of Impediments Do You Foresee, and What Will You Do About Them?
  - Step 11: How Will You Get Finished Deliverables into Customers' Hands?
  - Step 12: Which Touchpoints Will the Team Have?
- Structure the Team
  - Membership
  - Roles and Responsibilities
  - Space

- **Design** – how will your group/org work together (human dynamics, constraints, culture, etc.)
- **Initial** – your first attempt will likely not be your last; start small to avoid complexity; don't standardize anything until it's proven
- **Way of working** – once your values, principles, and strategies are set, focus on new procedures, tools, and standards
- Just because you adopt "best practices", that doesn't mean you can always expect results. Also, if you don't make software, copying practices from Agile software teams isn't going to be a good fit.
- Designing a system isn't linear. Read over the guidance here to help you iterate on what's important first.

Choose operating principles

- Principles help you determine what processes, practices, roles, and team touchpoints you'll need.
- Where to look
  - Ch 4: Determine Where You'll Try Agile for the First Time
  - Ch 7: Learn Enough about the Agile Principles to Get Started
  - Things from your current approach that are working

Design the workflow

- **Workflow** = what people do to get from idea to deliverable
  - Sequence of steps

    - People take output of one step, apply procedures/practices to produce a new output
    - Can be linear, diverge for decisions, made of sub-workflows, have loops for iteration
- Agile teams tend to be more *cyclical* rather than *sequential*

Step 1: Who Will Manage the List of Outcomes, and How?

- **Outcome** = problem, need, or goal; the reason for having the team do some kind of work
- Identify who would decide which outcomes to address and how they make those choices.
    - Common roles: product owner, group manager
    - Can also be a small group instead of putting this responsibility on one person
- Have a transparent process for making decisions when facing disagreements
- What cadence will you revisit and reprioritize the list? Wait too long and you may go too far in the wrong direction; too often, and there's no stability and the team thrashes.

Step 2: Who Will Determine and Sequence the Deliverables, and How?

- Favor involving your stakeholders, subject matter experts, and the team. (The default is the hierarchy of senior people or managers.)
- Attributes: collaborative, transparent

Step 3: How Small Can You Make Work Items?

- Size should allow frequent delivery so you can…
    - Learn and get feedback
    - Mitigate risk
    - Seize opportunities
    - Enable future delivery
- Aim for tasks broken down such that you can see tangible progress every day

Step 4: What Does "Done" Mean?

- Which attributes or qualities should work have that allow the team to move onto the next thing and not be concerned about loose ends?
- It's up to your team to decide how detailed this needs to be. For example, for the task "preparing a meal" it may be sufficient to say (1) table is set, (2) planned dishes are on the table ready to eat. There's nothing explicitly about the food being safe to eat, how much food there is, etc.

Step 5: What Should the Workflow of a Typical Work Item Be?

- Unless you have something defined, go with **To Do, Doing, Done**.
- Add states that help you understand what state things are in. For example, **Blocked** is another state you don't want things to stay in for too long.
- Balance
    - The more states you have, the better you can see where things are in your process.
    - The more states you have, the more complicated it becomes with overhead of determining current states for work.
- Not every team will have the same workflow. Prefer tools that prevent you from enforcing one workflow on everyone.

Step 6: How Will You Handle Sensitive or Confidential Work?

- Transparency is helpful, but there may be some things that should be protected from those without need to know.
- Digital tools make this fairly easy by only granting certain visibility to your work tracker to specific people.

Step 7: How Will You Visualize the Work?

- If your team isn't colocated, having a digital tool is best. If you don't like any of the existing tools, even a shared spreadsheet is better than nothing.
- Make it a habit for your team to look at work state. This helps alignment and gets visibility to what's important.

Step 8: How Will You Constrain the Team's Work Intake?

- Initially counterintuitive idea: having less work in progress helps you be more productive and stay on top of quality.

