Domain IV Team Performance

The PMI-ACP Exam consists of 120 questions which can be categorised into seven domains. The fourth domain: Domain IV Team Performance is the knowledge about "creating an environment of trust, learning, collaboration, and conflict resolution that promotes team self-organization, enhances relationships among team members, and cultivates a culture of high performance" (source: PMI-ACP Examination Content Outline).
Domain IV Team Performance accounts for 16% of all questions in the PMI-ACP Exam (i.e. ~19 questions among 120 PMI-ACP Exam questions)

According to the PMI-ACP Exam Content Outline, Domain IV Team Performance consists of 9 tasks grouped within 3 sub-domains:

Team Formation:

- The team works together to establish ground rules and processes to strengthen sense of belonging and create a shared goal of the team members.
- Form the team with members who possess all the necessary skills (interpersonal and technical) to deliver the intended outcomes and values of the project.

Team Empowerment:

- Create a high performing team in which members can perform as generalizing specialists carrying out cross-functional tasks.
- Empower team members to make decisions and to lead in order to create a self-organizing team.
- Understand motivators and demotivators of the team and individuals to ensure high team morale.

Team Collaboration and Commitment:

- Make use of collaboration tools and colocation to enhance communication within the team and between team and stakeholders.
- Shield the team from external distraction and pressure to ensure performance.
- Align the goals of the project and the team members with a shared project vision.
- Measure the team velocity by tracking work performance in previous iterations to allow more accurate forecasts.

Below is a collection of the key knowledge addressed in Domain IV Team Performance and the 9 tasks related to the domain:

- Individuals and interactions over processes and tools - i.e. in Agile project management, the team members and their interaction are considered far more valuable than following pre-defined processes or toolsets.
- Businesspeople and developers must work together daily throughout the project / Build projects around motivated individuals / Agile processes promote sustainable development / Self-organizing teams / The team reflects on how to become more effective
- **Team Formation Stages**
  - Tuckman model (Tuckman's stages of group development)
    1. **Forming -** the team is formed, everyone behaves independently
    2. **Stroming -** disagreements arise between team members
    3. **Norming -** team members accept each other by emphasising the team goal
    4. **Performing -** the team is highly motivated and efficient
    5. **Adjourning -** tasks completed
  - At stage 4, the team is considered to be most efficient and best performing. However, not every team goes through every stage of the Tuckman model, some may stay at stage 2 and jump to stage 5 without going through stages 3 and 4.
- **Building Empowered Teams**

- Agile teams are, ideally, highly motivated by practising self-management and self-organization. The organization gives the Agile team a high level of trust.
  - ✓ a team is "a small number of people with complementary skills who are committed to a common purpose, performance goals and approach for which they hold themselves mutually accountable" ~Jon Katzenbach and Douglas Smith
- empowered teams are:
  - ✓ **self-organizing:** since the team has the best knowledge about the project and is in the best position to organize project works
  - ✓ **self-directing:** the team can make their own decisions, not to be directed from the management
  - ✓ empowered team is more productive and efficient than teams with top-down decision making
  - ✓ mutual accountability and collective project ownership promote empowerment so that the team work as one whole

- **Tabaka's model for high-performing team**
  - self-organization
  - empowered to make decision
  - belief in vision and success
  - committed team
  - trust each other
  - participatory decision making
  - consensus-driven
  - constructive disagreement
- **High Performing Team vs Low Performing Team**
  - maximize performance by
    - ✓ clear and realistic goals
    - ✓ building trust
    - ✓ open and honest communication – even in case of disputes or conflicts
    - ✓ taking ownership, empowered, self-organizing coaching and mentoring
    - ✓ choose teammates with complementary skills to perform all tasks sense of belonging (identity)
    - ✓ limiting each team to have 12 members or below, break down the team if needed
    - ✓ make decisions through consensus (participatory decision model)
    - ✓ full-time, dedicated members
  - low performing teams are:
    - ✓ absence of trust
    - ✓ fear of conflict
    - ✓ lack of commitment
    - ✓ avoidance of accountability
    - ✓ inattention to results
- **Team Participation**
  - The whole project team would discuss in detail about the requirements of customers through face-to-face communication:
    - ✓ **Brainstorming –** everyone can voice out their opinions freely **without immediate judgement**
    - ✓ **Innovation games –** games are used to engage the team members and customer, e.g. 20/20 Vision, the Apprentice, Buy a Feature, Product Box, Prune the Product Tree (reference: Innovation Games: Creating Breakthrough Products Through Collaborative Play by Luke Hohmann).
    - ✓ **Parking lot chart –** a piece of paper to put important but off-topic issues / queries for later investigation / discussion, e.g. in requirement gathering
- **Two-way Communication**
  - Agile project management emphases feedback as feedback can help reducing misunderstanding and risks and generate better ideas.

