Working In Contemporary Teams

Dr Chitra Dey

Department of Management Studies

Anna University

Week – 04

Lecture - 15

Introduction to Agile teams

Welcome to lecture 1 of module number 4. In module number 3, we had seen an introduction to input process output model and we also examined two different kinds of contemporary teams. The human AI team and the teams where members have membership in multiple teams. These two teams were clubbed together primarily because of their similarities in team composition. That is, it is their team composition which made them so unique. In this module, we will be looking at two different kinds of contemporary teams.

Agile teams and remote work teams. These two teams have been clubbed together because they have arisen primarily in response to the demands of work. So, let us first go to agile teams. In this lecture, we will be giving an introduction to what is agile methodology.

Then we will look at what are the definitions for agile teams, some of their common applications. Then we will see what are the scenarios where we should not use agile teams and how we can design a good agility. The agile is a methodology and it arose due to requirements in software development. However, agile now is not solely used in the domain of software development, but has applications across many kinds of industries and in many processes where we require innovation and creativity. To help understand where and how agile can be applied, let us look at a development process.

Think about any product that you use now, a laptop or a mobile. Think that you are holding your mobile not as it is in its current form, but in the form that it was in 10 years back. Say that the company which is manufacturing your mobile decided 10 years ago that it had created the ultimate mobile and stopped making any kinds of improvements in it. Then what kind of a mobile would you be holding? Well, the mobile that you would be holding would not be catering to many of your requirements and would be quite

cumbersome to use. So what organizations understood was that they have to create a development process which is capable of quickly understanding customer requirements and converting it into product features and launching new products, not necessarily new products, but better versions of the old product.

So they realized that the old development process was cumbersome because it focused a lot on requirement gathering and then it spent a lot of time in designing and testing based on the requirements that was gathered in the first place and finally they came up with a finished product. The problem here is that customer requirements keep changing. The requirements which the customer had said in the first place may not be existing when the final products come out and therefore the old development process does not meet customer requirements and the developer usually spends a lot of time and effort in building a product which does not have any value when it finally hits the market. So what changed? Well, organizations realized that requirements change and that we cannot have one round of design and testing but we need to have iterations of design and testing to improve the product and that the focus should not be on creating a finished product but rather on meeting customer requirements. It was to meet these primary needs that the agile methodology was developed.

So what is agile? Agile was first penned in the Agile Manifesto which was crafted in 2001 and was developed as a guide for software development teams. The people who came together to develop the Agile Manifesto were a team of software developers and they were primarily unhappy with the old software development process which focused on gathering customer requirements through extensive documentation, design and testing and then coming up with a finished product. They said that there should be some way to incorporate customer requirements as the product was being developed and that the focus should not be on creating one final product but on meeting the customer requirements as and when it developed. So they said maybe we should set down different values so that the software development team can focus on those values and they consisted of four primary values. One, giving importance to the individuals and interactions in the teams over the tools and processes used to develop the software development, the software.

Focusing on creating a working software rather than focusing on comprehensive documentation. Focusing on collaborating with the customer and understanding his or her requirements rather than negotiating on the contract and finally responding to change rather than following a plan. Apart from the four values, they also set down 12 principles of the Agile methodology. At this point, I must say that what they had proposed was not actually a step-by-step process but rather a mindset to developing software. And there were many other methodologies or frameworks which had already been in use which

followed the process of iterative and incremental product development.

This included the lean manufacturing system which was followed in the Toyota production system and Scrum which was evolved as early as 1996 which was used in software development teams. So after 2001, once the Agile manifesto was created and set out, it began to involve all kinds of development frameworks which focused on creating products in an iterative and incremental manner. So having seen what is Agile, let us see what are Agile teams. An Agile team is a small number of people with complementary skills who are committed to a common purpose, set of performance goals and approach for which they hold themselves mutually accountable. Let us look at some of the key words here.

Well, first of all, the size. Agile teams are no bigger than 3 to 7 in size. In fact, the maximum prescribed size for an Agile team is not more than 9. The members in the Agile team have complementary skill sets. That is, they are not only good at doing their own work, but they are also highly skilled at getting along well with other team members.

They are also committed to a common purpose which is to deliver a good quality product to the customer and focusing on the customer's requirements. And to achieve this goal, they hold each of themselves mutually accountable. So let us see what are the applications of Agile teams. While the Agile was primarily prescribed for software development and some other methodologies which are very similar to Agile were used in manufacturing systems, today we find that Agile has a lot of application across different industries. It finds an application in construction, fashion, biotechnology and even music streaming services.

