simplilearn

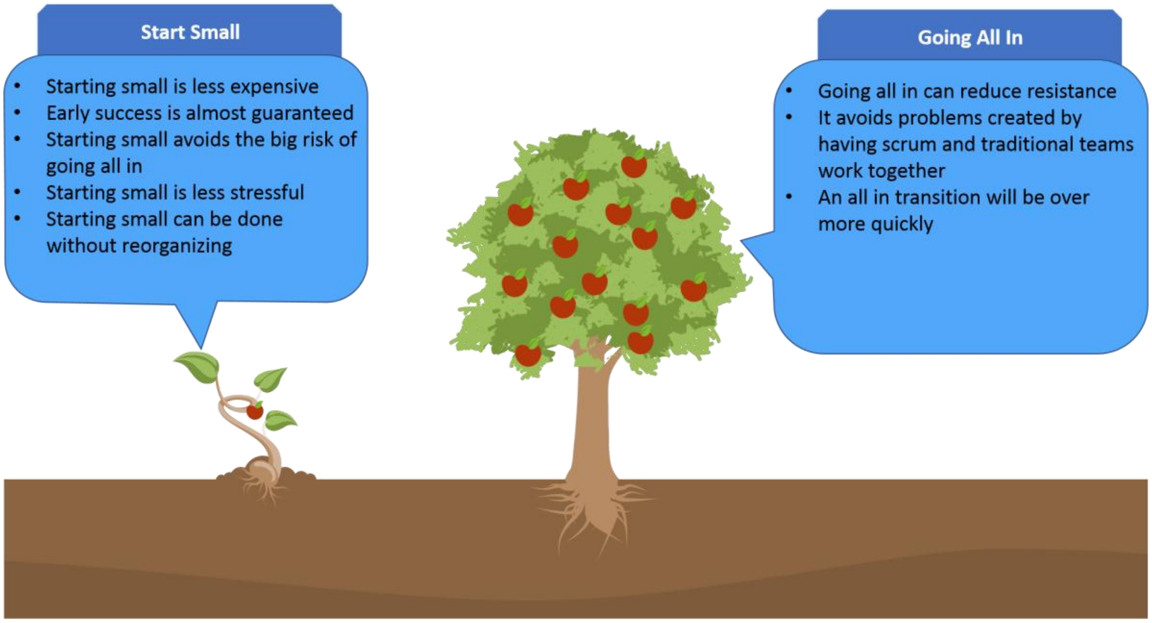
your pace, your place

Successful Scrum Adoption

Agile Adoption Patterns

- Any organization that is willing to start an Agile transformation needs to address the following questions at the start of Scrum adoption:
  - Shall we start with one or two teams OR shall we transform all teams at the same time?
  - Shall we announce our intent publicly OR shall we keep quiet?
- In this section, we will look at the options and patterns for spreading Scrum.

Start Small or Go All In



Start Small

- Starting small is less expensive
- Early success is almost guaranteed
- Starting small avoids the big risk of going all in
- Starting small is less stressful
- Starting small can be done without reorganizing

Going All In

- Going all in can reduce resistance
- It avoids problems created by having scrum and traditional teams work together
- An all in transition will be over more quickly

- Conventional and time-trusted advice regarding transitioning to Scrum is to start with a pilot project, learn from it, and then spread Agile throughout the organization.
- In this ***start small*** pattern, an organization starts with typically one to three teams, gets them successful, and then spreads Scrum from there.
- The start small pattern may not be used by every organization. A few organizations will transition ***all their teams*** to Agile at once – making it an ***all-in*** pattern.
- A few organizations may do both as per their requirement.

Reasons to prefer Starting Small

Starting small is less expensive

- Start small has a slower pace, allowing the organization to build internal Agile expertise and then leverage that to other teams.
- Start Small also saves money because the early mistakes can affect only a small team but not the bigger organization.

- All in approach might require a lot of investment just for external coaches, Scrum Masters, and trainers.

Early success is almost guaranteed

- Success of the first project is almost guaranteed by careful selection of initial project and team members.
- With start small, the goal is to generate knowledge and learn from experiences.
- Moreover, early success is vital to gain buy-in from skeptics or fence-sitters.