- Communication must always be two-way; all the parties are given the opportunity to voice out their concerns and points of views

- **Cross-functional Team**
  - a group of people with different functional expertise working together toward a common goal
  - often function as self-directed teams
  - members must be well versed in multi-tasking as they are simultaneously responsible for various functions
- **Agile Coaching and Mentoring**
  - Coaching and mentoring are needed to help steer the team in the right direction:
  - coaching– help achieving (personal / organization) goals mentoring – pass on skills, knowledge, and experience
- **Agile Leadership Style: Servant Leadership**
  - traditional leadership and management emphasize on command-and-control (i.e. Theory X – workers are lazy and need to be monitored closely) servant leaders will lead by serving to ensure the needs of team members are met and roadblocks are cleared (Theory Y – team members are self-motivated)
  - an Agile servant leader needs to:
    - ✓ protect the team from interference, distractions, and interruptions
    - ✓ remove impediments to the team's performance
    - ✓ communicate and re-communicate project vision – maintain a common vision to drive the team to perform
    - ✓ carry food and water – i.e. provide all the resources for the team to perform, including motivate the team, provide trainings
- **Important tools/processes/concepts to enhance team communication:**
  - **information radiator -** a communication tool to physically displays key information about the current project status to the Agile team/stakeholders in the work area in the most visible and efficient manner, e.g. Kanban boards
  - **team space -** prefer all team members to be collocated in the same room facing each other for pro-active support, free discussion, open collaboration, tacit knowledge sharing and osmotic communication. If physical co-location is impossible, can make use of virtual co-location tools (e.g. instant messaging, video conferencing, etc.)
  - **Agile tooling**
    - ✓ these are tools to promote more effective communication (e.g. reduce roadblocks for collecting, maintaining, and disseminating information)
    - ✓ types: **low-tech high-touch tools**, digital tools
      - ➢ low-tech high-touch tools are preferred because these tools can promote collaboration and communication, and everyone knows how to participate
      - ➢ Examples:
        - ❖ co-located teams: war room and/or a dedicated conference room (walls filled with information radiators like whiteboards, billboards, post-its, charts, task boards, etc.)
        - ❖ distributed teams: virtual shared space using digital tools (wikis website, instant messaging, online planning poker, card meeting, version control, CASE tools, other Agile tools for building/configuring/deploying deliverables)
  - **osmotic communications** for co-located and/or distributed teams
    - ✓ team members in a co-located space can overhear conversations/discussion of other members
    - ✓ the team members will be able to extract useful parts from the conversations or to join in if necessary
  - **daily stand-ups**
    - ✓ daily stand-ups are a **time-boxed** (~ 15 minutes) and focused meeting to be held at the same time and in the same place for all team members to do a quick update on the project
    - ✓ stakeholders may attend but are not allowed to talk during the meeting
    - ✓ Each member answers the following 3 questions:
      - ➢ What have you done since last meeting?

        - What are you planning to do by next meeting?
        - What impediments (obstacles) are impacting your work progress?
- **Motivational Theories**
    - **Maslow's Hierarchy of Needs –** five levels of personal needs, from the fundamental at level 1 to the ultimate need at level 5
        1. Physiological
        2. Security
        3. Social
        4. Esteem
        5. Self-Actualization
    - **Herzberg's Hygiene Theory**
        - ✓ **satisfaction (motivators):** such as recognition, achievement, or personal growth
            - these are key factors to make team members motivated
            - salary is not an effective motivator
        - ✓ **dissatisfaction (demotivators):** such as bad working conditions, unfairness, etc.
            - hygiene factors are factors that must be present to avoid dissatisfaction but do not provide satisfaction, also called KITA factors
                - e.g. Company policies, supervision, relationship with supervisor and peers, work conditions, salary, status, job security
    - **Expectancy Theory**
        - ✓ an individual will decide to behave or act in a certain way because they are motivated to select a specific behaviour over other behaviours due to what they expect the result of that selected behaviour will be
            - for a person to be motivated, efforts/performance/outcome must be matched – will only work hard for achievable goals
        - ✓ key elements of Expectancy Theory
            - Expectancy (extra work will be rewarded)
            - Instrumentality (good results will be rewarded)
            - Valence (the individual's expected reward)
- **Productivity**
    - both velocity and throughout can be used to measure the productivity of a team
    - **Team Velocity**
        - ✓ Velocity is a capacity planning tool used in Agile project
            - usually defined as the number of story points that are completed in an iteration
        - ✓ Velocity usually increases gradually over the first few iterations as the team becomes more "performing" but stabilises afterwards as the product becomes more complicated (more bugs, more documentations, more dependencies, etc.)
    - **Cycle Time and Throughput**
        - ✓ Cycle time is the time necessary to get a single item of work done from start (idea) to finish (as a shippable product that delivers value)
            - Cycle time can be reduced by shortening iteration time (breaking down task sizes), limiting Work-In-Progress and reducing wastes
        - ✓ Throughput is the number of things that can be done in an iteration
        - ✓ **Cycle Time = WIP / Throughput**
        - ✓ **Defect cycle time** is the time between defect identification and defect remediation, the shorter the defect cycle time the better
- **Emotional Intelligence** – people with higher emotional intelligence (E.Q.) can relate to the feeling of people so that they deal with people issues more effectively
    - According to Higgs & Dulewicz, emotional intelligence includes seven components
        - ✓ Self-awareness
        - ✓ Emotional resilience
        - ✓ Motivation
        - ✓ Interpersonal sensitivity
        - ✓ Influence

- ✓ Intuitiveness
- ✓ Conscientiousness

- **Negotiation**
  - collaboration over contract negotiation
  - communicate with two or more parties to reach an agreement and resolve conflicts
  - Negotiation strategies
    - ✓ **Distributive negotiation:** adopt extreme positions initially and work to reach a deal through tactics (the assumption is value is limited; everyone needs to fight for the best value they can get)
    - ✓ **Integrative negotiation:** work together collaboratively to achieve greater successes by creating more values for a **win-win solution** (value can be created)
- **Conflict Resolution**
  - conflict is inevitable and is good for project success when controlled focusing on turning conflicts into a win-win situation, often need to make use of emotional intelligence and active listening conflict resolution tactics:
    - ✓ **Accommodation –** identify points of agreements and play down disagreement
    - ✓ **Avoidance –** ignore the conflict
    - ✓ **Compromise –** both sides to give up something, a lose-lose situation
    - ✓ **Forcing –** force one side to accept something, a win-lose situation
    - ✓ **Confronting –** open dialogue leading to problem resolution, a win-win situation
    - ✓ **Collaboration –** work together for mutually consented solution

Why People Over Processes?

