*Sources - https://www.atlassian.com/agile/project-management/metrics*

*Metrics – Agile metrics give insight into the performance of a team or group of teams. They are meant to help teams analyze their productivity and effectiveness through different stages of their software development lifecycle.*

*Velocity*

*It measures the amount of work a team completes during a specific period, such as a sprint, and is usually measured in story points per sprint. It helps teams improve their planning and estimation processes and track their progress over time. By measuring how many work items can be completed during a specific period, teams can better understand their work capacity and adjust their plans accordingly. Tracking velocity can help teams identify areas where they may be struggling and take corrective action to improve their performance.*

*Cycle time -*

*Cycle time is a metric used to measure the time it takes for a team to complete a piece of work from the moment it enters their work in progress (WIP) queue until it is completed. This metric helps teams understand how long it takes for work to move through their process and identify areas for improvement. Measuring cycle time is an efficient and flexible way to improve a team’s processes. It allows them to make any further adjustments immediately. The end goal is to have consistent cycle times of acceptable length, regardless of the type of work.*

*Lead Time-*

*The lead time is the time it takes for a single unit of product to be created and added to the backlog when it is shipped. This is typically the time it takes for one project to be completed and sent to the customer. If you’re using a Kanban board, the lead time will begin once the item is added to the “To Do” list column.*

*In the marketing example scenario above, the lead time will end once all the content is published on the platform. By measuring lead time, you can determine how many items are flowing into your queue and how long it takes for your team to check them off.*

*Sprint burndown -*

*Scrum teams organize development into time-boxed sprints. At the outset of the sprint, the team forecasts how much work they can complete during a sprint. A sprint burndown report then tracks the completion of work throughout the sprint. The x-axis represents time, and the y-axis refers to the amount of work left to complete, measured in either story points or hours. The goal is to have all the forecasted work completed by the end of the sprint.*

*Epic and release burndown*

*Epic and release (or version) burndown charts track the progress of development over a larger body of work than the sprint burndown, and guide development for both scrum and kanban teams. Since a sprint (for scrum teams) may contain work from several epics and versions, it's important to track both the progress of individual sprints as well as epics and versions.*

*"Scope creep" is the injection of more requirements into a previously-defined project. While tolerating scope creep during a sprint is bad practice, scope change within epics and versions is a natural consequence of agile development. As the team moves through the project, the product owner may decide to take on or remove work based on what they're learning.*