Starting small avoids the big risk of going all in

- All in approach carries higher risk because small mistakes can impact the entire organization.
- In an all-in approach, there might not be a second chance. Any mistake made can increase the resistance.
- With start small on the other hand, even if a fatal mistake is made, your team size doesn't change for the next sprints, so the team is effectively back to square one, that is, restarting the transition process.

Starting small is less stressful

- In today's dynamic world, employees and organizations are under constant stress.
- Just a simple announcement that the complete organization is transitioning to Scrum can affect almost all the aspects of everyday work of team members.
- With start small, the stress is reduced as the team size is small. Early adopters, that is, team members can become coaches and ambassadors as they can share their success stories and encourage other teams to undertake the transition.

Starting small can be done without reorganizing

- When undergoing a transition to Agile, most organizations require some degree of reorganizing.
- This can create further stress (for example, there is no PM role Scrum, so PMs may think they are now redundant in the organization) and can increase resistance.
- With start small, reorganization can happen much later and in a gradual manner.

Reasons to prefer going All In

Going all in can reduce resistance

- In any organization, when the entire organization is not covered, there will be some skeptics who will hope that the transition efforts would be abandoned.
- An all in transition effort affects the complete organization. There would be almost no resistance, and there would be visible commitment to change.

Avoids problems created by having Scrum and traditional teams work together

- In situations where only few teams are using Scrum and others are not using it, there is a possibility that a Scrum team might have to coordinate with other Non-scrum teams. This creates a few challenges and problems related to requirements, documentation, deadlines, etc.
- These challenges and problems go away with an all in approach as the entire organization adopts Scrum at the same time.

An all in transition will be over more quickly

- Transitions done in silos, that is, one department or business unit at a time, creates a feeling among employees that transition was one of the worst things they have experienced. But, when an all in transition is carried out, the transition's negative effects would be absorbed more quickly and would vanish.

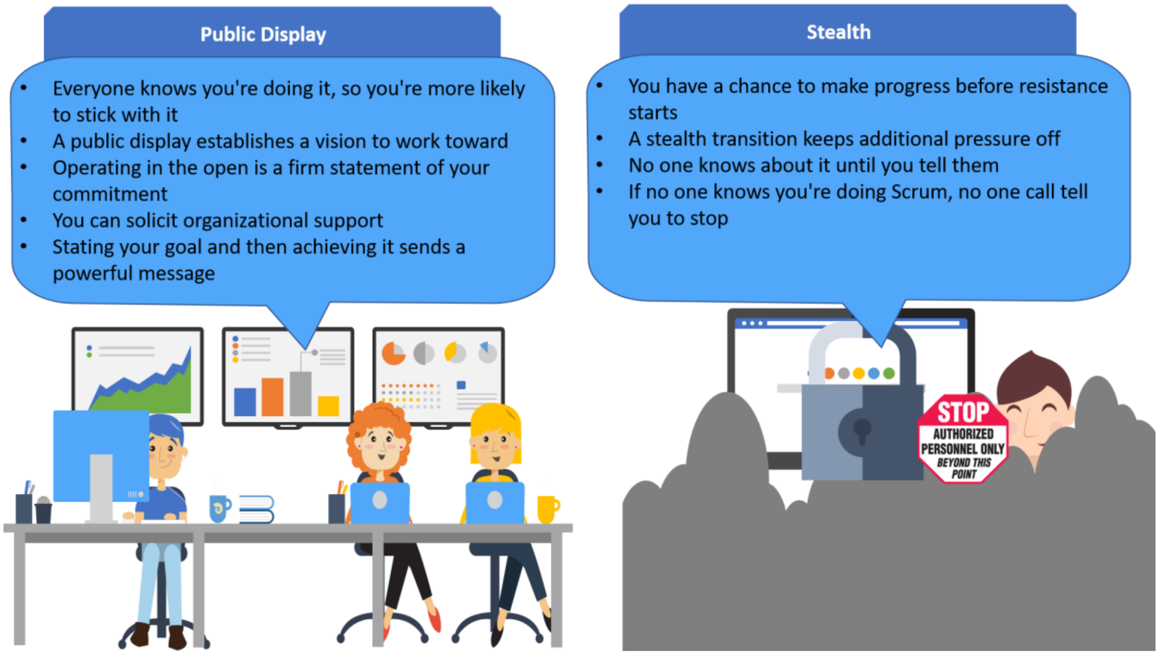
Choosing between going all in and starting small