Fashion and biotechnology are two areas where we constantly need to have teams evolve a product. For example, you would have noticed that when COVID struck, all the pharmaceutical companies were under a lot of pressure to develop a product or a vaccine that could help people and they could not go ahead with producing one finished product or vaccine. They had to iterate the product that they had developed by looking at it. So Agile was developed primarily as a response to software development teams. However, we saw previously that many other methodologies which prescribed to the values of Agile were already being used in manufacturing.

Today, the concept of Agile finds application across various industries and across many processes that require innovation. Some of the industries where it requires application are construction, fashion, biotechnology, music streaming services and so on. Fashion is one industry where Agile methodology finds a lot of application. You would notice that

many fashion companies have to quickly understand the changing needs of the customer and change a product quickly and make it to market. Here, we see a very good application of the Agile methodology.

But there is a question which comes up in the mind. Is Agile the secret key to all the problems in all kinds of teams? Is there a possibility that Agile suits certain kinds of industries? Definitely. Agile is most applicable when the task and the work environment in which the team is working is dynamic and is very important for the team members to collaborate extensively with external stakeholders and when there is a presence of multiple steps to complete the task. The task, the employees and the customers also should be admirable to agility. If the customers have a feeling that you should be using extensive documentation and you should be showing proof of all the steps that you have taken in completing the task, then it is going to be very difficult to convince them that Agile is the way to go.

The employees in the company should also be trained to be able to follow the Agile methodology. When we look at team design, we will see what are the characteristics of the employees which is essential for Agile methodology to take off properly in the organisation. Finally, the task itself. As I said, the task should require a lot of collaboration with external stakeholders. That is, the employees who are engaged in the process should need to communicate with people like vendors, customers, other teams to constantly get feedback to improve the product on which they are working.

If this is not required and if the work which they are doing is stable and requires little input from external stakeholders, following an Agile methodology may not help. In fact, it will just hamper the working of the team. So, having seen the application of Agile teams, let us look at some of the benefits of using Agile. 2. Agile definitely ensures better collaboration between diverse team members.

Because of its focus on constant meeting between team members and a lot of coordination between the team members, it ensures that the team members eventually understand the requirements of each other and better collaborate. Agile methodology also results in highly adaptable and flexible teams. That is, their focus is not on documentation and sticking to plans. Their focus is on constantly collaborating with different people, gathering their changing requirements and incorporating it into the design of the product. This ultimately reduces error and cost.

Just imagine, if the customer requirements had changed over the period in which the team was completely focused on developing and designing the product, by the time the product came to the market, it would not meet the changed customer requirements. So,

one of the final benefits of Agile teams is that it produces highly satisfied customers. So, let us delve into better understanding Agile teams by looking at the IPO framework. What are we going to focus on so that we better understand team functioning in Agile? Well, we are going to look at how teams are designed. Then we are going to look at some important processes for Agile teams.

And finally, we will look at some outcomes. In terms of processes, we will be focusing on communication, collaboration and boundary spanning. And we will look at team cooperation as an important emergent state. And finally, in outputs, we will be looking at performance and team viability. It is very important to understand that having changed the way, expecting the team to work in a particular way, that is, that is following the Agile methodology is not possible without thoughtfully structuring and composing the team so that it is able to meet the requirements of team processes.

So, let us see how to design a good or effective Agile team. The first aspect that we have to look at is the composition of the team. Like I said before, Agile teams are small in size. The prescribed team size for Agile teams is between 4 to 7. And in fact, many texts say that we should not have more than 9 members in an Agile team.

This is because there is going to be a lot of communication and coordination which is required between the team members. And having large teams drastically increases transaction costs and makes it very difficult for all the team members to be in tune with each other. Considering the requirements of Agile methodology, it is very important for the teams to usually be co-located. That is, usually Agile teams are based in the same location. What are the kinds of skill sets which are required for Agile team? Well, like I said before, the team members should have complementary set of skills.

That is, they will be responsible for handling a team task from the development to the release of the product. So, this is a self-contained and self-managed team which should be able to take care of every single step in the development of the product. And one very important aspect for team members to have is to have T-shaped skills. So, what is T-shaped skill? A T-shaped skill is when a team member is not only having the ability to work a T-shaped. The next question that comes into mind is what are the skill sets which are required of members of Agile teams? Like I said in the definition, members of Agile teams should have complementary skills.

That is, they should not only be good at doing their own work, but should also be able to get along with others. That is what is meant by T-shaped employees or T-shaped skills. Here, team members have a very deep understanding of the particular technology or area which they are working on. And they also have a broad understanding about how their

skill sets and their knowledge should be complementing the work which other team members in the team are doing. Such kind of team members are extremely rare to find and also very valuable.

