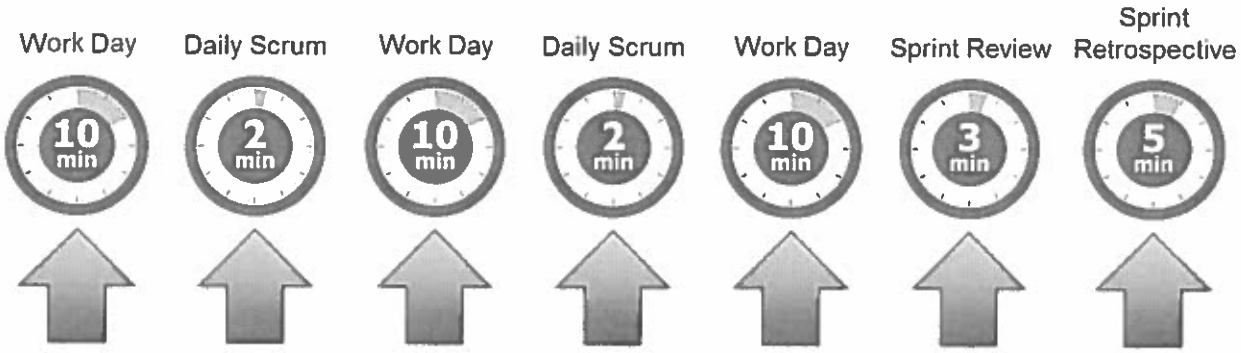
Simulation: Sprint 1

- For this Sprint simulation, follow the plan you created during Sprint Planning.
- Work on your simulated tasks as you can, creating paper prototypes.
- There will be 3 simulated "days" in each Sprint as follows:
  - Day 1: 10 minutes "work day."
  - Day 2: 2 minutes Daily Scrum, 10 minutes "work day."
  - Day 3: 2 minutes Daily Scrum, 10 minutes "work day."
  - Sprint Review: 3 minutes.
  - Sprint Retrospective: 5 minutes.



Discussion: Simulation

Questions:

- What observations do you have after the simulated Scrum events?
- Did your Team stick to a Scrum simulation or incorporate any influences from other Agile approaches?
- What challenges do you see in applying Scrum to your product or project? In your organization?

Simulation: Agile & Scrum Roles

Activity:

- We are going to move into the next round of the Simulation exercise.
- If you did not discuss what to approach differently at your Retrospective, take a moment to reflect on this.
- Did someone have to "role-play" the Product Owner or Scrum Master if they are not in attendance with your real-life Team?
- Did you have other approaches you wanted to experiment with?
- Do you have other people in your class Scrum Team who want to role-play as the Product Owner or SM?
- Take a minute to agree as a Team on any changes before we move into the Sprint 2 Planning session.

Simulation: Sprint 2

Instructions:

- In your Scrum Teams, simulate Sprint Planning, discuss velocity/capacity, and pull items from the top of the Product Backlog. Break Product Backlog Items down into tasks.
- Work on your tasks as you can, with your wireframing project.
- Following Sprint Planning, there will be 3 simulated "days" in each Sprint as follows:
    - Day 1: 10 minutes "work day."
    - Day 2: 2 minutes Daily Scrum, 10 minutes "work day."
    - Day 3: 2 minutes Daily Scrum, 10 minutes "work day."
    - Sprint Review: 3 minutes.
    - Sprint Retrospective: 5 minutes.
- Work together in breakout rooms and then be prepared to share your work at the end of the session.

Discussion: Inspect & Adapt

Question:

- How was what you did in Sprint 2 different from what you did in Sprint 1 in terms of your Scrum Team's approach?
- How did you reach the decisions to make these changes?
- Were the changes large or small?