One major reason is that processes and tools are simply easier to describe, explain, and classify than the trickier topics related to individuals and interactions. Also, it's more difficult to formalize practices related to individuals and interactions because people vary so much—in their skillsets, attitudes, experiences, perspectives, culture, and so on. So, the right thing to do with one team might be exactly the wrong approach to take with another.

This is why there is a common saying: "**The soft stuff is the hard stuff**". We need to recognize that the "**soft stuff**" is not just more difficult (harder) than processes and tools—it's also more important. We should focus on the people side of projects, even if that focus isn't our area of expertise or where we are most comfortable. Good people who have few or no processes in place can succeed even on difficult projects, yet poorly skilled or poorly aligned teams often fail, even with the best processes. As leaders of teams, we need to focus our efforts on people factors to get the maximum return on performance. This is why it's so important to understand how to build and support healthy teams.

Agile Team Roles:

- **Development Team/Delivery Team**
  - Build the product increments, using agile practices and processes.
  - Regularly update information radiators to share their progress with stakeholders.
  - Self-organize and self-direct their working process within an iteration.
  - Share their progress with each other in daily stand-up meetings.
  - Write acceptance tests for the product increments.
  - Test and revise the product increments until the pass the acceptance tests.
  - Demonstrate the completed product increment to the customer in the iteration review meeting.
  - Hold iteration retrospectives to reflect on their process and continually improve it.
  - Perform release and iteration planning, including estimating the stories and tasks.
- **Product Owner/Customer/Proxy Customer/Value Management Team/Business Representative**
  - Maximizes the value of the product by choosing and prioritizing the product features.
  - Manages the product backlog, making sure that it is accurate, up to date, and prioritized by business value.
  - Makes sure the team has a shared understanding of the backlog items and the value they are supposed to deliver.
  - Provides the acceptance criteria that the delivery team will use to prepare acceptance tests.

- Determines whether each completed product increment is working as intended, and either accepts it or requests changes (in the iteration review meeting).
- May change the product features and their priority at any time.
- Facilitates the engagement of external project stakeholders and manages their expectations.
- Provides the due dates for the project and/or its releases.
- Attends planning meetings, reviews, and retrospectives. (If this role is performed by a group of people, typically only one or two of them will attend these meetings.)

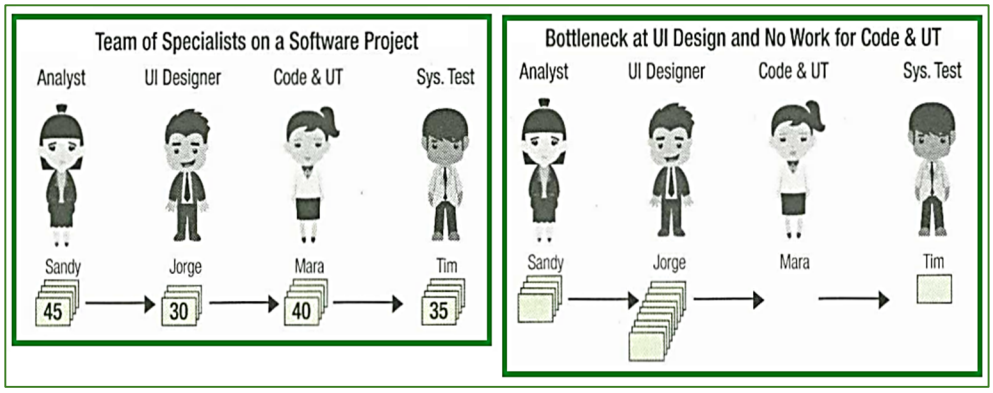
- **ScrumMaster/Coach/Team Leader**
  - Acts as a servant leader to the delivery team, helping them improve and removing barriers to their progress.
  - Helps the delivery team self-govern and self-organize, instead of governing and organizing them.
  - Serves as a facilitator and conduit for communication within the delivery team and with other stakeholders.
  - Makes sure the delivery team's plan is visible and its progress is radiated to stakeholders.
  - Acts as a coach and mentor to the delivery team.
  - Guides the team's agile process and makes sure their agile practices are being used properly.
  - Helps the product owner manage the product backlog.
  - Helps the product owner communicate the project vision, goals, and backlog items to the delivery team.
  - Facilitates meetings (planning, reviews, and retrospectives).
  - Follows up on issues raised in stand-up meetings to remove impediments so that the team can stay on track.
- **Project Sponsor**
  - Serves as the project's main advocate within the organization.
  - Provides direction to the product owner role (the person or team representing the business) about the organization's overall goals for the project.
  - Focuses on the big picture of whether the project will deliver the expected value on time and on budget.
  - Is invited to the iteration review meetings to see the product increments as they are completed but might not attend.

Building Agile Teams:

We can define a team as "**a small number of people with complementary skills who are committed to a common purpose, performance goals and approach for which they hold themselves mutually accountable**". There are several valuable aspects of this definition that are worth highlighting.
Agile methods recommend keeping the delivery team small (**typically 12 or fewer members**) since this allows the team members to develop better relationships and communicate more directly.

