**Best practices for Agile project management**

**Configure your teams**

1. If you have a number of development or feature teams, define a team in for each feature team. This way, each team can work independently while collaborating with each other.
2. Define a team for each development group of six to twelve developers.

**Configure your sprints**

1. Define a sprint cadence that all teams within your product group will use
2. Define at least six or more iterations that will support planning for the next six to twelve months
3. Determine how teams will use iterations to manage backlog items:

Unassigned sprint work is assigned to the default backlog, or

Unassigned sprint work is assigned to a designated future backlog sprint.

**Choose the work item types you will use**

1. Use Features to capture customer features you want to ship
2. Quickly add features or requirements from the backlog and fill in details later
3. Use Requirements—User Stories (Agile), Issues (Basic), Product Backlog Items (Scrum), or Requirements (CMMI)—to break-down Features into work the development team will own
4. Use Bugs to capture code defects
5. Map Requirements to Features to track progress at the project management level
6. Set up Requirements to be completed within a sprint
7. Set up Features to be completed within a sprint or several sprints
8. Set up Epics to be delivered quarterly or to some milestone objective
9. Let Developers use Tasks to break-down their work as needed.

**Create your product plan**

1. Periodically refine your backlog
2. Make sure features and requirements are sized appropriately
3. Define the acceptance criteria and the definition of done for features and work
4. Map unmapped work to Features
5. Forecast your backlog

**Use tags to support queries and filtering**

1. Have a policy in place about how your teams will use tags
2. Indicate how you'll use tags to support queries, filtering, reporting
3. Consider using tags to identify cross-team or cross-project dependencies

**Forecast and milestone planning**

1. Assign estimate to each requirement to get a team velocity to forecast what features or requirements team can complete within the next six sprints.

**Manage dependencies**

1. Manage tasks that depend on the completion of other tasks by linking them. To manage dependencies, add parent/child link to work items.
2. Add a dependency tag to work items participating in dependency management
3. Use parent/child link types to track dependencies of work owned by other teams or within other projects
4. Create queries to track, add, and triage dependencies
5. Use the [Dependency Tracker](https://marketplace.visualstudio.com/items?itemName=ms-eswm.dependencytracker) Marketplace extension to view work that you have dependencies on from other teams
6. Use the [Work Item Visualization](https://marketplace.visualstudio.com/items?itemName=ms-devlabs.WorkItemVisualization) Marketplace extension to visualize dependencies

**Work in sprints**

Each sprint, perform the following tasks:

1. Plan each sprint with your team
2. Use the team's Sprint backlog to review sprint deliverables
3. Ensure each sprint work item is assigned to a team member
4. Ensure each work item is scoped to be completed within the sprint
5. Ensure the acceptance criteria for the work is well defined and understood
6. Update the status of sprint work items as work moves from new to Active to Completed state to track sprint burndown
7. Check in with other teams on dependencies that your team's work depends on
8. Monitor sprint progress using the Sprint burndown chart

**Review progress and feature deliverables**

1. Customize your Features Kanban board to support your team's processes
2. Add fields to cards so that you can update their values quickly and easily
3. Update the Iteration Path (Sprint) of Features as you gain clarity as to when they will ship
4. Review the Features board to talk through status, blocks/issues/risks/changes, and update status
5. Use the filter feature to focus on tagged items, assigned by features, a specific sprint and more
6. Add rollup columns to your Feature backlog to monitor overall progress based on work item count completion
7. Use delivery plans to review features being delivered by several teams and discuss cross-team dependencies

**Process improvement**

1. Identify process improvement goals that your team can agree to, write them down and review them periodically
2. Use team dashboards to share information and work tracking charts which you and your team review periodically
3. At sprint planning meetings, have your team identify at least one sprint goal related to process improvement
4. Conduct regular retrospectives to capture what went well, what didn't go well, and actions to improve
5. Maintain an improvement tracking board, such as that available with the [Retrospectives](https://marketplace.visualstudio.com/items?itemName=ms-devlabs.team-retrospectives) Marketplace extension.