Transparency and Information Radiators

Review: Transparency

- As a reminder, the three pillars of any empirical process are: inspection, adaptation, and transparency.
- Agile methods are empirical and allow Teams to inspect and adapt, and provide a high degree of visibility or transparency into what is happening on the project or with the product.
- The intent is that this visibility is for the Scrum Team and key stakeholders (primarily Sprint Review, Product Backlog).
- With this level of transparency, they can make "course corrections" in real time if faced with an impediment that would delay the project or the Release of business value.
- This level of transparency increases the communication on the Scrum Team and establishes trust.

Discussion: Working Product Increment

- The Product Owner or customer should realize value at the end of each Sprint.
- The Scrum Team should strive to have "production worthy" code, even if it's a product increment, and even if it will be held for Release into other increments.
- Are we carrying over unfinished work every Sprint, creating technical debt, and not reaching "done"?
- What kinds of things prevent a Scrum Team from reaching the Definition of Done?

How Google Tests Software
- good test book, light weight tests

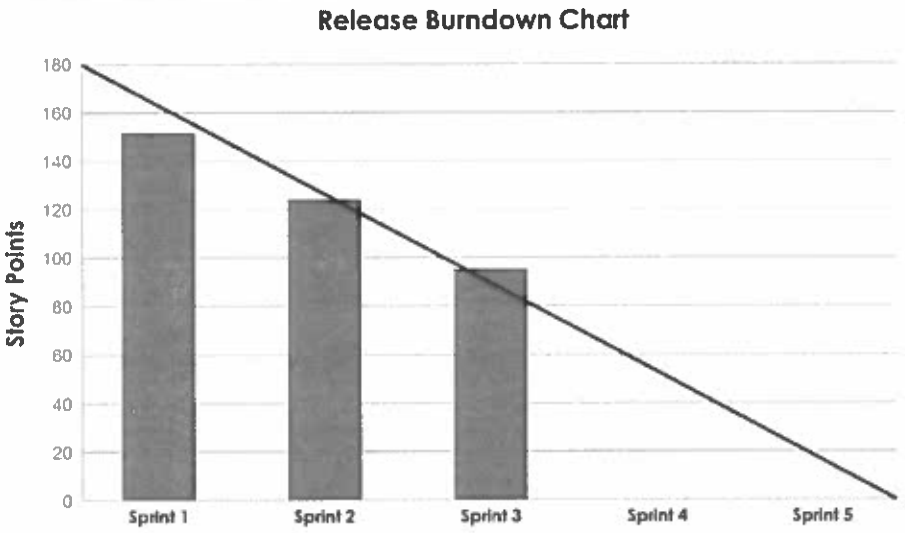
Discussion: Technical Debt

Question:

- What happens if we continually do not achieve the Definition of Done?

Release Burndown

- A simple visual of the overall completeness of a Release.
- Developed and updated by Product Owner following each Sprint.
- The baseline is the ideal line. It gives us the baseline of where we should be at any given Sprint for a Release.

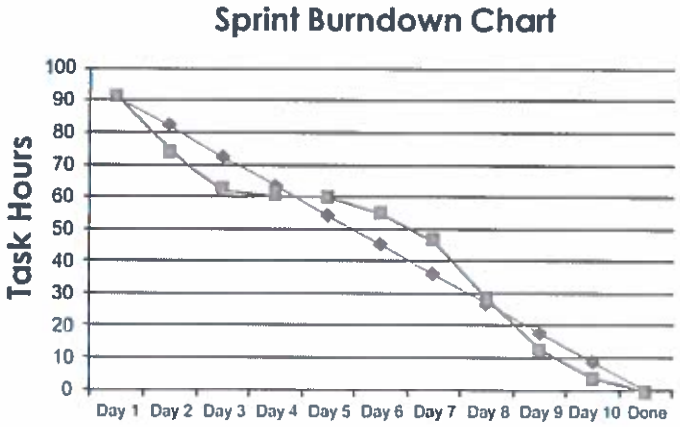


- Baseline: The black line on the chart shows where a Release's remaining work should ideally be.
- The image depicts a project with five Sprints.
- The project initially had 183 Story Points.
- The Development Team appears to be burning approximately 30 Stories per Sprint.
- This image depicts the Development Team's velocity.
- We are burning down as Sprints complete.
- At the end of each Sprint, the Scrum Master updates the completed Story Points count into the chart; the remaining work adjusts.
- The chart is a clear and simple indication of Release Health.
- Ideally at the end of the Sprint, work remaining should be zero. If all allocated Release Stories are not completed by the last Sprint in the Release, a blue bar will be present at Release five.
- If additional Stories are completed during the Sprint, then the blue bar will be present and dip below the 0 line.