- Starting small:
    - Default recommended approach
    - Low risk and high likelihood of success are excellent buy-in factors
    - When there is reluctance by leaders toward a full commitment to Agile transition, use this approach
    - If the cost of failure is too high (for example, an organization is developing a product for the first time, and it would have a severe impact on the organization if it fails), start-small is the way to go
    - Start-small is safe but slow

- All in
    - To be used in critical times such as when there is limited time
    - It will cost more to transition but takes less time
    - Choose this approach to silence a few critics and skeptical stakeholders who may show resistance

- This approach needs enough experienced Scrum Masters for all teams. Without them, do not try this approach
- If the total employees to be covered is, for example, above 400, going all in may create a few logistical problems

Public Display of Agility or Stealth



Public Display

- Everyone knows you're doing it, so you're more likely to stick with it
- A public display establishes a vision to work toward
- Operating in the open is a firm statement of your commitment
- You can solicit organizational support
- Stating your goal and then achieving it sends a powerful message

Stealth

- You have a chance to make progress before resistance starts
- A stealth transition keeps additional pressure off
- No one knows about it until you tell them
- If no one knows you're doing Scrum, no one call tell you to stop

- When initiating Agile transition using Scrum, an organization must decide whether they will publicize their transition or not.
- One approach is public display of agility:
  - The organization might announce the transition with much fanfare
  - Public announcements may include simple internal communication or press releases
  - In this public display of agility, teams will try to let others know that something Agile is going on
- The second approach that contrasts with the first one is stealth transition:
  - Only team members know they are using Scrum
  - Organizations deliberately do Scrum in stealth

Reasons to favor a public display of agility

Everyone knows you're doing it, so you're more likely to stick with it

- When an organization's intent of becoming Agile is announced publicly, there is unspoken pressure to succeed
- Moreover, the organization or teams may get support from others, which will encourage them to succeed

A public display establishes a vision to work forward

- Public display of work encourages team members to talk about their work even with those outside the teams. For example, if one team member gets another team member from Non-scrum project across the desk for a discussion, the team member from the Non-scrum project will notice the publicly-displayed Information radiators and will surely ask the team member from the Scrum project a few questions.
- Also, public display can help the team members to identify who are showing support and who are showing resistance. This will help the team to formulate effective stakeholder strategies.

Operating in the open is a firm statement of your commitment

- Once the project progress is publicly displayed, the team has no other option but to commit toward progress and succeed.
- It sends a powerful message that the team is committed and wants to succeed.

You can solicit organizational support

- A few organizational units such as facilities, human resources, etc. might offer support because of public display of agility
    - For example, Facilities might offer to provide a storage bin to keep the sticky notes, Story cards, markers, tapes, etc. in a safe and secure place.
    - Human resources might observe how everyone in the team is contributing toward project success

Stating your goal and then achieving it sends a powerful message

- Saying upfront that this project will be done using Scrum and announcing it again after successfully completing the project is more powerful than announcing the goal after achieving it.

Reasons to favor a Stealth transition

You have a chance to make progress before resistance starts

- As mentioned above, public display of agility will bring resistors out into the open.
- They will try to oppose the move strongly at the beginning itself.
- When stealth techniques are used, resistors may not be even aware of such transition. If they come to know after the project is successful, there may not be anything left for them to resist.

A stealth transition keeps additional pressure off

- Public display of agility might become a highly-publicized affair in the organization and may create additional pressure for the team.
- At the end of the project, the team may not even realize if success is due to Scrum or additional pressure.
- When stealth techniques are used, team members will not face additional pressure.

No one knows about it until you tell them

- When stealth techniques are used, other teams will not be aware that your team is using Scrum.
- Once the project is successful, your team can break the news that Scrum was used to deliver the project successfully.
- Even if the project fails midway, the team can find out how to overcome mistakes and share the knowledge with others.

