simplilearn

your pace, your place

Successful Scrum Adoption

Guidance for Teams – Team Structure

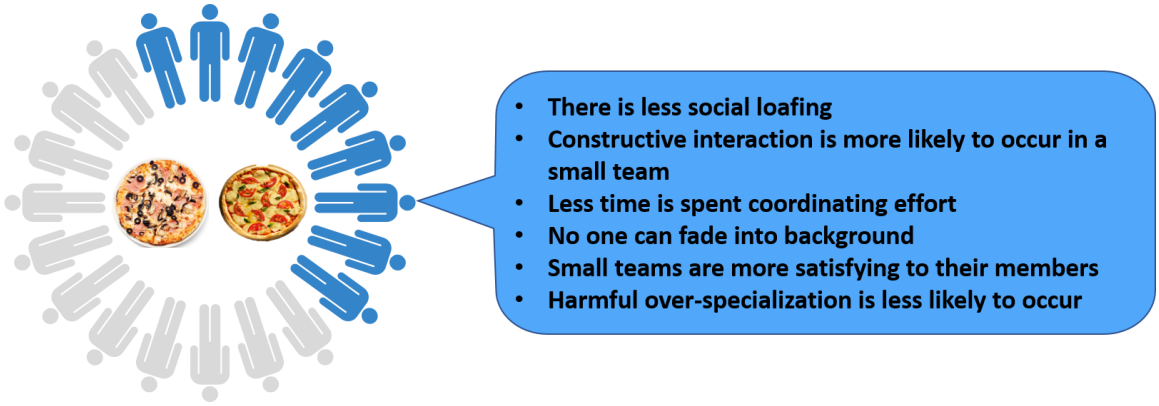
- Let us look at two crucial factors in Team Structure
  - Why small teams are better
  - Team orientation toward delivery of end-to-end, user-visible functionality

Two Pizza teams

- Generally accepted advice for ideal Scrum team size is ***five to nine*** team members, that is, a team that can have two pizzas for their lunch.
- With a larger team, there are communication overheads. Larger teams have a few advantages as well: loss of one key member doesn't impact the team and the team may have diverse members with different skills, experience, etc.

Advantages of small team

- Let us look at few advantages of having a small team, that is, Two pizza teams.



There is less social loafing

- Some team members may put less effort as they expect other team members to pick up the slack. This is called social loafing.
- A lot of research and study shows that individual effort is inversely related to team size, that is, individuals put forth lesser effort in bigger teams.
- With a small team, the social loafing is reduced.

Constructive interaction is more likely to occur in a small team

- Research shows that teams with 10 to 12 members have difficulties in establishing feeling of trust, mutual accountability, and cohesiveness.

- A small team interacts constructively as a feeling of trust is established easily when team size is small.

Less time is spent in coordinating effort

- Small teams spend less time coordinating the effort of team members.
- The coordination effort (for example, organizing a meeting, initiating a discussion, etc.) on the team and on the project would be less with a small team.

No one can fade into background

- In a large team, during team meetings and interactions, there is a possibility that senior or influential team members might hijack the agenda and keep the limelight on themselves, that is, they might speak on behalf of the team and not allow others to speak.
- In a small team, every member gets opportunity and recognition.

Small teams are more satisfying to their members

- A team member's contribution will be more visible and meaningful in a small team.
- Research has shown that participation in large teams is less satisfying to its members.

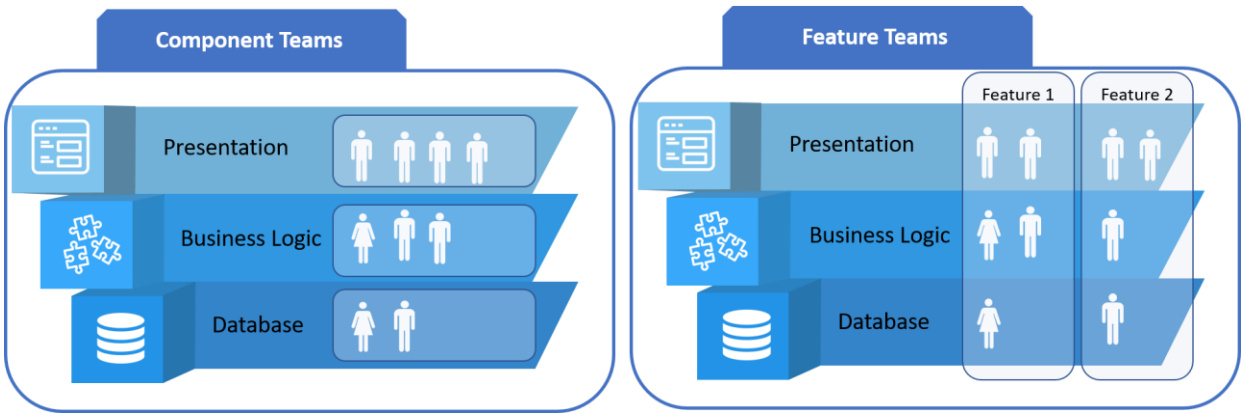
Harmful overspecialization is less likely to occur

- In a large project, individuals most likely will take specialist roles.
- This causes team members to work in silos and reduces the amount of learning.