Sprint Burndown Chart

Story Points

- A simple visual of the status of the Dev Team's Sprint forecast.
- Developed and updated by the Development Team every 24 hours.
- The baseline is the ideal line. It gives us the baseline of how much work the Dev Team should have remaining based on their initial forecast.
- The actual line shows how much work is actually remaining as of that day in the Sprint.



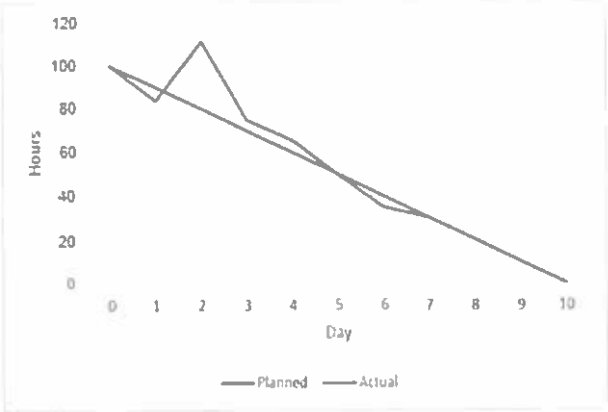
- The Sprint Burndown Chart, which is maintained by the Development Team, is an important tool.
- The Sprint Burndown Chart, if updated daily, shows the work remaining for the Development Team versus its original Sprint forecast.
- At any given time, the Development Team, Product Owner, and other stakeholders can get a feel for how the Development Team is progressing against its forecast.
- A flat line indicates no progress.
- A Spike in the line indicates a burnup of existing or newly added tasks. The Development Team made a discovery that was missed during Sprint Planning.
- A line that hits the bottom indicates there are no more hours remaining in the Sprint.

Discussion: Sprint Burndown

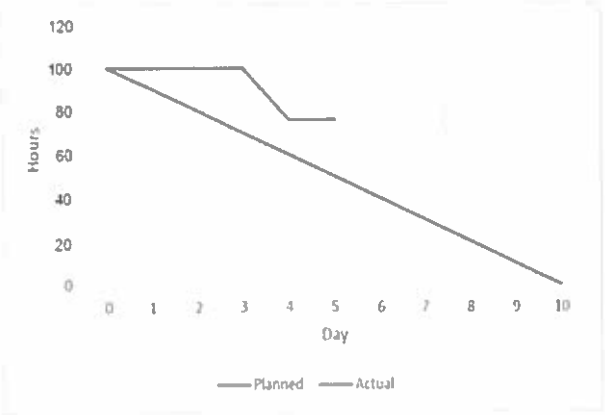
Question:

- Based on the two Sprint Burndowns below:
  - What is your analysis of the overall health of the Sprints?
  - What conversations should be taking place?

Sprint 3



Sprint 7



Assessment: Transparency and Information Radiators

Questions:

1. True or False? Transparency allows Agile Teams to inspect and adapt and make course corrections in real time if faced with an impediment that would delay the project or the Release.
2. Who develops and updates the Release burndown?
   A. Product Owner
   B. Scrum Master
   C. Development Team
3. How often is the Sprint burndown chart updated?
   A. Every day
   B. Once a week
   C. At the end of each Sprint
   D. After every Release

TEKsystems
Our people make IT possible.