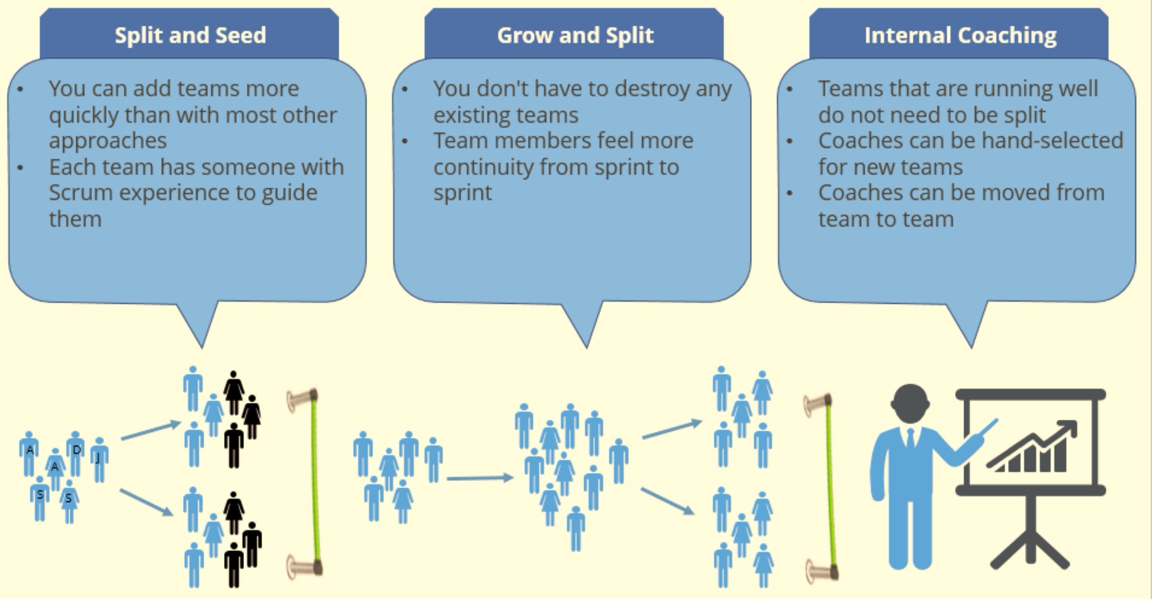
If no one knows you're doing Scrum, no one call tell you to stop

- Teams doing Scrum in stealth will never get someone to tell them to stop using Scrum because nobody might know.
- When under pressure, teams use Scrum in stealth so that they can ask for forgiveness later rather than permission upfront.

Choosing between a public display and stealth

- Public display is preferable when:
    - Teams want to enjoy a successful transition
    - Organization is committed and confident about the transition
    - Stiff resistance is expected and team leaders would like to overcome it quickly
- Stealth approach is preferable when:
    - Teams want to experiment with Scrum, either completely or partially
    - Teams must use it as the last and only option

Patterns for Spreading Scrum



Split and Seed

- You can add teams more quickly than with most other approaches
- Each team has someone with Scrum experience to guide them

Grow and Split

- You don't have to destroy any existing teams
- Team members feel more continuity from sprint to sprint

Internal Coaching

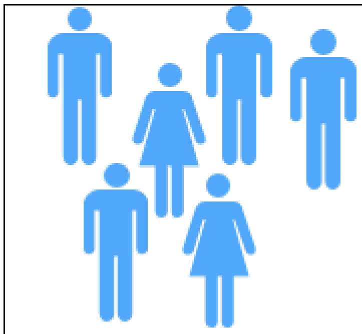
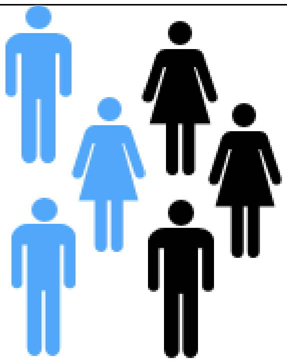
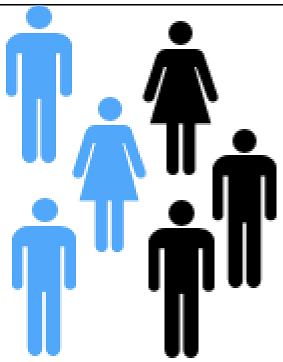
- Teams that are running well do not need to be split
- Coaches can be hand-selected for new teams
- Coaches can be moved from team to team

- Adopting Scrum is one thing, but spreading it across the organization is another.
- If all in transition approach is not chosen, the successes of first Scrum teams must be leveraged and built upon as Scrum is introduced to other teams.
- There are three patterns to do this.