Small team productivity

- A study conducted at QSM by Doug Putnam on 491 projects with team sizes from 1 to 20 found that the smaller the team size, the more productive each member was.
- He also found that smaller teams complete projects with lesser effort.
- Research shows that teams with 5 to 7 members complete a project in lesser time when compared to teams of other sizes.

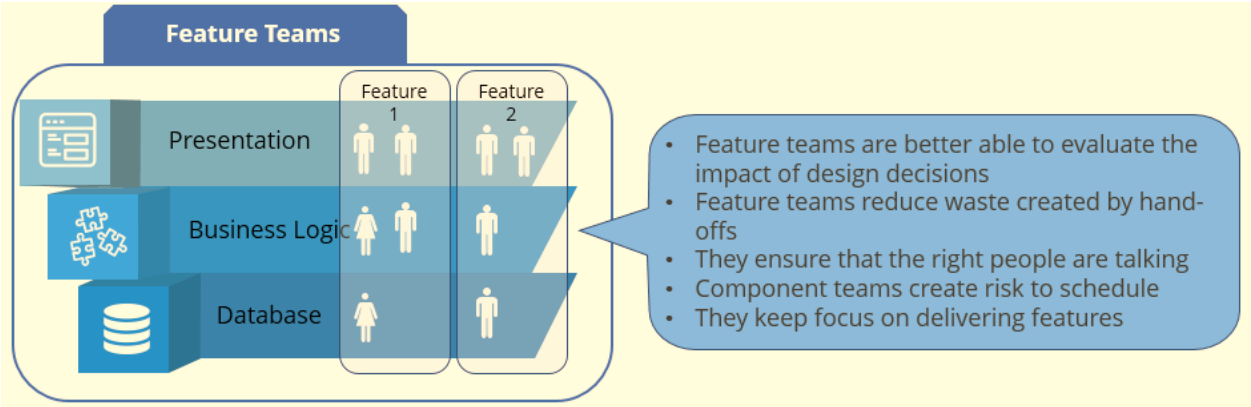
Component Teams and Feature Teams



- Let us take a typical client server application. It will have few tiers such as Database tier, Business logic tier, desktop client, and web client presentation tier.
- In a traditional project setup, you may find dedicated teams for each tier – Database team, Business logic team, web client team, etc.
- These teams that specialize on specific technology (or tier) are called ***component teams***.
- These component teams have the following problems:
    - Reduced communication between team members in each layer
    - A feeling that design by contrast is sufficient
    - Ending projects without potentially shippable product increment
- ***Feature teams*** work across all layers of the application. As a team, they are responsible for end-to-end delivery of tested and working features.

Advantages of Feature teams

- Let us explore the advantages of feature teams.



Feature teams are better able to evaluate the impact of design decisions

- At the end of the Sprint, a feature team would have developed the end-to-end tested feature by working across ***all*** technology stack layers.
- This helps the team to evolve the design over the course of Sprints.
- Team members can also validate the usability of their features (Are users happy with the feature that is offered?) and technology design (How does this feature work across all technology stack layers?)

Feature teams reduce waste created by hand-offs

- When component teams are involved, there is a hand-off, that is, knowledge transition between different teams. This creates waste (that is, team is not working on project deliverable during this transition time).
- In component teams, there are possibilities that expected feature will not be fully developed or wrong feature will be developed.

It ensures that the right people are talking

- Because a feature team works across technology stack layers, the right people (that is, team members working on required layer) will communicate with others during daily interactions.

Component teams create risk to schedule

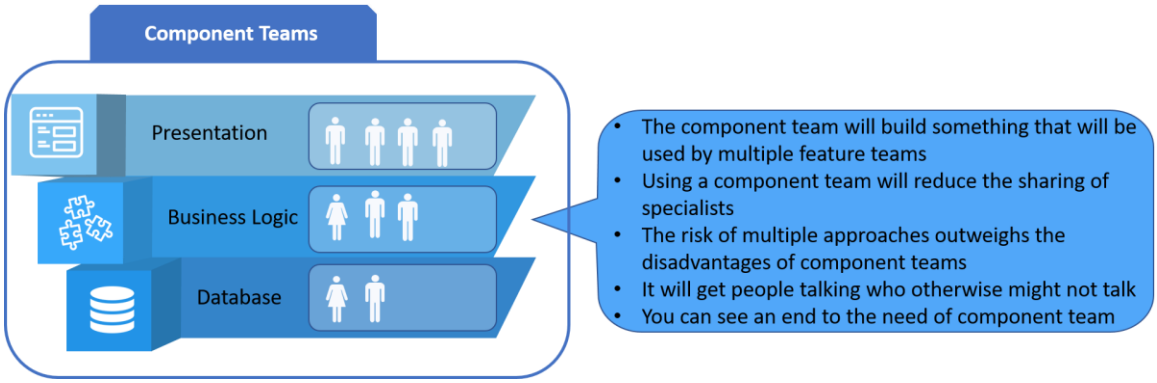
- When component teams are involved, there would be a lot of integration requirements. For example, Business logic layer integration with Database layer.
- These integrations create dependencies. These create a risk to project schedule.

It keeps focus on delivering features

- Scrum is all about delivering potentially shippable product increments for each Sprint.
- With the focus on delivering features, teams can deliver usable features to users. Thus, Business value is delivered much faster.

When to use Component Teams

- Whenever possible, feature teams must be preferred over component teams.
- However, in few situations as given below, component teams might be useful.