Continuous Improvement

Section 19

Continuous Improvement

- Remember The Agile Manifesto principle:
  - At regular intervals, the Team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
- This is not a simple reference to a Retrospective at the end of an Iteration or Sprint.
- This refers to the concept of Continuous Improvement in any Lean or Agile process.
- The Team looks at what is and is not working with the current process, and then makes adjustments they can immediately put into practice.
- In order to continue providing value to customers and clients, it is important that we continue to re-evaluate ourselves as well.

- Most of the Agile methods or approaches have roots in Lean thinking.
- At the heart of Lean thinking is always looking for Continuous Improvement in the process that is used.
- Scrum specifically sets aside time at the end of each Sprint for the Scrum Team to inspect their process and adapt based on what is working well and what is not working very well.
- Other methods speak to Continuous Improvement by examining engineering practices for improvement (XP), increasing throughput (Kanban), etc.
- It is important to note that we're talking about improving the process, not necessarily the product.
- Product improvement will improve and quality will improve if we improve our process.

Incremental Change

- Just as we have learned about incremental product development, change can also come about incrementally.
- We have learned a lot of new values, principles, and concepts, and it would be unrealistic for an entire company to consider applying all of these at once.
- Start with a pilot Team, project, or product group to help identify where your organization could more easily adopt these practices, and where it may face the most challenges or impediments.
- When choosing a pilot to gauge how your organization will adapt to Agile, avoid high-profile or politically charged Teams, products, or projects.
- Look for a business partner or sponsor who is willing and engaged, and will fill the role of or supply a Product Owner to provide direction and feedback to the Scrum Team.

Discuss with other participants: Is your organization approaching change incrementally or attempting "big bang"?

Incremental Change (Cont.)

- Choose a Team who will give Agile a fair try:
  - Some may be skeptical.
  - Find an open-minded Team to help determine if the method is feasible within organizational constraints.
- Ensure that the project is not too big or too small.
- Projects longer than six months in duration are probably too large for an initial pilot.

- Change does not need to happen all at once or "big bang."
- In adopting agility, it is recommended that organizations do NOT take this approach, but rather, adopt a more gradual change.
- Starting with a pilot Team to adopt an Agile method can be a great way for companies to realize where the impediments are and provide the opportunity to address them before rolling Agile out to another Team in the organization.

Assessment: Continuous Improvement

Questions:

1. True or False? Continuous improvement requires that we evaluate what is and is not working with the current process and then make adjustments we can immediately put into practice.
2. When an organization is adopting Agile, it is best to:
   A. Change everything all at once (big bang) ← Common
   B. Change gradually

TEKsystems
Our people make IT possible.

Impediments

Agile Anti-Patterns

- An "anti-pattern" is any behavior or practice that impedes your Agile adoption.
- These may seem useful at first glance, but in practice, they impede agility and your ability to achieve your Agile goals or deliver customer value quickly.
- These may also be referred to as "common problems" or "common reasons Agile fails."

- Instead of being open to something new, the mantra of "the way we've always done it" will start to be heard or the excuse of "the company's operational processes," "our change management process," etc.
- Staying entrenched in this type of thinking or practice can impede any Agile adoption and will set it up for failure—not success.

Impediments to Agile Success

1	Preserving Command and Control hierarchies over self-organization.
2	No active involvement from the business (Product Owner) or customer.
3	Documentation for documentation's sake and not for a useful goal or purpose.
4	Adopting Agile without defining goals or business problems that Agile will address, improve, or fix.
5	Unwillingness to invest in training and/or coaching.
6	Emphasis on metrics over project results.
7	Belief that Agile is for software development and not for business.
8	Over-forecasting in Iterations for fear the Development Team will not be busy enough.

- Several of the more common reasons why Agile adoptions or transformations fail are listed here.
- Discuss with other class participants: Do any of these resonate with you or your Team? Your company?