Split and Seed

- Split and seed pattern can be used after the first couple of teams have adopted Scrum and run at least few sprints.
- By now, these teams are familiar with Scrum, develop working software during every sprint, and are working well together.
- One functioning Scrum team is split into two, with each half of the original team forming the new team.
- New team members are added to these two teams as needed.
- These new team members will have guidance from experienced Scrum team members from the original team.
- After these new teams are fully functional and have experienced Scrum with few Sprints, they can be further split to seed other teams, and the cycle continues.

Illustrative Example

 Original Team	• Imagine these six members are the first to adopt Scrum Adoption in an organization: ○ Adam ○ Dave ○ Anita ○ Sam ○ Samantha ○ John • This team has worked together for 4 sprints. • The team has delivered working Product Increment in the sprints and is comfortable with how Scrum works.
 First Team	• This team is split into two teams • First team consists of ○ From original team ▪ Adam ▪ Anita ▪ John ○ New team members ▪ Smitha ▪ Caroline ▪ Mohammed • These six members now form the first team. • Adam, Anita, and John will guide the new team members on how Scrum works.
 Second Team	• In the second team, we have: ○ From original team ▪ Dave ▪ Samantha ▪ Sam ○ New team members ▪ Gloria ▪ Raj ▪ Eugene • This second team will be guided by Dave, Samantha, and Sam from the original team

Reasons to prefer Split and Seed

You can add teams more quickly than with most other approaches

- By moving at least 2 team members from a Scrum Team to a new team, a team of 8 members in the original team can be split into 4 teams easily, with each team having 2 experienced members from the original team. After about 5 to 6 sprints, these 4 teams, that is, 32 team members (4 teams each with 8 team members) can be used to seed 16 more teams (2 members each will be part of these 16 teams).
- This seeding pattern can continue easily across the organization.

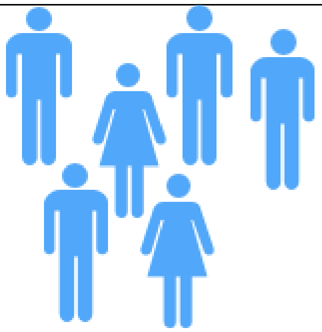
Each team has someone with Scrum experience to help guide them

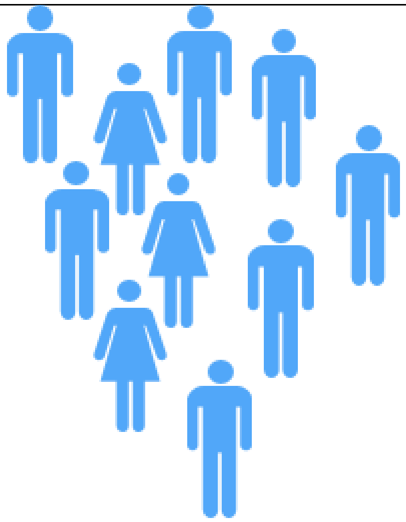
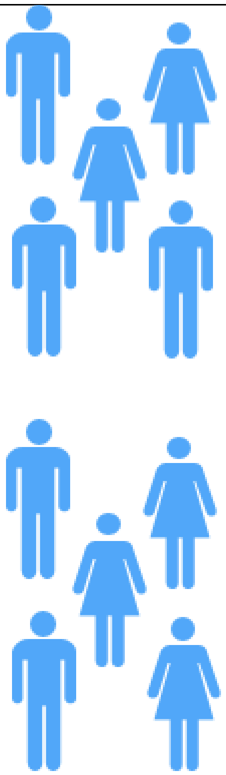
- Only the original team will have some difficulty in transitioning to Scrum as they don't have prior experience.
- The subsequent teams will be guided by the team members from the original team who will have experienced at least 2 to 3 sprints.
- As the new team members are learning Scrum from other team members (not from managers or external trainers or coaches), their discomfort about transitioning to Scrum will vanish quickly.

Grow and Split

- Grow and split is a variation of the split and seed approach.
- A large team is split into two teams with 5 to 8 team members in each, the desired team size of Scrum.
- New teams will continue to sprint for a minimum of 2 to 3 iterations. During these 2 to 3 sprints, additional team members will be added to experience Scrum with the guidance of existing team members.
- Once these new teams become larger and have completed 2 to 3 sprints comfortably, these teams are split further and the pattern continues to cover the remaining teams in the organization.