That is, they not only have the technical skill sets which are required to have an in-depth understanding of their own field, but they also have social skills which enable them to collaborate with the other members on the team. Now, are they just social skills? No. This team member who has T-shaped skills has not only deep technical knowledge in his or her own area, but also a broad understanding of technical aspects of what other members are doing. Upon this, they have the social skill sets which are required for the team members to collaborate with each other. So, what kind of skill sets should members of agile teams have? Like I said in the definition, members of agile teams should have complementary skills.

That is, they should not only be good at doing their own work, but should be able to complement the work which other members in their team are doing. This kind of skills are called as T-shaped skills. That is, each team member not only has a deep understanding about the technology that he or she is working, but should also have a broad understanding or technical knowledge about the work of other team members. And upon this or along with this, they should also have skill sets which allow them to collaborate with others. So, a wholesome member of an agile team should not only have deep technical knowledge, but broad technical knowledge which along with social skills will allow him or her to collaborate with the other team members.

Such kind of team members are very rare to find and are highly valued by organizations which adopt and employ agile methodology. The second aspect of team design is to understand the roles. Now, we must understand that in traditional teams, there is usually the presence of a hierarchy. That is, there is going to be a team leader. This team leader has a broad and deep understanding of the work that has to be done and he or she will use his experience and expertise to break the work into small pieces and allocate it among the team members.

The task of the team members is usually to understand his or her work and then focus on that. They are primarily responsible for doing and delivering their work. Now, this kind of a situation is possible when there is low amount of complexity and dynamism in the task. However, when there is high level of dynamism and complexity in work, team leaders or one person cannot be trusted with the work of managing. The second aspect in team design is to look at the team structure.

Let us for a moment look at traditional teams. In traditional teams, you have a hierarchy

where there is a team leader and team members. The team leader usually has a good understanding about the work and based on his or her experience and expertise will break the work down into small pieces and assign them to different team members. The team members are primarily responsible for completing his or her piece of work. Now, this kind of a situation is quite okay when the work is not dynamic or complex. In situations like current software development where the customer requirements are constantly changing, it becomes very difficult for one person to have total control of the dynamism and complexity of the project.

There is also another underlying assumption in the team structure of traditional teams. It is the fact that there is very little change that is going to happen and that requirements of the team task are completely understood at the beginning of the team cycle itself. However, when you come to agile teams, there are two, three things. First, the team does not have all the information that is required to complete and finish the product at the beginning of the team's life cycle. They are probably going into the task without having a complete understanding and this lack of complete understanding is accepted by all the team members.

So, the team's understanding of the task evolves as the team members work on the task. So, to be able to manage such kind of a situation, we have to structure the team such that team members have, are clearly allotted different kinds of tasks. So, there are three primary roles in agile teams. The first is that of a team lead. The team lead is not a person who is hierarchically above the other team members.

He or she is a supportive leader whose primary duty is to help the other team members plan and manage their work. In certain frameworks like Scrum, this person is also called as a Scrum master. The primary role of the team lead is to empower the team members and make sure that there are no roadblocks in standing in the way of the team members. The next role is that of a product owner. The product owner is a person who has a very deep understanding of the customer's business and how the product that the team is building supports the business of the customer.

This person prioritizes the work for the team and it is very important for the team members and the team lead to trust the decisions of the product owner. However, this is not a tyrannical role, but in fact, the role of the product owner evolves by having constant dialogue with the team lead, the external stakeholders and the team members. Finally, the team members are not roles which are hierarchically below either the team lead or the product owner. In fact, at any given point in time, team members are also responsible for taking up leadership in case the situation demands them to do so. They are people who have the skill sets which are required for completing the task from the

beginning to the end.

Like I had said, agile teams are self-sustained and self-managed teams and therefore, the team members should have all the required skill sets, competencies and knowledge to complete the task from beginning to end. The final member of the team who may not be completely involved in the team's day-to-day functioning is the stakeholder. It is important to mention the stakeholder here because the stakeholder, because the team members depend upon the stakeholder to give them continuous input so that they can iteratively develop the product. Another important aspect of team design is the volatility or the boundedness of the team. Boundedness refers to the extent to which membership in the team is stable.

There are three aspects to it. The first aspect is whether team members work on multiple teams or are assigned only to one team. The second aspect is whether team members are working on a core area or whether they are working on peripheral areas. For example, some team members might be developers in which case they can be called as core members. Some members might be subject matter expert and therefore, they will be working on peripheral roles. So, if you look at team boundedness, the members of an agile team are not having.

The next important aspect that we have to keep in mind when we are designing agile teams is the extent of volatility or boundedness. There are three aspects to boundedness. The first one is whether members in the team have membership in multiple teams, whether they join the team and stay with the team throughout the duration of the team's life cycle. And the last one is whether they are working on a core area or a peripheral area. So, boundedness is a property of the team that exists on a continuum.