Determine Root Cause of Impediments

- Another common problem is turning work in Iterations or Sprints into "mini-waterfalls."
- This occurs when we stay locked in our roles, ignoring the fact that our goal is to deliver a working increment at the end of each Iteration or Sprint.
- What does it mean to be "code complete" at the end of a Sprint:
  - Does it mean the Backlog Items in the Sprint complete?
  - Does it imply that you are Test Complete?
  - Has your "customer" or Product Owner signed off and accepted the increment?
- If the Item is only code complete, Items are "carried over" for testing, acceptance, and refactoring into subsequent Sprints—thus, your mini-waterfall project!
- Basically, you are not finished with a feature until it meets the benchmark defined by done.

- In identifying root causes, Team members are encouraged and empowered to openly and honestly give visibility to what the problem is.
- Their Scrum Master, Coach, Agile champion, etc., should encourage these types of discussions.
- Those advocates can also be instrumental in facilitating discussions, so focus moves off of problem identification, or "ranting", about the issue and onto possible solutions and how to solve the problem(s).

Exercise: Recognize Anti-Patterns

Instructions:

- Self-organize into Teams, even if you are with different Team members from previous exercises.
- Your instructor will provide scenarios from Agile projects.
- Your Team's job is to discuss and identify the anti-patterns and what might be the root cause of the problem.

Anti-Pattern Scenario #1

- At the end of each Sprint, there is always a pile of Stories "not complete" or that did not reach the "Definition of Done."
- Instead of putting incomplete items back on the Product Backlog, the Development Team insists on getting "credit" for what WAS done and splits the Stories.
- This artificially inflates the Development Team velocity and keeps the Development Team in this cycle of carrying over Technical Debt from Sprint to Sprint.
- Is this optimal? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Anti-Pattern Scenario #2

- Your Development Team has decided to use Story Points and uses cards to collaboratively reach consensus on estimates.
- The Product Owner and manager attending planning sessions routinely interject, and have asked for a points-to-hours "conversion" so they can more effectively manage the Team's work.
- Is this optimal? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Anti-Pattern Scenario #3

- Your Development Team has been together for six 2-week Sprints to date with a running average velocity of 76 Story Points.
- With the holidays coming up, the Scrum Master sends out new meeting invitations extending one Sprint for several days and shortening another Sprint to "get around" holidays.
- Is this optimal? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Anti-Pattern Scenario #4

- The Tech Lead on your Team is also the Scrum Master. Scrums run over, planning sessions run over due to lack of preparation, impediments have not been addressed for some time, the Product Owner is rarely ready for refinement sessions, etc.
- Coding tasks given to a "Tech Lead/SM" are sometimes late, and other times are contributing to Technical Debt for the Dev Team.
- Is this optimal? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Anti-Pattern Scenario #5

- Your Development Team has been together for a little over a year adopting Scrum and blending in some XP practices.
- The group has decided Sprint Retrospectives should be cancelled since there are rarely impediments or issues any more.
- Is this optimal? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Anti-Pattern Scenario #6

- During Sprint Planning, the Development Team has gotten into the habit of planning "the entire Sprint," assigning specific tasks to specific people based on comfort zone, history, etc.
- At the end of the Sprint, there are several Stories that are not done based on the "silos" created by individuals working on the tasks for those Stories.
- Is this an optimal scenario? Why or why not?
- What anti-patterns do you see?
- What would you do differently?

Discussion: Anti-Patterns

Questions:

- Based on our exercise around anti-patterns and impediments, what challenges do you see in your real-life Team adopting Agile? Your organization?
- What are some possible choices your Team or your organization can make to correct these?

Assessment: Impediments

Questions:

1. True or False? A company can successfully implement Agile without any active involvement from the business or customer.
2. True or False? A common problem in Agile is turning work in Iterations or Sprints into mini waterfalls.

What Now?