Illustrative Example

 Original Team	• This original team, which is the first one to initiate Agile transition, comprises of: ○ Ahmed ○ Bob ○ Cathy ○ Dinesh ○ Matt ○ Oliver

	• This team works for 2 to 3 sprints.
 Original Team + new members	• Four new members are added to this team: o Samantha o Sameer o Rahim o Xavier • Existing team members from the Original team will guide these new members. • This Original Team plus new members will continue to sprint for 2 to 3 sprints.
 Original Team split into 2 Teams	• Now the original team is split into two teams • Team 1 consists of: o Ahmed o Bob o Cathy o Oliver o Sameer • Team 2 consists of: o Dinesh o Samantha o Matt o Rahim o Xavier

Reasons to prefer Grow and Split

You don’t have to destroy any existing teams

- In split and seed strategy, teams that are just beginning to work together well are split into other teams. Essentially, we are breaking up a good team, and this must be done with caution.

- Grow and spilt overcomes this breaking up of good teams because we split when the teams are a bit larger. The remaining team members can continue to work together well as in the past.

Team members feel more continuity from sprint to sprint

- In split and seed strategy, teams are continuously split and new members are added.
- In grow and split, the team is divided only when it has become large enough for the split. This enables team members to work together longer without a feeling of discomfort.

Internal Coaching

- From the original team, team members who truly understand Agile and excel at Scrum are identified as internal coaches.
- These internal coaches will then work with other teams:
  - Attending sprint planning and review and retrospect meetings
  - Attending daily scrum, perhaps once each week
  - Available, for example 2 hours a day, to aid the new team members

Reasons to prefer Internal Coaching

Teams that are running well do not need to be split

- In both approaches presented above, functioning teams are split, which can be a major drawback.
- When internal coaches are used, teams can stay intact for a longer duration with minor disruptions to team composition.

Coaches can be carefully selected for new teams

- When the complete team is coached, a few might excel and a few might not.
- This approach helps us to identify those who excel and keep them ready for future coaching assignments for new teams that might be formed in the future.

Coaches can be moved from one team to another

- A coach associated with a team might not see any new areas for improvement after a while.
- In such situations, new coaches, that is, a fresh pair of eyes can help identify new ways of working and areas of improvement.
- These new coaches move from team to team with new ideas.