Benefits of Generalizing Specialists:

By definition, specialists have a narrow skillset, since they specialize in one role or function; so, if a project relies on specialists, there are two kinds of problems that can arise. First, there will need to be multiple handoffs between people who have different skills to get the work done—and handoffs of knowledge work are slow and risky. Since we are rarely building the same product twice, each handoff involves a lot of unique knowledge and information. A large portion of the work usually consists of theory, ideas, or models, so there are few tangibles, visible assets to hand over. Instead, one specialist has to try and explain the ideas, goals, and design constraints they used to another specialist—and generally something gets lost in the transition



Characteristics of High-Performing Teams:

- **Create a shared vision for the team.** This enables the team to make faster decisions and builds trust.
- **Set realistic goals.** We should set people up to succeed, not to fail, so goals need to be achievable.
- **Limit team size to 12 or fewer members.** Small teams are able to communicate face-to-face and support tacit (unwritten) knowledge. If an agile project requires more resources, the preferred approach to split the work among smaller sub teams. Representatives of the sub teams the need to come together every day to coordinate and synchronize work across the project (for example, in a scrum of scrums).
- **Build a sense of team identity.** A sense of team identity helps increase the team members' loyalty to the team and support for other team members.
- **Provide strong leadership.** Leaders should point out the way, then let the team own the mission.

Lyssa Adkins has also explored high-performance teams and has identified that they have the following Eight characteristics:

- They are self-organizing, rather than role- or title-based.
- They are empowered to make decisions.
- They truly believe that as a team they can solve any problem.
- They are committed to team success rather than success at any cost.
- The team owns its decisions and commitments.
- The members are motivated by trust, instead of fear or anger.
- They are consensus-driven, with full divergence and then convergence.
- They are in constant constructive disagreement.

Encourage Constructive Disagreement:

High-performing teams speak to establish a safe environment in which debating or arguing over issues is seen as healthy. Constructive conflict is encouraged because it ultimately leads to better decisions and stronger buy-in for those decisions once they are made. Divergence (argument and debate) and convergence (agreement about the best solution) increase the team's commitment.

The Five Dysfunctions of a Team:

- **Absence of trust:** Team members are unwilling to be vulnerable within the group.
- **Fear of conflict:** The team seeks artificial harmony over constructive, passionate debate.
- **Lack of commitment:** Team members don't commit to group derisions or simply feign agreement with them.
- **Avoidance of accountability:** Team members duck the responsibility of calling peers on counterproductive behaviour or low standards.
- **Inattention to results:** Team members prioritize their individual needs, such as personal success, status, or ego,before team success.

Models of Team Development:

Shu-Ha-Ri Model of Skill Mastery:

It describes a three-step process of increasing mastery that progresses as follows:

1. **Shu:** Obeying the rules— **shu** means "to keep, protect, or maintain"
2. **Ha:** Consciously moving away from the rules— **ha** means "to detach or break free"
3. **Ri:** Unconsciously finding an individual path— **ri** means "to go beyond or transcend"

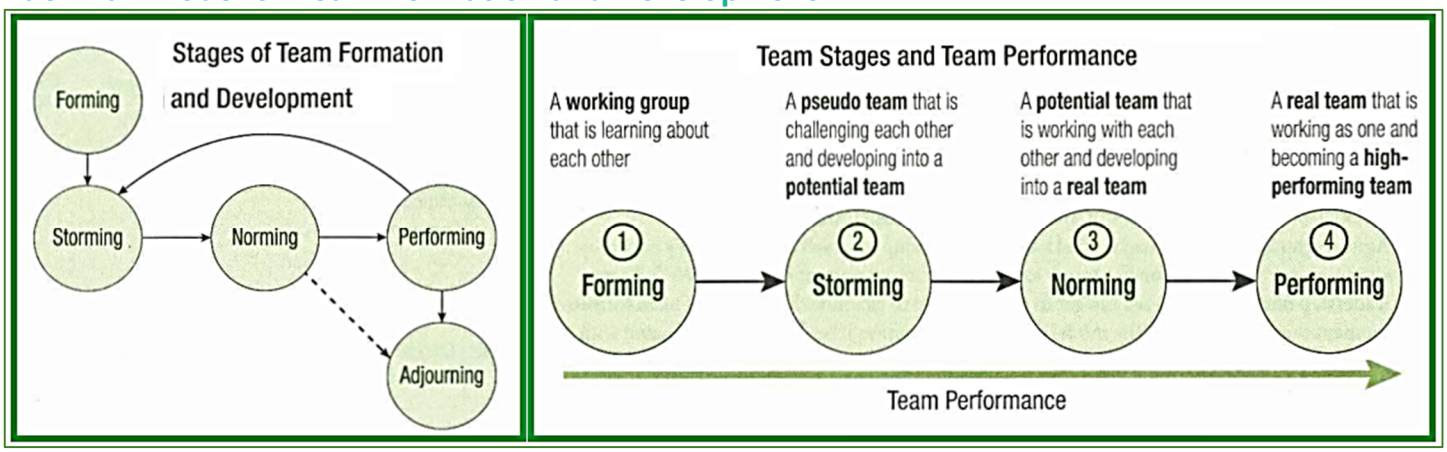
Dreyfus Model of Adult Skill Acquisition:

The five stages of this model -

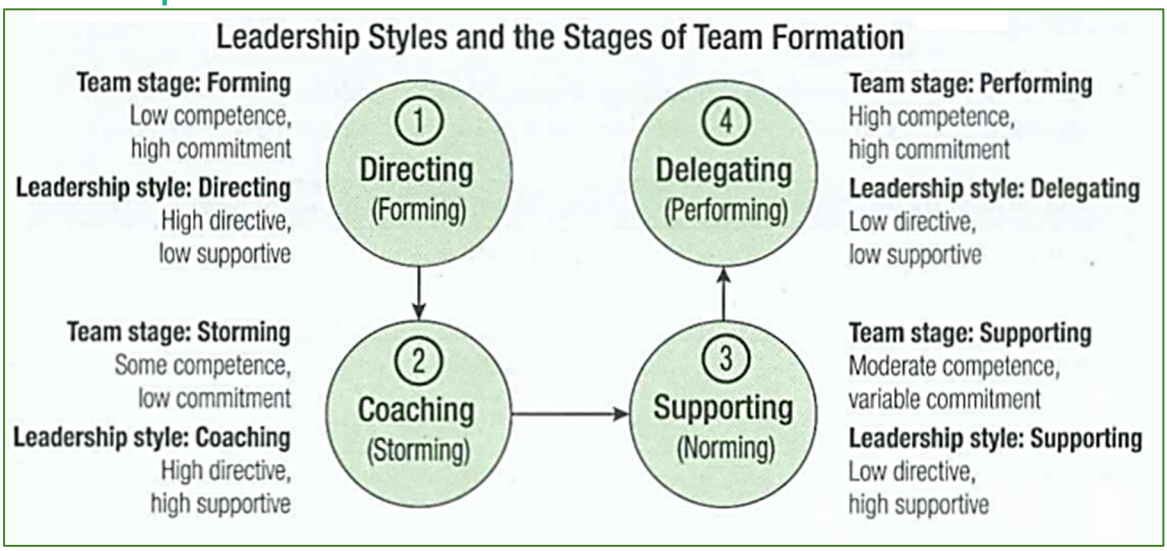
1. **Novice:** Novices follow the rules they have been given and make analytical decisions. For example, if we are learning to drive a manual transmission, we might be told to change gears as the engine approaches 2,500rpm. Although this rule is a good start, it could lead to problems if we try to drive on a hill, which will influence when we should switch gears.

2. **Advanced beginner:** At this stage, we are still following rules and making analytical decisions, but now we have gained enough experience with real-world situations to begin to understand the context of the rules. To decide when to change gears, we might use a guideline such as "gear up when you hear the engine racing."
3. **Competent:** As we gain competence, the number of rules and guidelines for different contexts becomes overwhelming. Since we can't apply them all, we begin to decide which rules are the be for each situation, and this makes us feel more personally responsible for the choices we are making.
4. **Proficient:** At this level, our decision making is still analytical, but we are actively choosing the best strategy rather than relying on the rules. In the process, we become more emotionally involved in the task. For example, as a proficient driver, we will have a gut feeling if we are approaching a corner too fast on a rainy day.
5. **Expert:** As we develop expertise, our decision making becomes intuitive—we are able to spontaneously assess the alternatives and select the best approach without having to first examine all the possible strategies analytically.

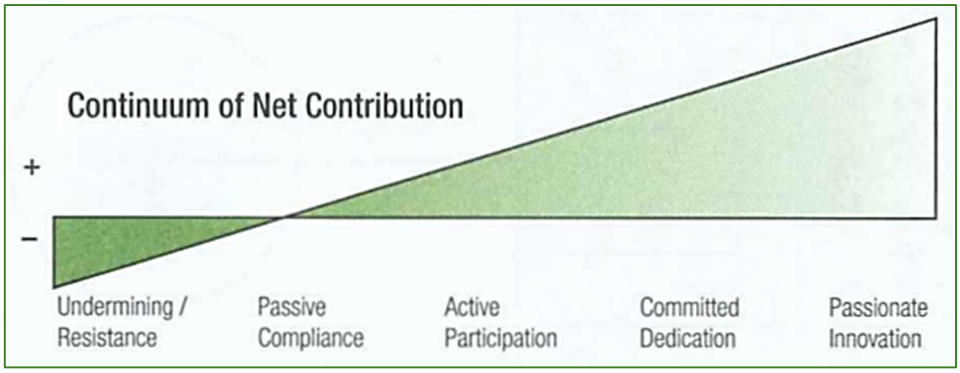
Tuckman Model of Team Formation and Development:

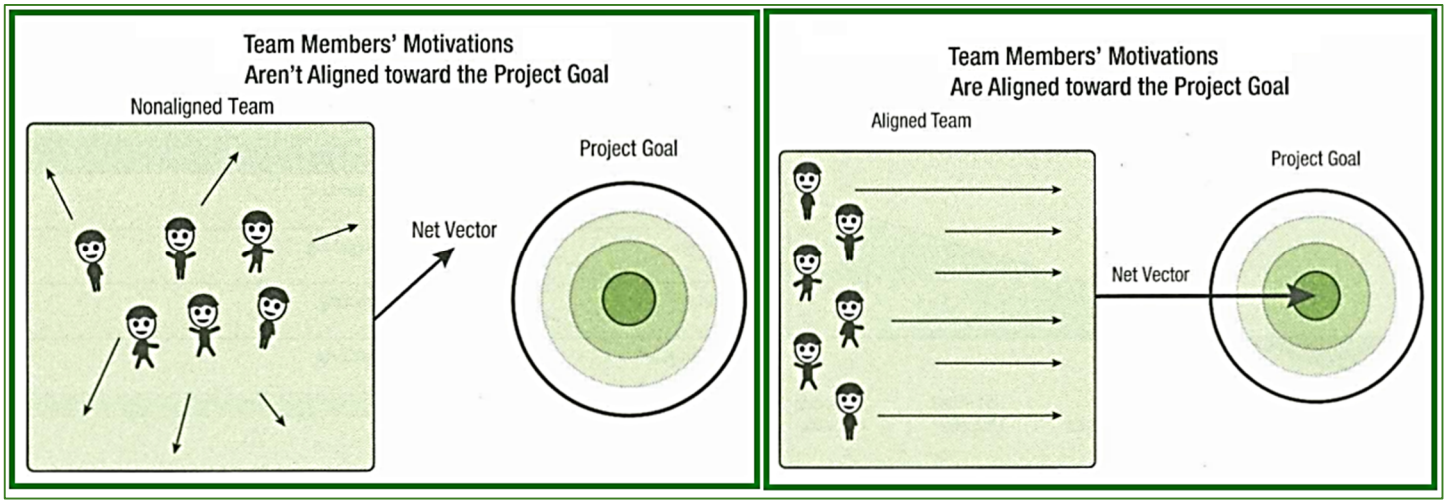


Adaptive Leadership:



Team Motivation:

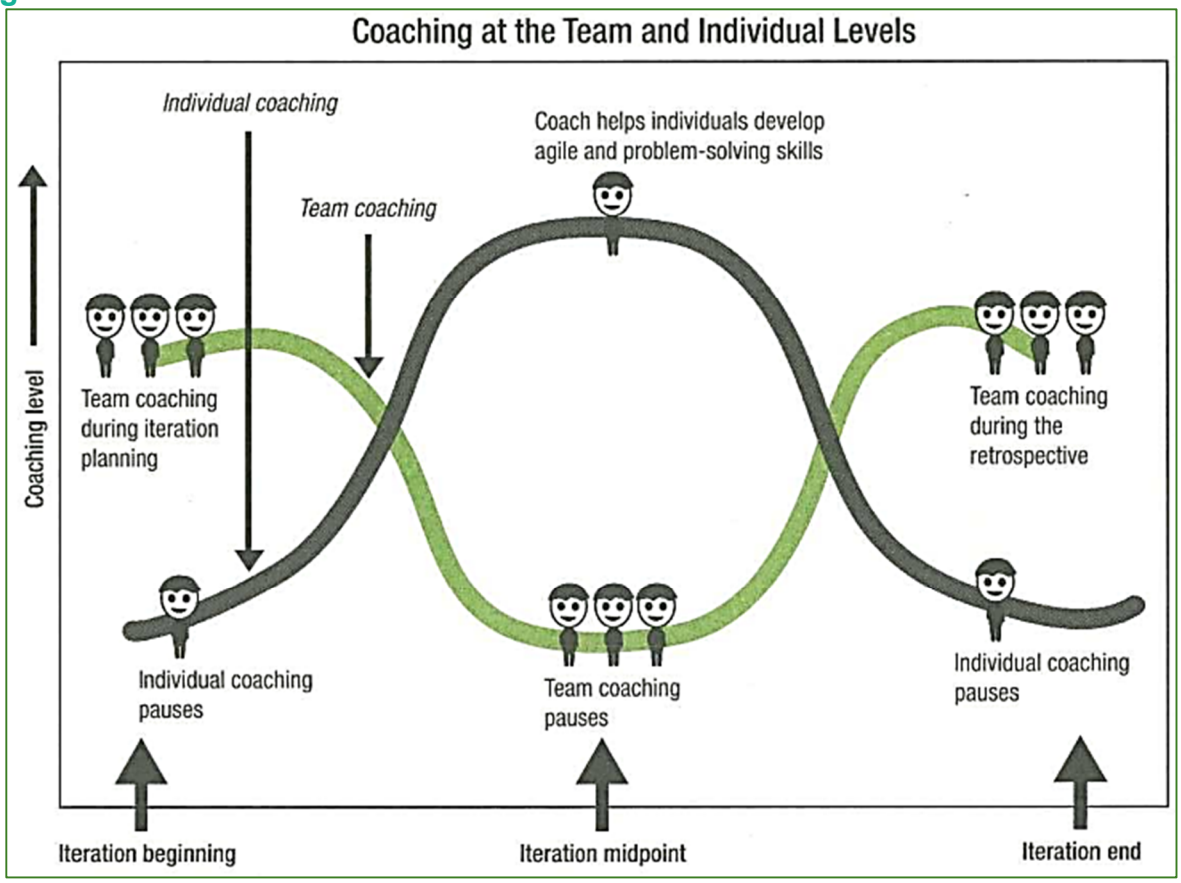




Training, Coaching, and Mentoring:

- **Training** is the teaching of a skill or knowledge through practice and instruction. The agenda is usually created by the trainer, and the format is very structured (and typically prepared in advance). An example of training would be attending a one-day course on agile planning techniques.
- **Coaching** is a facilitated process that helps the person being coached to develop and improve their performance. Coaching often starts with an initial topic that is then expanded and developed in collaboration with the coach. Coaching sessions are usually scheduled in advance and have a defined structure. An example of coaching would be arranging a session with the ScrumMaster to get help with project reporting.
- **Mentoring** is more of a professional relationship thana specific activity. The mentor can be a sounding board for tackling issues on an as-needed basis. The mentee owns the agenda, and the format is free flowing. An example of mentoring would be having a relationship with an experienced agile leader who you meet with from time to time, bounce ideas off, and ask for advice.