Shu Ha Ri

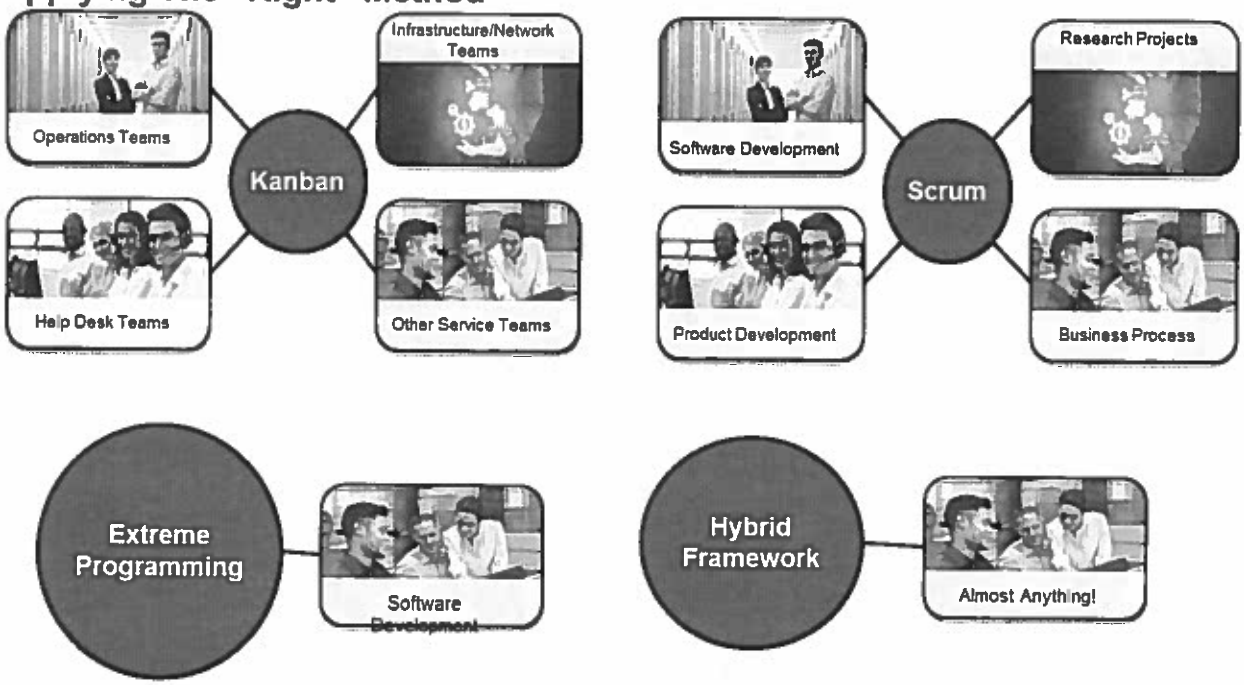
Learning, applying, and customizing new practices can be a challenge:

- "**Shu Ha Ri**" is a learning technique that comes from the Japanese martial of Aikido:
  - **Shu** "*learn*": The student has a "master" that provides guidance. The student follows the master—not worrying about alternatives, underlying theory or nuances.
  - **Ha** "*detach*": The student is now able to apply the basic practices and start to learn underlying principles and theories. Other masters are identified and some of their teachings are incorporated.
  - **Ri** "*transcend*": The student now learns from her own practice and efforts. She creates new approaches and adapts practices to her situation.
- Shu Ha Ri can be applied in your own growth in learning about Agile.
- Shu Ha Ri can be used as you learn, adopt, and succeed with Agile.

- Further discussion of Shu Ha Ri can be found via:
  - http://agilecoach.typepad.com/agile-coaching/2010/02/shuhari-considered-harmful.html.
  - http://c2.com/cgi/wiki?ShuHaRi.
  - http://alistair.cockburn.us/Shu+Ha+Ri.
  - http://martinfowler.com/bliki/ShuHaRi.html.
- Remember that when working with the Master, they are not dictating a one-size-fits all approach to the adoption of the new skills. They are aware of the context and are guiding you to useful skills and practices. In the early stages you are placing trust in the master to guide you to appropriate practices. Later, as you gain experience and mastery, you start to develop new approaches.