The component team will build something that will be used by multiple feature teams

- If the component to be built is to be used only by one team, then the feature team itself can try to build it.
- In case the component is used by multiple feature teams, then a component team can be used to build that specific component.

Using a component team will reduce the sharing of specialists

- In a few organizations, there could be shared teams for Database services, Business Logic, and middleware. These team members most likely work on multiple projects—working part-time on each project.
- In such situations, consider creating a dedicated component team with full-time participation from those shared teams.

The risk of multiple approaches outweighs the disadvantages of component teams

- When more than one feature team is working on a shared component, there are two risks:
    - One feature team may implement a solution that is different from that of another team
    - Teams may lack a cohesive vision and this may cause integration issues
- In such situations, a dedicated component team will be useful.

It will get people talking who otherwise might not talk

- People tend to talk more to their teammates than outsiders.
- Component teams can be created to group team members who might otherwise not talk among themselves, keeping in mind the team's boundaries.

You can see an end to the need of component team

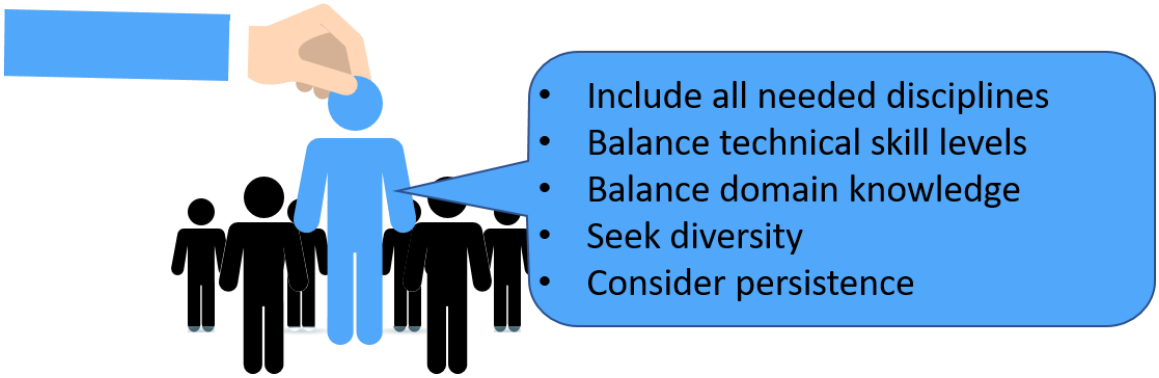
- Component teams should develop the required functionality and disband.
- These teams should not exist throughout the project.

Decision makers on team structure

- Decisions on team structure are to be taken by teams ***themselves.***
- In many situations, due to team's lack of authority and exposure or contractual obligations, functional managers, Project Managers, Scrum Masters, or those leading Agile transitions can make decisions on how the teams are organized.

Getting the right people on team

- Some of the factors to be considered when organizing a Scrum team are as follows.



Include all needed disciplines

- Scrum teams are cross-functional.
- All skills, from idea creation to implementation of a feature, must be present in the team.
- Initially, this may make large teams. But that is OK. After mastering Scrum, these team members can be moved to other teams.

Balance technical skill levels

- Based on team size considerations, there must be a balance of skill levels on a team.
- For example, having all senior developers or junior developers will not work.
- A right mix of senior, mid, and junior skill levels should be used.

Balance domain knowledge

- Like balancing the technical skill levels, domain knowledge expertise should also be balanced.
- This means that teams should have people with deep knowledge of domain.

Seek diversity

- Gender, race, and culture provide diversity options.

- In addition, how individuals think about a problem, how they make decisions, how much information they need before deciding, etc. also help to build a diversified team.

Consider persistence

- Try to keep team members working together on same project.
- When forming a team, check with individuals about their commitments and how long they can serve on the project.

Multitasking

- Multitasking is attempting to work more than one project or more than one thing at once.
- Multitasking may create an illusion of good progress, but it creates far more problems.
- Research shows that the total amount of time an individual spends on a task goes up when that individual has two tasks to work on. Research also clearly shows that with just three tasks, the amount of time an individual spends on a task decreases drastically.
- But in some cases, multitasking can be helpful.

Remember that a person who is working on multiple projects, that is, multitasking is most likely to get less total work done than if the same individual was dedicatedly working on one project only.