Coaching:



Guidelines for one-on-one coaching:

- **Meet them a half-step ahead.** Don't try to push people directly to the endpoint. Instead, coach them so that they move toward the end goal and take the next step from where they are now. For example, let's say our aim is to have the team members select their own work tasks, but they aren't there yet Instead of just telling them bluntly that agile teams self-select their work, try to get them halfway there by asking them questions like, "Who can do this one?"
- **Guarantee safety.** Tell the team members that all coaching conversations will be kept confidential, and then make sure they are. People will be more willing to contribute and share if they aren't afraid that their concerns will be repeated out of context.
- **Partner with managers.** Often the team members' functional managers are not on the project team. These managers might not be using, or even aware of, agile methods. Because of this, the way the team members are measured in their functional roles might not match agile values. We need to partner with functional managers to align everyone's goals and ensure that the team members' project contributions are reported appropriately to their functional managers.
- **Create positive regard.** We might not personally like every individual we coach, but we do have to help them equally. If we dislike someone, this sentiment can show through, often in subtle ways. Therefore, it's important to cultivate a genuine compassion for others and a desire to help everyone improve in their roles.

Creating Collaborative Team Spaces:

Since agile methods recommend face-to-face interaction as the preferred means of communication, it is no surprise that they also recommend that teams be co-located—in other words, that all the team members work together in the same location. It isn't enough to simply locate all the team members in the same city, or even on the same floor of the same building. An agile team is only considered to be co-located if all its members are working within **33 feet—10 meters**—of each other, without any physical barriers such as walls or doorways between them.

Team Space:

To take full advantage of the benefits of co-location, the team space should also be supplied with the following tools and equipment:

- Whiteboards and tackboards
- Sticky notes, sticky paper, flip charts
- Round table with screen/laptop
- Video conferencing capability
- No barriers to face-to-face communication
- Food, snacks, and toys!

Caves and Common:

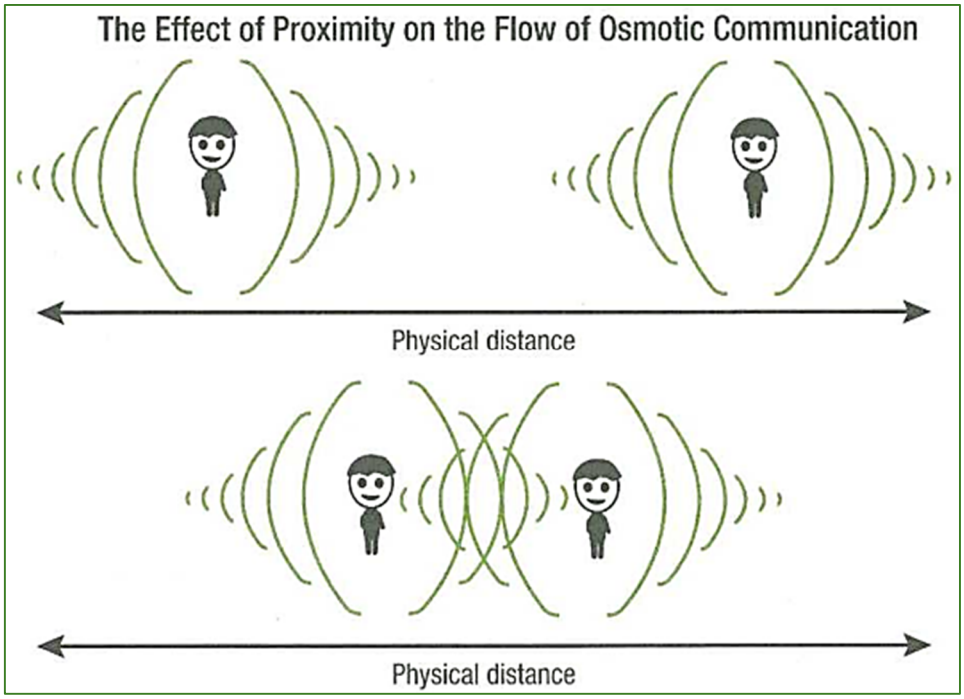
One of the challenges of working in an open area without barriers is the lack of privacy and quiet time. To address this, agile team spaces follow a "caves and common" model. Most of the work is done in the large "common" area, where the team members work together as a group. However, they also have access to private spaces, called "caves", where they can go to make private calls, have one-on-one conversations, or work on their own for short periods of time.

Tacit Knowledge:

"**Tacit knowledge**" is the unwritten information that is collectively known by the group, such as how to restart the printer when it jams. Tacit knowledge works best for small, co-located teams—in fact, the emphasis on facilitating and maintaining tacit knowledge is one of the reasons why agile methods recommend limiting teams to 12 or fewer people.

Osmotic Communication:

Osmotic communication refers to the useful information that flows between team members who are working in close proximity to each other as they overhear each other's conversations. To improve their osmotic communication, we want to get people sitting and working closely together with few barriers between them.



Global, Cultural, and Team Diversity:

- Different time zones
- Different cultures
- Different communication styles
- Different native languages

Distributed Teams:

Distributed teams are teams that have at least one team member working off-site. Agile methods work better for distributed teams than non-agile approaches because of the following factors:

- The short iterations used in agile development force continuous close collaboration and coordination.
- The project will be easier to control because a releasable product is built each iteration.