- How will you know you have too much? Consequences: overloaded team members, work started but not finished, too much solo work. If this happens, what can you learn?
- The team needs a strategy for choosing which tasks to work on next. This is especially important when you have different streams of work (e.g., exploratory projects, business-as-usual work).

Step 9: Which Feedback Loops Should You Have for Work Content?

- Questions to consider
  - What questions lead to answers that validate our approach?
  - How frequently should we ask?
  - Who should we ask?
  - Where would we get the feedback?
  - What types of questions would we try to resolve within the team before asking for help?
  - How can we make sure our feedback givers feel safe “getting real” with us?
  - How will we respond if the answers invalidate what we thought was true?
- It’s useful to meet on some regular basis to reflect on learnings.
- Where in your workflow can you get stuck because you need approval? What’s the cost of delay in getting that approval?
- Other ways of getting approval…
  - Collaboration – multiple people work together to produce something
  - Peer review
  - Early/frequent feedback from stakeholders
  - Clear acceptance criteria upfront

Step 10: What Sort of Impediments Do You Foresee, and What Will You Do About Them?

- Every organization and process encounters impediments.
- Generate a list of likely impediments
  - Are they due to team design (e.g., Team A must finish before Team B starts)?
  - Are approvals needed that act as bottlenecks?
  - Are there environmental factors – excessive interruptions, faulty assumptions, fractured communication, dropped balls?

Step 11: How Will You Get Finished Deliverables into Customers' Hands?

- When the whole thing is ready? Batches of small deliverables? Specific windows of opportunity?
- What form do they take?
- How will stakeholders know when they’re ready?
- How will you know they’re actually getting value from your work?

Step 12: Which Touchpoints Will the Team Have?

- What are your cycles?
- Common examples
  - Daily standup
  - Iteration (e.g., 1-2 week) review
  - Weekly backlog review to adjust to shifting priorities

Structure the Team

Membership

- Qualities
  - **Communication** – can people communicate easily and effectively?
  - **Relationships** – do they all get along?
  - **Equality** – is everyone treated equally, even if they’re contractors/interns/etc.?
  - **Collaboration** – can people work across departmental divides?
  - **Motivation** – do people want to work on this?
  - **Leadership** – is there someone to help the team succeed?
  - **Finishing** – does the team have everything they need to get things done?

        - **Delays** – when the team starts, can they get enough runway to finish?
        - **System** – can other teams proceed if they only have limited access to this team?
        - **Change** – does this team get and use feedback to learn and improve?
- What happens if those qualities don't look so good right now?
    - Team size isn't right
    - Members can't dedicate enough time because of other responsibilities
    - People with the info (e.g., subject matter experts) are stretched across too many other teams

Roles and Responsibilities

- Start with skills needed to get the work done
- Add in proposing solutions, processing feedback, managing incoming work
- Add these in as explicit responsibilities
    - Look after the way of working (process stewardship)
    - Look after team health
    - Remove impediments
    - Facilitate team meetings
- Who prioritizes outcomes, clarifies requests, sequences deliverables, gives feedback once produced? (Typically called a product owner.)
- Managers in Agile move more into people leadership, career development, staffing, budgeting, organizational capabilities, escalation, and strategy. The team does more self-management, or works with a product owner.

Space

Think about what physical and digital needs the team will have to work best together.

Ch 9: Support the Team During the First Few Months

- Start with a Kickoff
- Finish Small Valuable Work Together
- Make Working Agreements
- Stabilize the System
- Watch for Attitudes and Behaviors that Hamper Agility
- Reflect and Improve Frequently
- Lead Intentionally
- Assess How It's Going

Start with a Kickoff

- Call attention to this significant shift
- Acts as a formal start; should have an emotional impact, so think about a compelling delivery, confidence, and engagement
- Possible topics
    - Review the team charter (what the team will achieve for themselves, customers, and the org)
    - Reiterate the motivation for working differently than before
    - Remind people how the new design came about
    - Describe how the team will manifest their values and principles
    - Reiterate this is an experiment and what you hope to learn
    - Answer any open questions
    - Answer "what if" questions
    - Gain commitment

Finish Small Valuable Work Together

- Avoid punishments and rewards; use empathy, observation, listening, coaching, reminders, and reasoning.
- Find tangible ways to reinforce the new norms (e.g., posters, mantras).
- It's important to get some early wins of finishing work; otherwise, it's easy to slide back into the previous status quo.