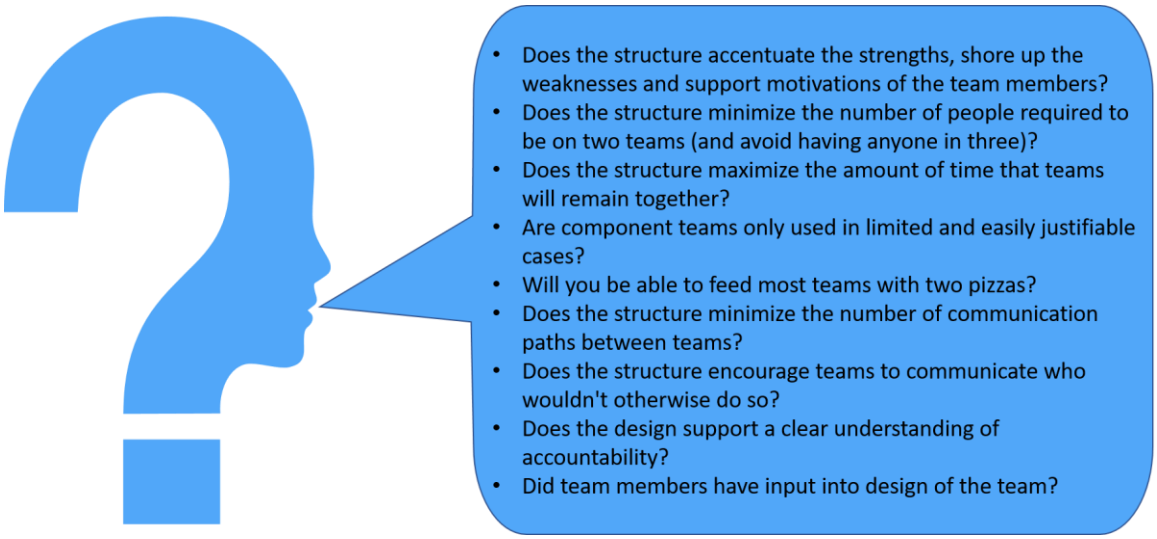
- For example, a shared team member such as Database Administrator or User Interface Designer working on more than one project may have to devote enough time to each project so that there are no delays due to multitasking.
- The following guidelines are to be kept in mind when using Multitasking in Scrum teams:
    - In general, multitasking is to be avoided.
    - Multitasking is acceptable if an individual can't be fully utilized on one single project.
    - Instead of allowing everyone to multitask, it is better to allow only a few to multitask.

Techniques for better multitasking

- Don’t start a new project until it can be fully staffed (or at least one team is ready)
- Include ramp-up and wind-down time in enterprise plans
- Institute simple rules such as “No one can be assigned to more than two projects”
- Go slow but go, that is, start small

Guidelines for good team structure

- These questions should be asked for a current or proposed team.
- These questions should be asked iteratively.
- Ask one question and change the structure based on answer.
- As the structure changes, ask the questions again, until all answers are “Yes.”



Guidance for Teams – Teamwork

- Scrum teams succeed together and fail together.
- Becoming a truly high-performing Scrum team requires a concerted effort toward continuous learning and improvement.

Embracing whole-team responsibility

- Scrum teams must come to terms with whole-team responsibility.
- Team shares full-team responsibility for the overall product and all aspects of its development.
- Because of the shift to whole-team responsibility, individuals will often be called on to perform work outside their specialties.
- In some organizations, this shift might be difficult. But until this shift occurs, teams will find it difficult to complete their backlog items together.
- With whole-team commitment, the team member who is ahead of schedule will help the one who is lagging.

Rely sparingly on Specialists

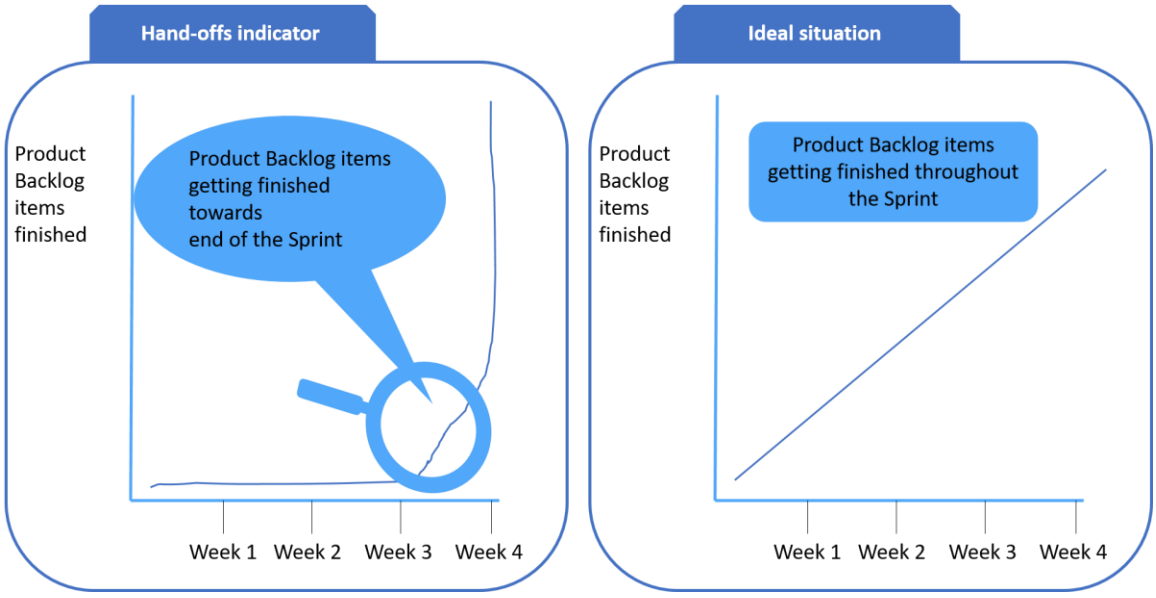
- Scrum teams may provide a misconception that every team member must be a generalist. But in a few situations, we might need dedicated specialists.
- It is better to have a few generalists.
- These generalists will enable the specialists to specialize.

