

Engagement Team Development

In every ISE engagement, dynamics are different so are the team requirements. Based on transfer learning among teams, we aim to build right "code-with" environments in every team.

This documentation gives a high-level template with some suggestions by aiming to accelerate team swarming phase to achieve a high speed agility however it has no intention to provide a list of "must-do" items.

Identification

As it's stated in Tuckman's [team phases](#), traditional team development has several stages. However those phases can be extremely fast or sometimes mismatched in teams due to external factors, what applies to ISE engagements.

In order to minimize the risk and set the expectations on the right way for all parties, an identification phase is important to understand each other. Some potential steps in this phase may be as following (not limited):

- [Working agreement](#)
- Identification of styles/preferences in communication, sharing, learning, decision making of each team member
 - Talking about necessity of pair programming
 - Decisions on backlog management & refinement meetings, weekly design sessions, social time sessions...etc.
 - Sync/Async communication methods, work hours/flexible times
- Decisions and identifications of charts that will be helpful to provide transparent and true information to everyone

- Identification of "Software Craftspersonship" areas which means the tools and methods will be widely used during the engagement and taking the required actions on team upskilling side if necessary.
 - GitHub, VSCode LiveShare, AzDevOps, necessary development tools & libraries ... more.
 - If upskilling on certain topic(s) is needed, identifying the areas and arranging code spikes for increasing the team knowledge on the regarding topic(s).
- Identification of communication channels, feedback loops and recurrent team call slots out of regular sprint meetings
- Introduction to [Technical Agility Team Manifesto](#) and planning the technical delivery by aiming to keep technical debt risk minimum.

Following the Plan and Agile Debugging

Identification phase accelerates the process of building a safe environment for every individual in the team, later on team has the required assets to follow the plan. And it is team's itself responsibility (engineers,PO,Process Lead) to debug their Agility level.

In every team stabilization takes time and pro-active agile debugging is the best accelerator to decrease the distraction away from sprint/engagement goal. Team is also responsible to keep the plan up-to-date based on team changes/needs and debugging results.

Just as an example, agility debugging activities may include:

- Dashboards related with "Goal" such as burndown/burnout, Item/PR Aging, Mood Chart ..etc. are accessible to the team and team is always up-to-date
- Backlog Refinement meetings
 - Size of stories (Too big? Too small?)
 - Are "User Stories" and "Tasks" clear ?
 - Are Acceptance Criteria enough and right?
 - Is everyone ready-to-go after taking the User Story/Task?

- Running [efficient retrospectives](#)
- Is the Sprint Goal clear in every iteration ?
- Is the estimation process in the team improving over time or does it meet the delivery/workload prediction?

Kindly check [Scrum Values](#) to have a better understanding to improve team commitment.

Following that, above suggestions aim to remove agile/team dysfunctionalities and provide a broader team understanding, potential time savings and full transparency.

Resources

- [Tuckman's Stages of Group Development](#)
 - [Scrum Values](#)
-

Last update: August 26, 2024