守破離

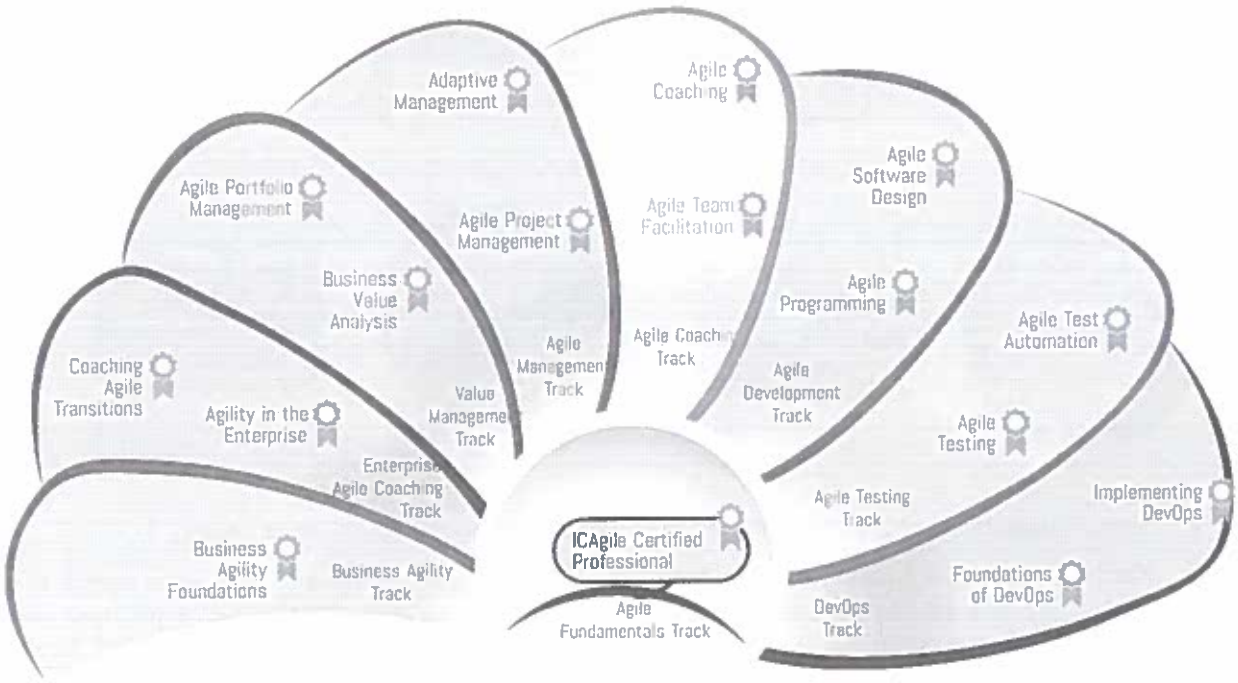
Applying The "Right" Method



- Many Teams find that a particular Agile method may be more suited to their needs
- **Kanban:** Kanban is typically more suited to service teams, help desks and operations teams. Any process serving a continuous stream of relatively independent tasks may be well suited to Kanban.
- **Scrum:** Scrum is typically more suited to product development or research based projects. Any process serving groups of co-dependent task that all add up to a lager deliverable product my be well suited to Scrum.
- **Extreme Programming:** Extreme programming can be very effective in a software development project. It is rare to see this Agile method employed outside the realm of software development.
- **Hybrid Methods:** Many Teams find it best to use a hybrid method. These Teams borrow practices from several Agile methods to create a custom method specifically suited to their needs. One example, referred to as "ScrumBan," uses the Kanban framework and borrows practices from Scrum such as the daily standup meeting, retrospectives and the roles of Product owner and Scrum Master. Teams using this method may also borrow practices from Extreme programming such as Test Driven Development and Pair Programming.

