

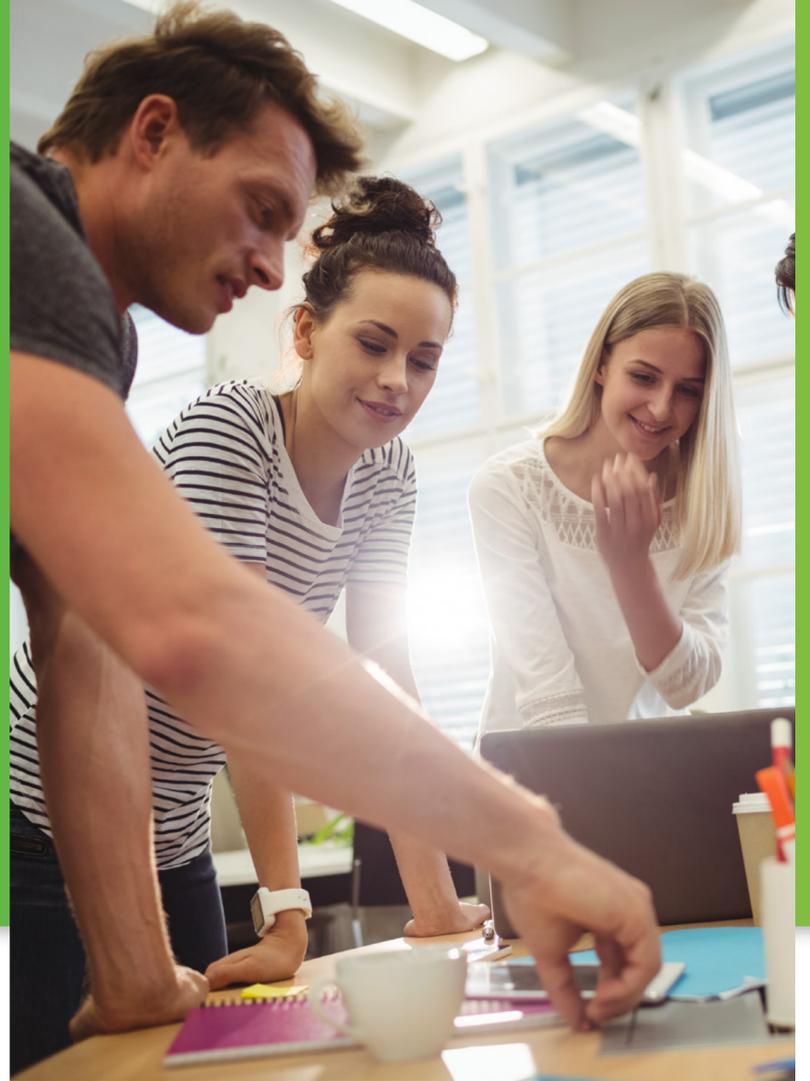
DISTRIBUTED SCRUM TEAMS



Manoj Khanna
Chief Methodologist at IT Labs

Distributed Scrum Teams

The use of distributed teams, where members of the development team do not share the product owner's location — or that of some other team members — is increasingly prevalent in small companies and large organizations. Invariably, it results from the use of offshore teams in lower-cost geographical regions. However, with this lower base cost comes an increased coordination cost. Amongst many things, it includes delayed communication and meetings due to differing time zones, lack of intra-team coordination, and a higher chance of an unsuccessful project. The choice of a distributed team model that an organization selects will have a significant impact on whether its distributed scrum efforts fail or succeed.



Development Models

Multiple development models can be used in agile software development, and these models can be made more effective with distributed scrum teams.

Xtreme Programming, or **XP**, is ideal for improving a product's responsiveness to altering customer needs. Development cycles tend to be shorter, allowing for more releases, where customer feedback can be gleaned, and the product can be adjusted accordingly. The shorter development cycles also contribute to lowering the cost of changes to the product's customer requirements.

Similar to XP, **Test-Driven Development**, or **TDD**, makes use of multiple, short development cycles. It involves the creation of a test — intended to fail, as the feature has not yet been implemented — for each new product feature. This is the aspect of TDD that allows the development team to think through the requirements and design a feature before writing the code for it. After the code has been written, it is tested and adjusted accordingly.

Behavior-Driven Development, or **BDD**, is another development model that reflects traces of TDD, but with modifications. BDD emphasizes the clarification of user stories, such that they relate directly to business outcomes, and it involves the implementation of only the features that contribute to those business outcomes. This model offers a more precise way to organize communication amongst testers, developers, and domain experts.

Feature-Driven Development, also known as **FDD**, combines best practices that emphasize features valued by customers. As indicated by its name, FDD revolves around a product's features. It consists of building a list of features to be included in the product, planning the development process by ordering them, designing how each will be created, and then building them.

Hyper-productivity

Hyper-productivity is a new scrum team's achievement of even higher team velocity, compared to that in their first sprint. It's above and beyond Scrum's typical velocity boost over the traditional waterfall methods. The desire to achieve hyper-productivity is one of the reasons that some organizations have distributed scrum teams. The driving notion is distributed team members' varying skills can contribute to quicker completion of certain tasks — thereby resulting in raised productivity.