Make Working Agreements

- You can't anticipate everything.
- Think about how you will collaborate, be transparent, make decisions, give/receive feedback, keep each other informed.
- Write these down; human memory is fallible. Also, writing things down allow you to put it out in the open to clarify and refine.
- These aren't bylaws or rules; agreements stay in place until the team decides to change them.

Stabilize the System

- Balance the amount of work the team takes on with the rate it can complete work.
- Benefits: reliability, business performance, agility
- Visualize work because you'll see where it's piling up
- Use WIP limits for workflow stages
- Manage chronic waits and blockers
- Demand will fluctuate, so don't keep people at 100% capacity
- Determine how you'll handle emergency work so that people can work sustainably while absorbing those urgent requests

Watch for Attitudes and Behaviors that Hamper Agility

- Change is difficult and takes time to become the new norm.
- Common issues to watch for
    - "Don't split the work because we need the whole thing" – remember that delayed feedback makes it harder to change course
    - "That work is Bob's specialty, so he'll do it" – look out for overloaded specialists; invest in spreading knowledge and skills around the team
    - "When will this be done?" – some work is difficult to estimate; offer a range, ask for help, ask for simplification
    - "Do your best to finish the remaining work (which is more than you have time for)" – ask how you can split the work up, simplify it, or renegotiate it
    - "I've done my part" – the team should work together to deliver, otherwise you just have internal handoffs instead of intra-team handoffs
    - "Can you do this extra work?" – use queues and prioritization instead of handling all drive-by requests
    - "I'll need one of your people for a few weeks" – share the impact of not having that person on your team

Reflect and Improve Frequently

- Retrospectives are learning opportunities to find out if your experiments are working.
- Resist the urge to skip reflection meetings.

Lead Intentionally

- Manage things; lead people.
- Leaders create culture through rewarding desirable behaviors and discouraging undesirable ones.
- "Leadership is what separates engaged, responsible workforces from ones where people only check boxes on their job descriptions."
- Some tips…
    - Support people through the change
    - Start and end with outcomes
    - Draw and defend clear boundaries
    - Watch the work, not the workers
    - Reduce the load of approvals and reviews
    - Set behavioral expectations for people
    - Look to the system, not the people, for explanations of problem behaviors
    - Notice how you communicate and act
    - Seek allies and support

Assess How It's Going

- How well are you following your principles? (Think of creating a scorecard.)
- "Your way of working doesn't have to look like everyone else's."
- Is your team more successful?

- What are the downsides to this new way of working?

Ch 10: Increase and Expand Agility

- Continuous Improvement
- Expand the Scope of Agility
    - Same Team, More Kinds of Work
    - More of the Value Stream
    - More Teams, Doing Different Work
    - More Teams, Doing Codependent Work
- Metrics and Measurements
- Warning, Dangers Ahead
    - The Lure of Standards, Frameworks, and “Best Practices”
    - Cultural Pull and Drift
- Managing People

Continuous Improvement

- Teams regularly and collaboratively improve the way they work (in context with their beliefs and values)
- Simplify – remove accidental complexity (e.g., a jigsaw puzzle has its own complexity, but attempting to use a table too small to spread out the pieces adds unnecessary complexity)
- Eliminate waste – it’s usually easier to do this than to improve the productive parts
    - Overproduction
    - Partially done work
    - Relearning
    - Hand-offs
    - Task switching
    - Delays
    - Defects
- Tighten the workflow
    - How is your work visualization board working?
    - Set expectations about how work enters and moves through the system
- Make opportunities for learning (e.g., observe your customers, experiment, reflect after activities, teach each other skills, get ideas from others in the org, hold a book club)
- Upgrade the team space to help you collaborate better
- Keep asking, “How do we know?” – don’t get blinded by confidence

Expand the Scope of Agility

Can you apply Agile to bigger things in your org?

