

| | Scrum | Kanban |
|------------|---|--|
| Difference | <ul style="list-style-type: none"> • More structured; • Based on short iterative sprints; • Has mandatory roles within the team members; • Focuses on workflow in a team; • Requires detailed planning for each sprint; is done by the whole team; • KPI: Burndown chart and Velocity chart; • 4 mandatory types of meetings; • The Scrum board is reset after each iteration; • Uses Story points to estimate work • For projects that often require changes; • Team is self-organized and small; • Prioritizes communication within the team; | <ul style="list-style-type: none"> • Continuous workflow; • Less prescriptive and more flexible; • Based on phases that go one after another; • Doesn't require any mandatory roles; • Focuses on visualizing all tasks • Planning is based on past workflow data; not so much on the team; • KPI: Lead time and Cycle Time; • 2 types of meetings(team-level cadences and service-oriented cadences) that can be skipped if they are not found necessary; • The Kanban board doesn't reset, it controls the amount of work that enters and leaves the process; • Instead of story points estimation, it uses historical data of actual work items; • Used to track and reduce inefficiencies; • WIP limits - used to control team's capacity; |
| Advantages | <p>→ Because of the short sprints, it is easier to adapt to any changes that may occur in the middle of the work.</p> <p>→ Is easy to shift the direction the project was initially going for;</p> <p>→ Because of the daily scrum meetings, every member knows what is going on with the project;</p> <p>→ Because of the sprint reviews and retrospectives, the team improves its work consistently because it goes through continuous feedback either from the stakeholders, or the team itself.</p> <p>→ Because of the short sprints, the team ensures that it will deliver a small working product at the end of every sprint;</p> <p>→ transparency for the client</p> | <p>→ Because it focuses so much on visualization, it's easy to understand from the start the tasks and how they are organized to be implemented;</p> <p>→ It's easy to understand from the start which tasks or phases need more care or more resources in order to be efficiently completed;</p> <p>→ High delivery speed because of the above mentioned; it's easy to visualize where a block could potentially be and ensure the fix of the problem from the start;</p> <p>→ Because it uses previous data of actual work items, it is more likely to anticipate the workload and assure a steadier flow;</p> <p>→ it's better at adapting to fluctuating</p> |

| | | |
|---------------|--|---|
| | | <p>needs;</p> <ul style="list-style-type: none"> → it's less stressful because there is no deadline approaching; → better for larger teams. → Because it lets team members work on one task at a time, a random change of plans mid-way work will not require to replan the whole work. |
| Disadvantages | <ul style="list-style-type: none"> → if the team doesn't anticipate their capacity correctly, the sprint might fail; → if any work appears in the middle of the sprint, it can only be started in the next sprint; → the team needs to be trained for such a framework; → because its process involves clients working closely, the team needs constant communication so it relies on it and can fall behind in case it is not constant; → it relies on the capacity of the team members so if a member leaves, it can set back the progress a lot. | <ul style="list-style-type: none"> → if the Kanban board is outdated, it could lead to problems in the development of the product; → there is no time frame associated with a development phase so it might mean that the work is done slower; → because of the lack of a deadline, the team might not react to urgent tasks. → because of the lack of programmed sprints, it's harder to measure progress and predict release dates; → might be difficult to prioritize incoming tasks. |