Distributed Team Models

There exist multiple models that distributed scrum teams can follow in their organization. These models include isolated Scrum, distributed Scrum of Scrums (SoS), and integrated scrum teams.

Isolated Scrum

This formation, also known as IsoScrum, is an area-specific scrum and consists of multiple, individuated scrum teams with their own geographic location. Each scrum team is independent, complete with its individual Product Owner, with teams not generally interacting collaboratively.

Distributed Scrum of Scrums

DisSoS is similar to IsoScrum, in that there are full-standing scrum teams in each geographic location, but it differs in that teams do not work independently of one another, and normal Scrum of Scrum (SoS) meetings are conducted. The teams work collaboratively, which leads to the need for SoS meetings to coordinate their efforts.

Integrated Scrum Teams

InteScrum is a cross-functional and geographically distributed team model. It involves team members in each location, producing different features of a given product, requiring heavy communication and understanding across the board. Representatives of the scrum teams frequently meet to collaborate and plan.

Costs Affecting Outsourcing

The act of having distributed scrum teams beyond a central location has its benefits, a large one being lowering monetary costs in certain areas, but with additional overheads in others. Some of them being time and communication. Outsourcing the work for a scrum project increases the coordination costs; setting up daily meetings – as required in Scrum – is difficult, as team members are in different time zones, thus delaying communications. There may also be costs resulting from cultural barriers between team members and the PO. The PO's statement regarding a task may be interpreted differently from what was intended, depending on the team member's culture and how directly things are stated in that person's language. Team members of different locations may feel less unity as a group, due to cultural differences, which leads to a lack of inter-team dynamics. This lack of team solidity can negatively impact the self-organization of development team members, which would affect the project. These costs of outsourcing, if not properly addressed, can increase the chances of an unsuccessful scrum project.

IsoScrum and InteScrum – The Intent Behind These Models

IsoScrum

Given the fact that isolated scrum teams work independently from one another, the members of each team are co-located, which decreases the challenges of distributed scrum teams. The primary driver behind the IsoScrum team formation is this ease in communication, as face-to-face meetings are more easily feasible, and it increases the likelihood that members are on the same page. A leading cost associated with this model, though, is the lack of variety in development team members' abilities in a given location; not all of the necessary skills will be available by the members of a given site.

InteScrum

InteScrum is driven by the desire to have multiple skillsets on a given project. A drawback of this model arises from the integration of multiple, geographically different, scrum teams. It is difficult for scrum team members to be on the same page, given different cultures and time zones. This model might hamper lateral communication between members, as well.

Forces Driving Distribution

With the challenges of distributed Scrum teams, why would Product Owners look to distribute their development team members? Several reasons explain this.

One main reason is complexity, in terms of the portfolio or project. The project may require a wide range of skills or skills that are more abundant by individuals in another geographic location than the project's central location. With distributed teams, qualified individuals with skills to contribute to a project can be hired without consideration of whether or not they are in the project's central location.

Business goals or requirements can also affect the decision to have distributed teams, such as a small project budget that requires outsourcing the work. Offshore team members often have lower costs in terms of payment. This can be a large driving factor behind having distributed teams.

Solutions

Having distributed scrum teams is not without its challenges; however, these can be handled well if the team formation model is chosen well. Of the models, InteScrum and DisSoS, are those that are most successful for distributed teams, though InteScrum has more disadvantages than DisSoS.

InteScrum teams consist of development members in varying locations, and those in each location work together, independent of the teams on other sites, creating different features of a single product in each sprint; they then collaborate to integrate the features at the completion of each sprint. A key issue with Integrated Scrum is the potential for chaotic systems-level testing in each sprint, as the multiple teams must decide which one executes systems-level tests and handles the defects. Another disadvantage of InteScrum is that the integrated scrum team's composition can change from sprint to sprint, as each team is strong on certain features, and the features needed may differ from one sprint to the next. InteScrum also pales in comparison to DisSoS, because InteScrum magnifies communication issues amongst those involved in the scrum project. There is a lack of processes to enhance unity and communication amongst distributed team members, which is detrimental to project success.

Distributed Scrum of Scrums (SoS) is the most plausible option for those seeking to minimize the communication issues of distributed teams. It emphasizes Scrum-of-Scrum meetings to bring geographically different teams together in their work. Representatives from each Scrum meet daily to discuss what their respective teams had accomplished, what they will do, and what challenges they have met or expect. Product specifications and other requirements, along with progress updates, are more effectively communicated across the teams. This model is successful in that it enhances communication and ensures that teams have a common goal. The DisSoS model helps to mitigate the issue that many distributed Scrum teams face, through frequent and clear communications amongst the POs, Scrum Masters, and the development team members in various locations.

Summary

There are numerous reasons behind the desire to create distributed scrum teams, with some potential tradeoffs and challenges. The time and coordination costs of having a distributed team can be mitigated effectively with the most appropriate selection of team model. With a proper choice, distributed scrum teams can lead to immensely successful results.