Same Team, More Kinds of Work

- Now that you’ve proven one kind of work is suitable for Agile, what about other kinds of work?
- How can you manage the new work without extra friction (e.g., separate boards, single board)?

More of the Value Stream

- Can you expand to lengthier flows of work?
- Remember that the expansion is also an experiment; don’t mandate it.

More Teams, Doing Different Work

- If you’ve proven it works for one team, try another non-dependent team.
- Don’t standardize the first team’s process because the second team likely has a different context. However, encourage the new team to look to the first for ideas.

More Teams, Doing Codependent Work

- What works for one team is still important for multiple teams… outcome orientation, frequent feedback, understanding the cost of change.
- New things that need more attention: decision-making, coordinating, communication for alignment
- Resist the urge to fall back into centralized, top-down planning
- As you get bigger, you need to think about integrating the smaller pieces into a whole piece
- Try to keep functioning teams together instead of treating people like resources that can plug-and-play at will. “In many conditions the performance of stable, solid, collaborative teams far outweighs the sum of individuals' performance.”

Metrics and Measurements

- Metrics are easy to come up with; metrics **that matter** are more difficult
- It’s easy to measure local improvements, but harder to correlate them to system-wide effects
- Pitfall: The “good” metric is too hard to collect, so let’s read too much into the easier ones. For example, measuring a team by how much output they’re producing rather than by whether what they’re producing is what the customers want.
- Metrics don’t just measure a system; they **influence people’s behavior**.
- When there seems to be a problem (based on learnings from the metrics), it’s easy to look for people to blame, instead of examining whether the system itself is flawed.
- It’s worthwhile to measure **people metrics** (engagement, team health, motivation, morale); healthy teams achieve results.

Warning, Dangers Ahead

The Lure of Standards, Frameworks, and “Best Practices”

- Common questions
  - What’s the best way to…?
  - Other teams are doing X; should we do X as well?
  - Are we following best practices?
- Books, courses, certifications, etc. reinforce the perception of best practices as valuable, necessary, or best. But does that mean it’s best for you? Beware the **halo effect**.
- Just because there is no “right” way to do Agile doesn’t mean you should ignore what others are doing. Use them for ideas or inspiration.

Cultural Pull and Drift

- To get agility, you need people-orientation, empowerment, psychological safety, tolerance of ambiguity, openness, etc.
- Cultures aren’t fixed. Influential people change cultures gradually and intentionally through words and actions.
- Drift = gradual move away from one culture to another

Managing People

- Traditional – industrial mindset, people are “resources” to be utilized, carrot/stick mindset, weaknesses need fixing
- Agile – frequent coaching based on observations, empathy, and honest feedback
- Prefer **trust over accountability.** Treat people like responsible adults and make it possible for them to succeed. “Hold people accountable” typically means that you need to watch for people to fall short.
- Motivators
  - Intrinsic – autonomy, mastery, purpose, enjoyability, challenge
  - Extrinsic – promotions, rewards, praise
  - Hygiene – safe workplace, living wage, competent management, knowing how to succeed
- Extrinsic motivators can work in the short-term, but over time people start caring more about the rewards than the work. They can also create competition between teams. Do recognize team and individual contributions, but focus less on rewards and praise.

Ch 11: Epilogue - Agile for Non-Software Teams

- Definitions of organizational agility
  - Build the right things, build them right, and build at the right speed
  - Be adaptive, creative, and resilient when dealing with VUCA
- More people face customers, act like entrepreneurs, and determine direction
  - Teams are focused on product, service, solution, outcome, and value
  - People in similar roles/titles work together for career development, mutual learning, and consistency
- Managers

- Enable decisions made by people closer to the work
- Work with other managers to support initiatives, learn, and improve systems
- Coach up-and-coming leaders
- Are intentional about the culture they want