On one extreme end of this continuum, you will have teams which are highly bounded. That is, at any given point in time, members of the team are members only of one team. That is, they do not work on multiple teams simultaneously. Second, their membership in the team starts with the beginning of the team and often goes on till the team is disbanded. And finally, most of the team members in highly bounded teams work on core areas of the team task.

On the other end of the spectrum, we have teams which have very low boundedness. In such kind of teams, at any given point in time, team members might be working in multiple teams. They may come and go working in the team at different points in time. And finally, you will have members who work on core area and some other members who work on peripheral areas.

In Agile, usually the teams have very high boundedness. That is, the volatility of the team membership is very low. This is because of the requirements of the task that impinge upon the team members to dedicate their time and effort to the team task alone. Therefore, you will see that Agile teams have dedicated members and that most of the core members, that is the developers in the example of the software development team, are mostly working only in one team and do not leave the team till the end of the sprint. And finally, in Agile teams, you will see that the boundedness is very high.

That is, members will not have membership in multiple teams. They will usually be working in not more than one or two projects at any given point in time. Then the membership, that is, people who are working in core tasks of the team will not have membership in multiple teams while people who are working in peripheral roles like the subject matter experts might find themselves assigned to multiple teams at the same time. And finally, we will find that team members, given the requirements of Agile methodology, it is very difficult for an Agile team to function properly unless it is highly bounded. That is, Agile teams, you will find that team members do not have membership in multiple teams simultaneously. Especially, the team members who are working on the core tasks of the team will not be participating in more than two teams at any given point in time.

Also, these team members are dedicated. That is, they start with the lifecycle of the team and go on till the team is disbanded. Again, you might find that there are people in peripheral roles like the subject matter experts who come and give advice and help to the team as and when required. Their volatility will be high but the rest of the team is a dedicated, co-located team. The next important aspect of team design is the leadership. Two important concepts of leadership have gained a lot of traction in research and practice in Agile methodology.

They are shared leadership and servant leader facilitator. So, let us see what they are. In traditional teams, we usually find that there is a hierarchy with the team leader and team members. But as I mentioned before, when I was discussing team roles, Agile teams have no fixed hierarchy. Even though you have the designation as a team lead and a product owner, they are merely designations and are not in positions to dictate orders to team members. In fact, the team members and the product owner and the team lead work together and collaborate to ensure that they stay fixed on the final goal of the team which is to deliver a working product to the customer.

So, a shared goal and an understanding that that is the priority for the team is very important for all the team members to hold. In fact, if you look at the definition of shared leadership, it is not talking about a leader and followers, but it is talking about a dynamic

and interactive influence process that happens between members of the team. Now, one important benefit in having shared leadership is that it greatly reduces stress level for individual leaders. Considering the dynamism and complexity of the work in which the team is involved, it is better to have shared leadership to reduce stress levels for individual leaders. Again, I would reiterate the product owner and the team lead are merely designations.

It is their primary duty to be able on the part of the product owner to ensure that the team is meeting customer requirements and on the part of the team lead to make sure that there are no obstacles in the path of the team members. Next, we look at servant leadership theory. Servant leadership theory was proposed because of many accusations to traditional leadership theories. These accusations said that traditional leadership is primarily involved with the leader and often the leader is selfish in thinking about his or her primary goals. The servant leader idea was proposed to talk about a leader whose focus is on meeting the needs of the team members.

That is, it is not a burden which is thrust upon the leader, but in fact, it is the leader's right to take it upon himself or herself to serve and make sure that the needs of the different team members are met. In fact, if you look at the team lead, the team lead is primarily a person who is designated to have an eye on what is happening in the team, to understand points of friction and help team members stay on track in delivering the objectives of the agile team. If you look at the product owner, the product owner is having a role to make sure that they are looking at the requirements of the customer, that these requirements are incorporated into the product that is being developed and making sure that the change is as seamless as possible. So, it is from this perspective that the servant leadership theory has evolved. Ten characteristics have been identified for servant leaders and I am just mentioning two - listening and empathy are considered as very important traits for servant leaders to have.

To summarize, in this lecture, we have looked at what is agile methodology. We saw what are the situation in which agile methodology developed. We looked at what are agile teams and some of its common applications. Then we looked at how we can design agile teams, the composition of agile teams, the structure of agile teams. As usual, I will leave you with some interesting topics that you may want to look up.

I have just mentioned what is agile methodology. There are so many frameworks which fall under the agile methodology and one of the most interesting aspects of working with agile methodology is to identify one that suits your organization and your team. It might also be very beneficial if you look up what are Scrum, Kanban and the Toyota production systems. Looking forward to work in contemporary teams, it is very

important also to understand what are T-shaped workers and T-shaped skills. Finally, one very interesting leadership theory which I have introduced but not looked into in great depth is the servant leadership. With these interesting topics, I leave you. Thank you for joining me.