Agile Beyond Software Development

ICAgile Roadmap highlights additional areas of Agile: Leadership, Coaching, Portfolio Management, Project Management, Facilitation, Business Agility



In this course, we're prepared you for receiving the ICAgile Certified Professional designation. This is the foundation for all of the other ICAgile certifications. Take a moment to review the diagram and note that many other topics are covered in the program. This highlights that there is more to Agile than just Software Development.

Discussion: Agile Beyond Software

Question: Suppose you are putting together a Team of writers to create a monthly blog website. Which Agile practices would you adopt? Would you have to modify them? If so, how?

- ❑ The 12 Agile Principles
- ❑ Transparency, Inspection, Adaptation
- ❑ Stand Up Meeting
- ❑ Retrospective
- ❑ Acceptance Tests
- ❑ User Stories
- ❑ Demo / Sprint Review
- ❑ Test-Driven Development (TDD)
- ❑ Other?

Agile values and practices have been used outside software products and projects. Such projects range from sales and marketing campaigns to running businesses to educating students. As you think about non-software projects, you may feel some of the items on this slide do not apply. But it may surprise you how many do apply, or can with a little adaptation. The blog article "Agile marketing helps you innovate faster – but it's not right for every team" provides some insight on marketing teams adapting Agile for their own needs. https://www.teksystems.com/en/resources/teksavvy-blog/2016/agile-marketing-not-right-for-everyone

Exercise: Next Steps

Instructions:

- Discuss your next steps or plan of action for improvement in your job or role.
- Use whatever materials you need to facilitate this discussion: index cards, sticky notes, flip chart paper, etc.
- This process should include identification of the next steps, any order or priority for tackling them, and timelines if at all possible.
- Discuss with your instructor as needed.

Discussion: Next Steps

Questions:

- What did you come up with?
- Could you use more time to complete this exercise? If yes, you may want to schedule time outside the class to do this, especially if there are other roles, such as a Product Owner, needed for the discussion.

Recap

Let's Recap

- What Agile values and principles are most important to you and your Team?
- Based on the methods we've discussed, which Agile method do you think will work best for your Team and organization? Why?
- What are your Goals in Adopting Agile?
- Who is your customer?
- What is your Product, Service, or Project Vision?
- What is the difference between a Product Backlog and a Sprint Backlog?
- What Estimation method is your Team or organization most likely to use, and why?
- What levels of planning are standard for Agile?
- Why is it important to continuously review and adapt at all levels of the Agile process?
- How can your organization change incrementally to ensure the success of Agile?
- What impediments to success do you think you may need to overcome in your organization?

Appendix A: Continuing Education

Claiming Scrum Alliance SEUs

Claiming Scrum Education Units (SEUs):

- If you already have a Certified Scrum Master®, Certified Scrum Product Owner®, or Certified Scrum Developer® certification, the hours in this class will count towards your continuing education needs to advance to or maintain the next level: Certified Scrum Professional® (CSP).
- Check the Scrum Alliance website at http://www.scrumalliance.org for the latest requirements and to claim your SEUs.

Certified Scrum Master®, Certified Scrum Product Owner®, Certified Scrum Developer®, and Certified Scrum Professional® are registered trademarks of the Scrum Alliance

Claiming PMI PDUs

Claiming PMI Professional Development Units (PDUs):

- As a Project Management Professional (PMP) or PMI-Agile Certified Practitioner (ACP), you may be eligible for Professional Development Units (PDUs) or qualifying educational hours from taking this course.
- You may claim 1 PDU for each 1 contact hour of this course that is considered an applicable activity for the respective credential that you hold.
- For more information about claiming PMI PDUs, refer to the document "Steps to Report PDUs" provided with your training materials for this course.

Thank you for your time!

Let's clear our Product Backlog!

TEKsystems

Our people make IT possible.

TEKsystems Education Services would like to thank you for attending this class. We hope it has met your expectations.

Please go to www.metricsthatmatter.com/teksystems to complete the final course evaluation. We greatly appreciate your feedback.

Thank you!

Sincerely,

TEKsystems Education Services Team