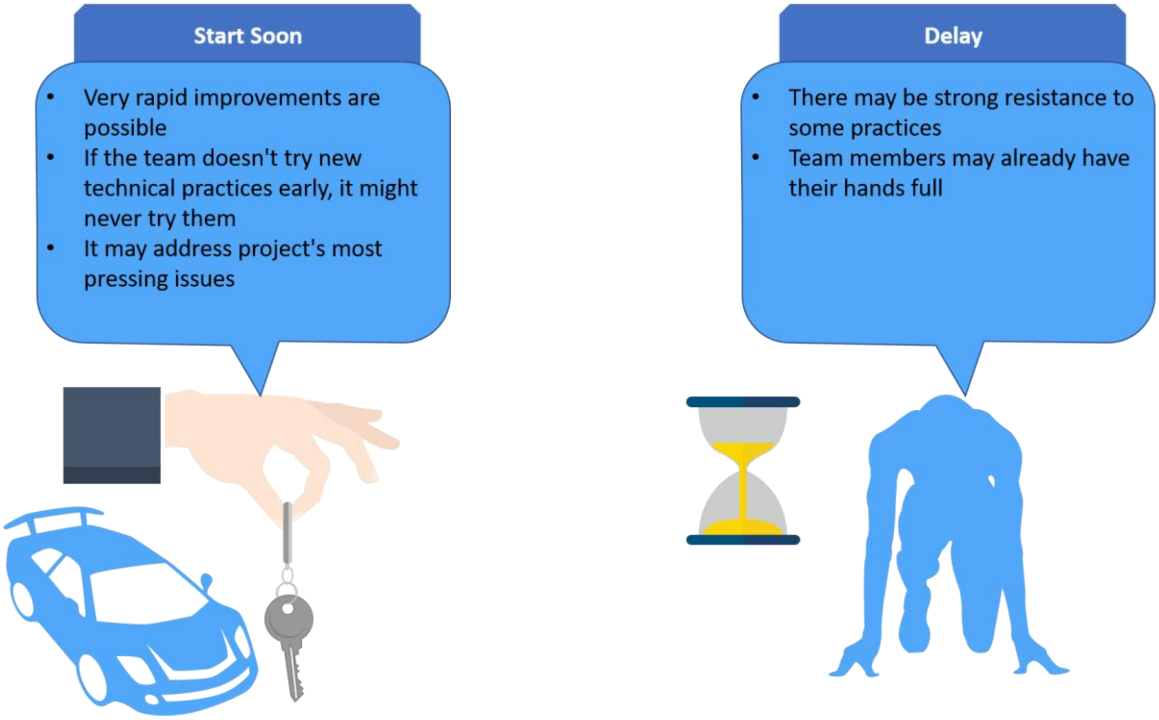
Choosing the right approach

- Split and Seed
    - This is the fastest way to spread Scrum through an organization if it is in a hurry.
    - The process can be accelerated, that is, instead of using two or more experienced team members to pollinate new teams, less experienced members can be deployed in new teams, creating more new teams.
    - This may not work in technology and domain-specific teams. For example, seeding .NET Teams with Python programmers who have experience in Scrum won't work.

- Grow and Split
    - This is the most natural approach as it can happen with no one intervening to spread Scrum
    - This approach can be used when there is no urgency to push Scrum
    - This is ideal when the team size is growing because of project needs

- Internal Coaching
    - This can be used when the group is large enough that good practices can be spread to other teams
    - This is ideal when teams can't be split due to project constraints or technology/domain limitations
    - This approach is recommended when there are enough internal coaches available (or external coaches can be brought in, if required)

Introducing new technical practices



Start Soon

- Very rapid improvements are possible
- If the team doesn't try new technical practices early, it might never try them
- It may address project's most pressing issues

Delay

- There may be strong resistance to some practices
- Team members may already have their hands full

- While working with Scrum, a few technical practices must be followed.
- These new technical practices include:
  - Pair Programming
  - Code refactoring
  - Automated testing and so on
- During transition to Scrum, organizations face the dilemma of when to introduce these technical practices to team members.
- In this section, we look at two approaches.

Start Soon

- The thought process behind this approach is that everything should start with the technical practices.
- Technical practices such as automated testing and pair programming will drive Agile as naturally as possible.

Reasons to Start Soon

Very rapid improvements are possible

- Technical practices can provide some quick wins to teams and organizations.
- For example, pair programming helps team members to discover coding mistakes from other programmers and catch them before the code goes for testing. Similarly, code refactoring can clean up and streamline code

- If the team doesn't try new technical practices immediately, it might never try them
- Teams may often think that if they continue doing Scrum, they may deliver working software frequently and achieve agility anyway, even without using technical practices.
- By not using or trying new technical practices, teams will lose many improvements that are otherwise possible. For example, teams may prefer doing manual testing instead of the new technical practice of automated testing. This will never improve the team's ability to deliver high-quality software quickly and frequently.

It may address the project's most pressing issues

- Teams may be already facing a few pressing issues such as poor quality, over-engineered and design-heavy solutions, long and missed delivery cycles, etc.
- Sometimes, team composition can cause issues with how team members write code. For example, a project team may have 5 fresh college graduates with 2 senior experienced tech leads. These 2 tech leads might be busy, overworked, and may not guide these 5 fresh college graduates adequately. Pair Programming can be useful in this case.

Delay

- In this approach, teams are left alone longer and given time to discover new technical practices that work best for their project requirements and environment.

Reasons to Delay

There may be strong resistance toward some practices

- Introducing new technical practices is one of the most difficult challenges faced during Agile transition.
- Teams and individuals and certain functions may be extremely reluctant to try new things.
- For example, in a few organizations, the Quality Centre of Excellence may be a well-established function that independently monitors Quality Control of all projects. This team may object to automated testing practices.

Team members may already have their hands full

- Team members must learn these new technical practices, but they may not have enough time due to already tight project deadlines and schedules.

- Within a pressure situation, if pushed to learn new technical practices, team members may not try or even breakdown.
- Therefore, it is highly advised that you give enough time to team members to learn these new technical practices.