Do a little bit of everything all the time

- In traditional projects, teams will be mostly used for sequential development processes with hand-offs between specialists.
- Each of these overheads (or Knowledge transitions) include some overheads, such as meetings, documents to review, and sign off. Added together, these overheads might account to a considerable amount of project time. During this time, no productive work is being done by the team.
- Scrum teams should look to minimize these hand-offs because there will always be some hand-offs.
- During a hand-off, a Scrum team should collaboratively discuss how to get started and how much work has been done with the team members who are going to work on the project.
- Rather than doing an analysis phase, programming phase, followed by testing, a little of each of those activities should happen at all times.

Don’t wait until the end of the Sprint to finish everything

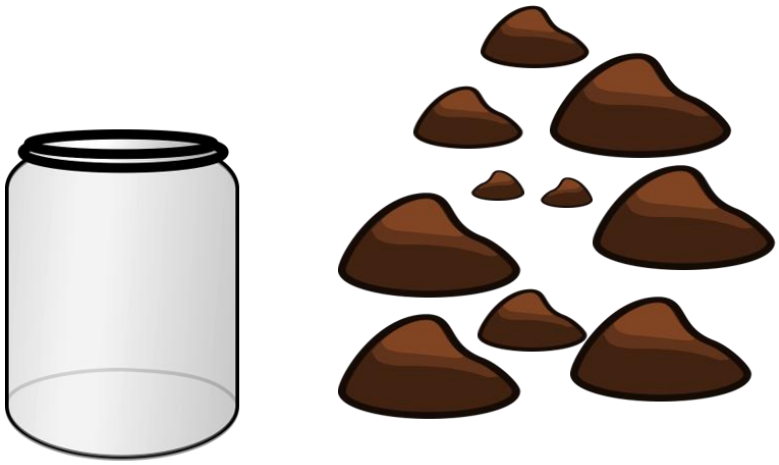
- One indication of continuing hand-off work in a Scrum project is emergence of a trend—Product Backlog items are not finished until the last few days of the Sprint.
- Testers complain that they don’t have enough time to test as they are given only the last 2 or 3 days of the Sprint to test. This is indicated by Hands-off indicator in figure below.
- The team should create a chart showing number of Product Backlog items finished each day in the Sprint. This is indicated by the Ideal situation in the figure below.



Mixing Product Backlog Items toward Sprint commitment

- During Sprint Planning, the team picks up the high priority items, estimates, and then commits to those items that it can deliver with available capacity.
- Imagine a situation where capacity still available, but the next high priority item would not fit in that capacity. The team can look at lower priority items so that their capacity is filled in.
- The series of figures below indicate this using a glass jar analogy.

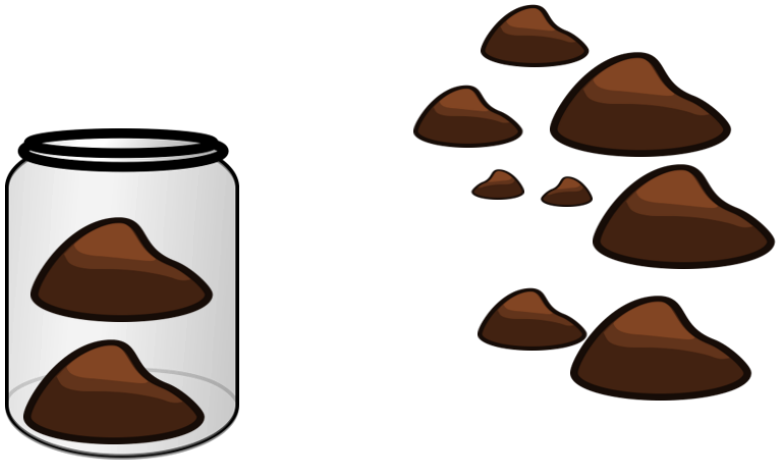
- At Sprint Planning meeting:



Team Capacity

Product Backlog items

- Team picks up high priority items, estimates them, and adds to Sprint Backlog.



Team Capacity

Product Backlog items

- Now, team tries to take the next high priority item. That one is beyond the team's remaining capacity. The team can try to break that large item into smaller ones. If that's not possible, the team can pick other smaller ones – till the remaining capacity is filled in.