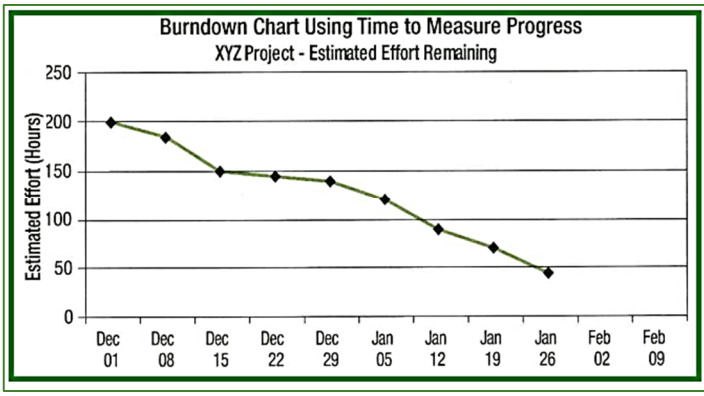
Digital Tools for Distributed Teams:

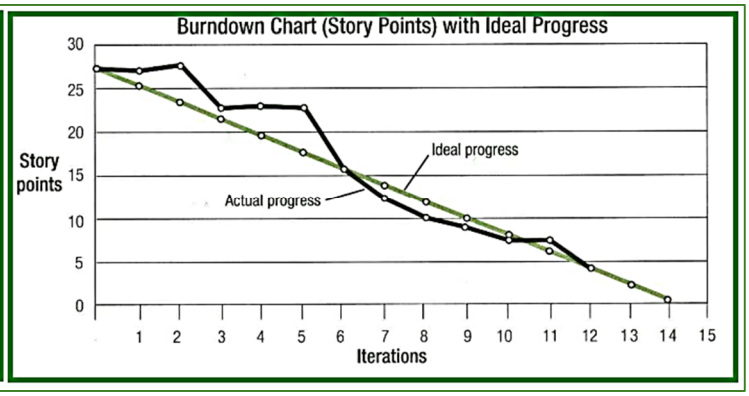
- Video conferencing, live chat, Skype
- Interactive whiteboards
- Instant messaging(IM) and VoIP (Voice over Internet Protocol) headsets
- Presence-based applications
- Electronic task boards and storyboards
- Survey applications
- Agile project management software
- Virtual card walls
- Smartboards
- Wiki sites, document management tools, and collaboration websites
- Automated testing tools, automated build tools, and traffic-light-type signals
- CASE tools

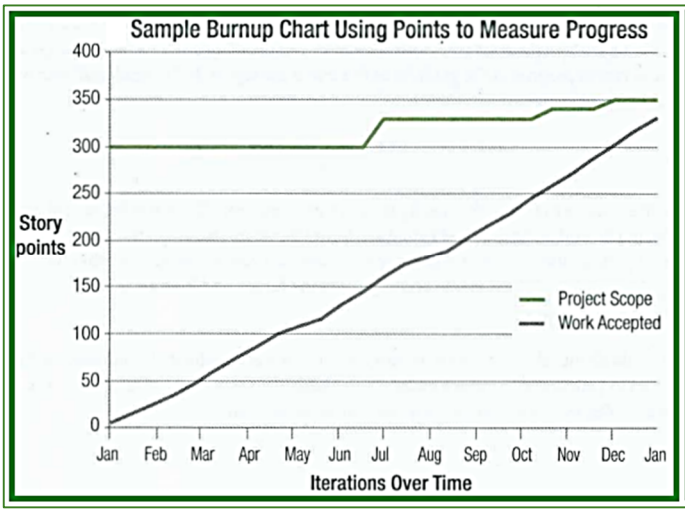
Tracking Team Performance:

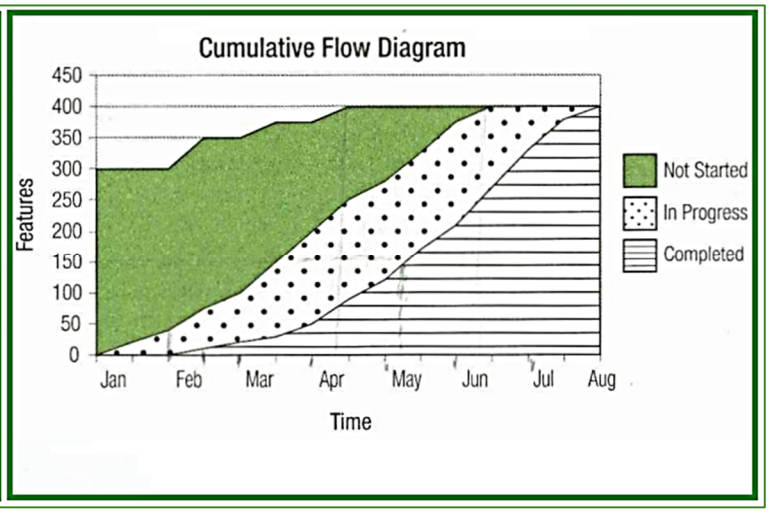
Burn Charts:

Bum charts are important because they make the team's progress visible at a glance and make it easy to forecast when the project (or a release within the project) will be complete. Slowdown of progress as depicted by these charts could have had several causes: people taking vacations over the holidays, the customer adding more scope, or the team discovering more tasks that needed to be done. We cannot discern the reason for the slow progress by looking at this chart.









Velocity:

Velocity is defined as the "**measure of a team's capacity for work per iteration**". This provides a way to track and communicate what they have accomplished, anticipate what they will be able to accomplish in the future, and forecast when the project (or release) is likely to be done.

While it may seem logical to predict ever-increasing velocity as the team gains experience, velocity does typical plateau. One reason for this is that, as the product gets bigger, there is more to maintain, refactor, and possibly support if early versions of the product have been deployed. In general, knowledge work projects tend to increase in complexity as the work is being done.