Team Capacity

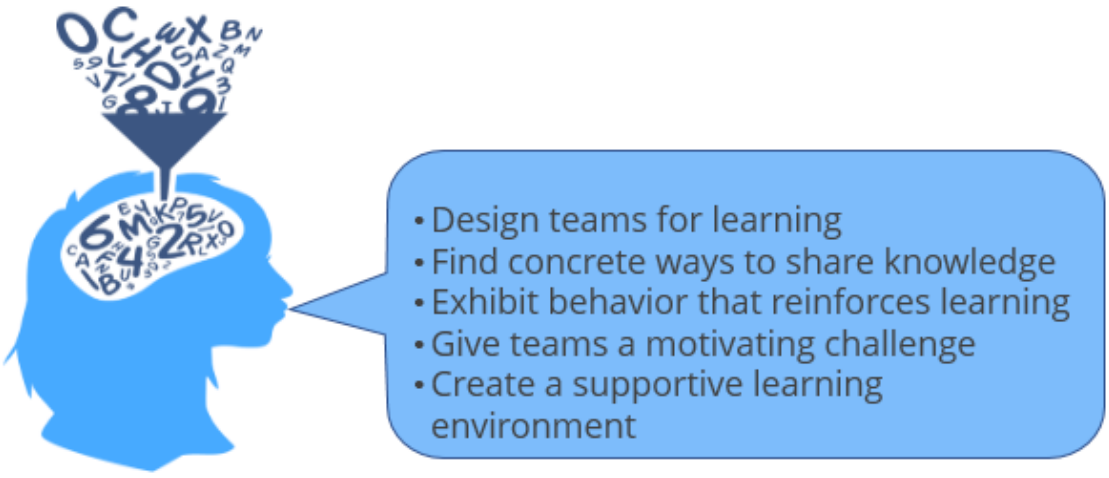
Product Backlog items

Fostering Team Learning

- To become a truly high-performing team and to realize all the benefits of Scrum, teams must proactively seek out new ways to learn and share knowledge.
- The most effective teams and their leaders take a very active role in optimizing the rate and significance of learning.

Learning Conditions

- The following are the five necessary conditions that must exist for team learning to occur.



Design teams for learning

- Team composition is influenced by functional and/or Project Managers.

- They should ensure that when a team is formed, there is enough diversity so that new and creative ideas are generated.
- The first thing a manager can do is to keep the team together as long as possible. If team composition changes frequently, teams might have to start all over again.

Find concrete ways to share knowledge

- For organizations to be successful, teams must have concrete ways to share what they have learned, not just with other teams but with the rest of the organization.
- Scrum teams can achieve this using in-built communication forums.
- Daily Scrums can include other team members as silent participants. Sprint Reviews can be attended by larger audiences and relevant stakeholders.
- Information radiators also help to share the knowledge.
- Teams can also communicate with each other via informal channels.
- In a few organizations, they may form practice communities to share solutions and problems.

Exhibit behavior that reinforces learning

- Functional managers, Product Owners, and other executives are seen as leaders by team members. Team members try to see how they can model their communications and learning based on these leaders.
- It is essential for these leaders to demonstrate learning behaviors that they would like to see on their teams.
- These leaders should also ensure the following to foster and reinforce learning:
    - **Be accessible**
        - Leaders must be accessible to team members.
        - This will create a welcoming environment in which team members are not afraid to reach out when required.
    - **Ask for input**
        - Leaders can ask team members for input so that their opinions are valued.
        - Team members' inputs can be sought even in decision-making. This creates trust and confidence.
    - **Serve as a fallibility model**
        - By admitting their own mistakes, leaders demonstrate to others that bugs, bad decisions, and problems can be discussed without repercussion.

Give teams a motivating challenge

- Team members are influenced by how a challenge is presented to them and respond accordingly.
- For example, if a Product Owner says "No matter what, I want these features completed in this Sprint" with an angry voice, team members may feel offended. Instead, if the same Product Owner asks politely, "What can we do so that these features are completed in this Sprint?" team members may respond differently.

Creative a supportive learning environment

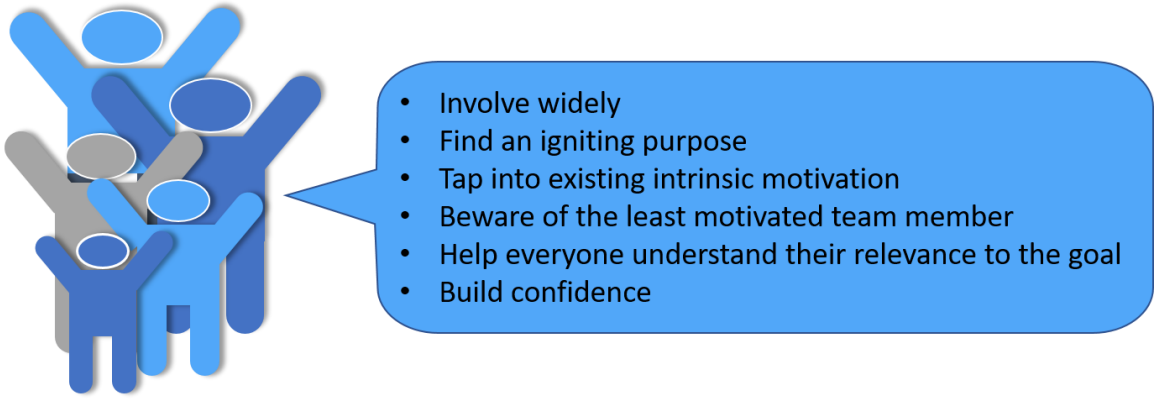
- It is up to leaders and managers in aspiring Scrum organizations to create a supportive learning environment for their teams.
- The following characteristics are exhibited by organizations that support a learning environment:
    - **Psychological safety**
        - Product Owners, functional managers, and Scrum Masters must find ways to create a feeling of safety around the activities to be carried out by team members.
        - When transitioning to Scrum, existing expertise is disrupted. Therefore, it is essential for the leaders to create psychological safety so that resistance to the transition lessens.
    - **Appreciation of differences**
        - Individuals on a team need to appreciate rather than attack differences.
        - Teams need ***deviants*** who can help the team by challenging the tendency to want too much homogeneity, which can stifle creativity and learning.
    - **Openness to new ideas**
        - Scrum teams are often asked to work on difficult challenges.
        - To meet these challenges, team members have to look beyond current methods.
        - An openness to ideas is essential to promote the handling of these challenges.
    - **Time for reflection**
        - Teams also need time for learning apart from their current work.
        - Real-time learning is the best way for a team to learn.

Eliminate Knowledge Waste

- **Knowledge waste** refers to team members losing opportunities to learn or learning less than they could.
- Knowledge waste can be divided into three categories:
    - **Scatter** refers to anything that breaks the flow of work. At project level, scatter occurs when the team is interrupted from what it's doing or when team composition changes (for example, a new person is added or existing member is removed). Causes for scatter are barriers to communication (teams may be located at different locations) and tools.
    - **Hand-offs** occur when one team hands over knowledge and responsibilities to another team. Hand-offs cause delays.
    - **Wishful thinking** refers to making decisions without adequate information to support those decisions. Planning projects and committing deadlines hoping that the project will run exactly as planned is an example of wishful thinking.

Encourage Collaboration through commitment

- A self-organized team working as a unit instead of collection of individuals must constantly reenergize and focus on shared goals.
- For this to happen, the team members' commitment to the purpose and to each other must be renewed throughout the project.
- The following are the ways to nurture this kind of commitment.



Involve widely

- Involve project team members in as many project activities as possible.
- For example, all team members can be involved in writing Product Backlog User Stories.

- Team members will be fully engaged and committed to the project when they are aware of the big picture about the project and product they are developing.

Find an igniting purpose

- Igniting purpose involves finding something that people find exciting and interesting and worth engaging with.
- It must be something that excites and interests the team members so that they are anxious to be part of it.

Tap into existing intrinsic motivation

- According to https://www.verywell.com, intrinsic motivation refers to behavior that is driven by internal rewards.
- The organization should feed team members' existing motivation.
- A project that is structured in such a way that everyone's unique personal goals are aligned with project goals will generate the desired commitment.

Beware of the least motivated team member

- One unmotivated team member can bring down the whole team with him/her.
- Organizations and teams should identify such individuals and ensure that their concerns are heard. They must work out ways to motivate them.

Help everyone understand his or her relevance to the goal

- It is difficult for team members to fully engage and commit to a project's goals if they feel that their contributions are not significant.
- The Product Owner can help the team members feel relevant to the goal by appreciating the work done by the team members.
- Team members can also appreciate the effort of each other.

Build Confidence

- When presented with a challenge, team members want to feel that they can achieve it.
- Confidence comes from belief in the team and team members believing in themselves.

Guidance for Teams – Leading a self-organizing team

- Let us look at how leaders, managers, and change agents can influence the self-organizing path of a team or company.
- There are three conditions for self-organizing to exist.

Influencing Self-organization

- Self-organizing teams are ***not free*** from management control.
- Management decides what products the team will build and who will work on which project.
- Self-organizing within the boundaries and constraints laid by the management and working around the challenges is the job of a Scrum team.
- This needs a balance between command and influence. For this, let us look at three conditions that can impact how teams organize and perform.

Containers, Differences, and Exchanges

- **Containers**
  - Container is the boundary within which the self-organization occurs.
  - Container can be the office location (physical container) where the team operates. Containers can also be behavioral, organizational, and conceptual. For example, everyone working on a mobility project, all contractors on the project, etc.
- **Differences**
  - The team members within a container can have differences among themselves.
  - These differences can involve technical expertise, domain knowledge, power, gender, race, etc.
  - These differences influence how the team self-organizes.
- **Exchanges**
  - Exchange is an interaction between team members within a container where one or more members are changed or influenced by that interaction.
- By ***adjusting containers, amplifying or dampening differences, and altering exchanges,*** leaders can influence how a team can self-organize.

Adjusting Containers

- The following ways can be used to adjust the containers:
  - Change the number of people on the team
  - Change team membership, that is, who works on the project
  - Introduce a new container – such as a specific one for Oracle expertise

- Give the team more or less (as the case may be) responsibility
- Change the team's physical space – remove or move cubicles, move team members to the same location, etc.

Amplifying or dampening differences

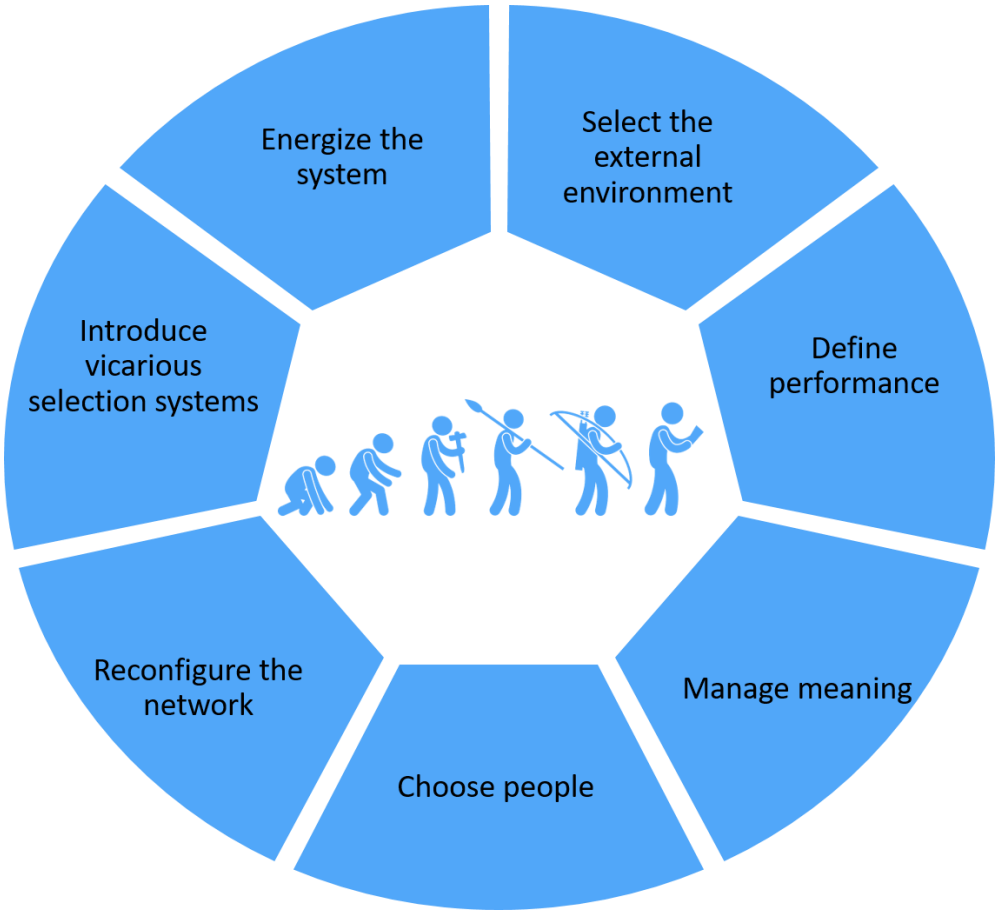
- The following ways can be used to amplify or dampen the differences.
  - Introduce a new team member with more significant power, experience, knowledge, etc.
  - Ask challenging questions (What could go wrong with this approach? What could make us regret this decision?) to ensure that different viewpoints are heard
  - Change the team's decision-making style
  - Encourage dissenting viewpoints

Altering exchanges

- The following ways can be used to alter exchanges.
  - Add or remove people from an exchange
  - Formalize or deformalize an exchange
  - Change how an exchange occurs (for example: face to face, documented, etc.)
  - Change the frequency of the exchange

Influencing evolution

- Organizations evolve over periods of time in response to environmental factors, competitive forces, strengths and weakness of employees, and other influences.
- There are seven levers a leader can use to guide an evolving organization.



Select the external environment

- Self-organization and evolution occur in response to the environment in which the team works.
- Leaders can have a significant amount of influence on that environment.
- Leaders determine the organization's approach to innovation.
- Leaders also control the type of projects an organization is going to work on.

Define Performance

- Over millions of years, life has evolved to fit into the environment. For example, cats have evolved over millions of years into various forms such as domestic cats, tigers, cheetahs, etc. to fit into the environment. Organizations also evolve to fit into their environment.
- As evolution progresses, traits that will help the species to survive the environment will be developed. For example, giraffes developed long necks so they can reach leaves at top of a tree and continue their survival.
- In case of an organization, the leaders will define the traits that help the individuals and organizations to survive.

- Promotions and public praise are the traits an agile team exhibits. These traits are to be encouraged for the teams and organizations to survive.
- An organization that is looking at long-term performance will most likely invest in training, support working at a sustainable pace, be willing to allow employees to try new things, etc.

Manage meaning

- Individuals in a self-organizing system evolve in response to the messages they receive.
- These messages can be internal (generated within the organization) or external.
- Managers and leaders manage the meaning of these messages by providing context to help team members interpret the messages.
- Let us see an example.
    - In an organization, the Vice President says, "We will move more toward automation. Automation is the future" during a town hall meeting with all employees.
    - This message can be interpreted in multiple ways, such as:
        - We need to use more automation tools
        - There could be job losses as automation can replace manual tasks
    - The VP further adds, "We will provide opportunities for everyone on automation training. We will also consider requests to move individuals from one project to another. But there will be no layoffs." ***This is the meaning that leader has added to the message*** so the message is clear.

Choose people

- This is already covered in "Adjusting Containers" and "Amplifying or dampening differences."

Reconfigure the network

- This is already covered in "Altering exchanges."

Introduce vicarious selection systems

- Over the years, in an organization, products that generate profits displace products that do not generate profits. Team structures that generate profitable products drive out the structures that don't.
- Leaders and managers can introduce vicarious selection systems to improve the rate of evolution.

- A vicarious selection system is a process for selecting desirable projects, products, or behaviors but that does so without the lengthy feedback of a market-driven selection system.
- For example, one employee can nominate another employee for a small cash reward.

Energize the system

- Teams and organizations rely on energy.
- Teams or organizations will suffer from entropy when energy is not pumped in.
- Managers and leaders provide the energy by challenging the employees.
- When a team is inspired to accept a challenge, it self-organizes around how to achieve the challenge.
- An elevated clear goal can drive a lot of energy in the team.
- Teams can come up with a small project charter providing a vision about why the project is executed. This project charter creation involves:
    - Summarizing the project in one or two sentences
    - Designing the box the product would ship in (even if the product will not ship in a box)
    - Writing a one-page product description
    - Writing an imaginary press release statement that would accompany the product launch
    - Writing a product review that you would like to see in a magazine