Blocker vs. Dependency—Glossary of Terms

Blocker Definitions

1. **Blockers** are also known as impediments in Scrum, and issues in more traditional project management approaches.

2. **Blockers** are issues that prevent a team from completing a task or delivering a product increment. Blockers can be technical, like a software bug, or non-technical, like a lack of approval from a stakeholder.

3. **Blockers** are a specific issue that is preventing the team from completing a task or delivering a product increment. Blockers can be technical in nature, such as a software bug, or non-technical, such as a lack of approval from a stakeholder.

https://mudassiriqbal.net/impediments-obstacles-and-blockers

4. **Blockers** are obstacles that prevent a team member from making progress on their current tasks or user stories. They're akin to roadblocks on the highway of productivity, causing delays and sometimes bringing work to a complete standstill.

- A software bug that is too complex to fix without help
- A decision that needs to be made by someone outside the team
- A team member who is absent or ill and has specialized knowledge
- Unexpected technical issues like server downtime
- Unclear project specifications
- Dependency on another team member

https://www.spinach.io/blog/what-is-a-blocker-in-a-standup-meeting

5. **Blockers** might be an issue or bug which you have come across during development or testing and which is not allowing you to develop or test further.

6. **Blockers** don’t necessarily have to be a complete showstopper. Drag factors are also another form of blocker, that is, things that are slowing the team (or individual) down but not stopping them entirely.

https://www.solutioneers.co.uk/blockers

7. **Blockers** indicate a piece of work being blocked — the flow of work is impeded.

https://stefan-willuda.medium.com/being-blocked-it-s-not-what-you-might-think-f8b3ad47e806

8. **Blockers** are anything that blocks the team from making progress.

https://glossary.agileforgrowth.com/faqs/blocker

9. **Blockers** - When work cannot proceed because something, a 'blocker' is preventing it from doing so.

https://www.agilebusiness.org/knowledge-base/glossary-of-terms.html

10. **Blockers** or impediments are any issue or obstacle that prevents individuals or teams

from completing a task or project. Examples could be unscheduled meetings, technical issues, lack of knowledge or expertise, a distracting workplace, and office conflict.

https://www.agile-academy.com/en/agile-dictionary/impediment

11. **Blockers** or impediments refer to anything that obstructs or hinders a team's productivity, thus impacting the successful delivery of a product.

- **Missing or unavailable resources** − This could be anything from a missing team member or resource, like a tool or infrastructure required for the project.

- **Technical issues** − This could include unexpected bugs, issues with software or hardware, or problems with integrations between different components of the product.

- **Dependencies** − Unresolved dependencies between tasks, components, or teams can cause delays and impact the project's progress.

- **Incomplete or unclear requirements** − Unclear or incomplete requirements can lead to confusion and delays in development and testing.

- **Business or customer-related issues** − These could include changes in market or customer demands or unavailability of stakeholders for timely feedback.

- **Poor communication** − A lack of effective communication between team members, stakeholders, or other parties involved in the project can cause delays and misunderstandings.

- **Inefficient or inadequate processes** − Inefficient or inadequate processes can slow down the team's progress and impact the quality of the product.

- **Team issues** − These could include a lack of teamwork or collaboration, conflicting priorities, or low morale.

- **External factors** − External factors, such as a pandemic, natural disaster, or political changes, can also impact the project's progress.

https://www.tutorialspoint.com/impediments-in-agile-development

12. **Blockers** or impediments are technical, personal, or organizational issues that prevent progress on delivering products.

https://agilevideos.com/wp-content/uploads/2016/04/PMI-ACP_glossary.pdf

13. **Blockers** or impediments are factors causing a hindrance or blockage from performing scrum in an effective manner in a team or organization.

https://www.scrumstudy.com/freeresources/scrum-agile-glossary

Dependency Definitions

1. **Dependencies** are the relationships between work that determine the order in which the work items (features, stories, tasks) must be completed by Agile teams. Dependency management is the process of actively analyzing, measuring, and working to minimize the disruption caused by intra-team and or cross-team dependencies.

https://www.agilebusiness.org/knowledge-base/glossary-of-terms.html

2. **Dependencies** are situations where one activity or outcome relies on another activity or

outcome to be completed or delivered. For example, a Scrum team may depend on another team to provide an API, a vendor to deliver a hardware component, or a customer to approve a design. Dependencies can create risks, delays, and conflicts if they are not managed properly. Therefore, it is important to recognize and address dependencies as early as possible in the Scrum process.

https://www.linkedin.com/advice/1/how-do-you-manage-dependencies-between-scrum-artifacts-other

3. **Dependencies** in Scrum happen when one task or team needs another task or team to finish their work. There are different types:

- **Technical Dependencies**: When one task needs something from another task to move forward, like a front-end team needing a back-end team's work.

- **Sequential Dependencies**: Some tasks must be done in a certain order, like designing the user interface before doing front-end work.

- **Resource Dependencies**: Limited resources, like experts or special tools, can slow things down because teams have to wait for them.

- **External Dependencies**: Sometimes, Scrum teams depend on outside vendors or partners to provide something important.

- **Compliance Dependencies**: In projects with rules and regulations, tasks must be done in a certain order to follow the law.

https://targetagility.com/dependencies-in-scrum-how-theyre-handled

Dependency vs. Blocker

1. **Dependencies** are potential **blockers** (impediments). Some dependencies have an amazing power to consistently crush a team's ability to deliver. We want to eliminate or minimize dependencies, but what if you can't?

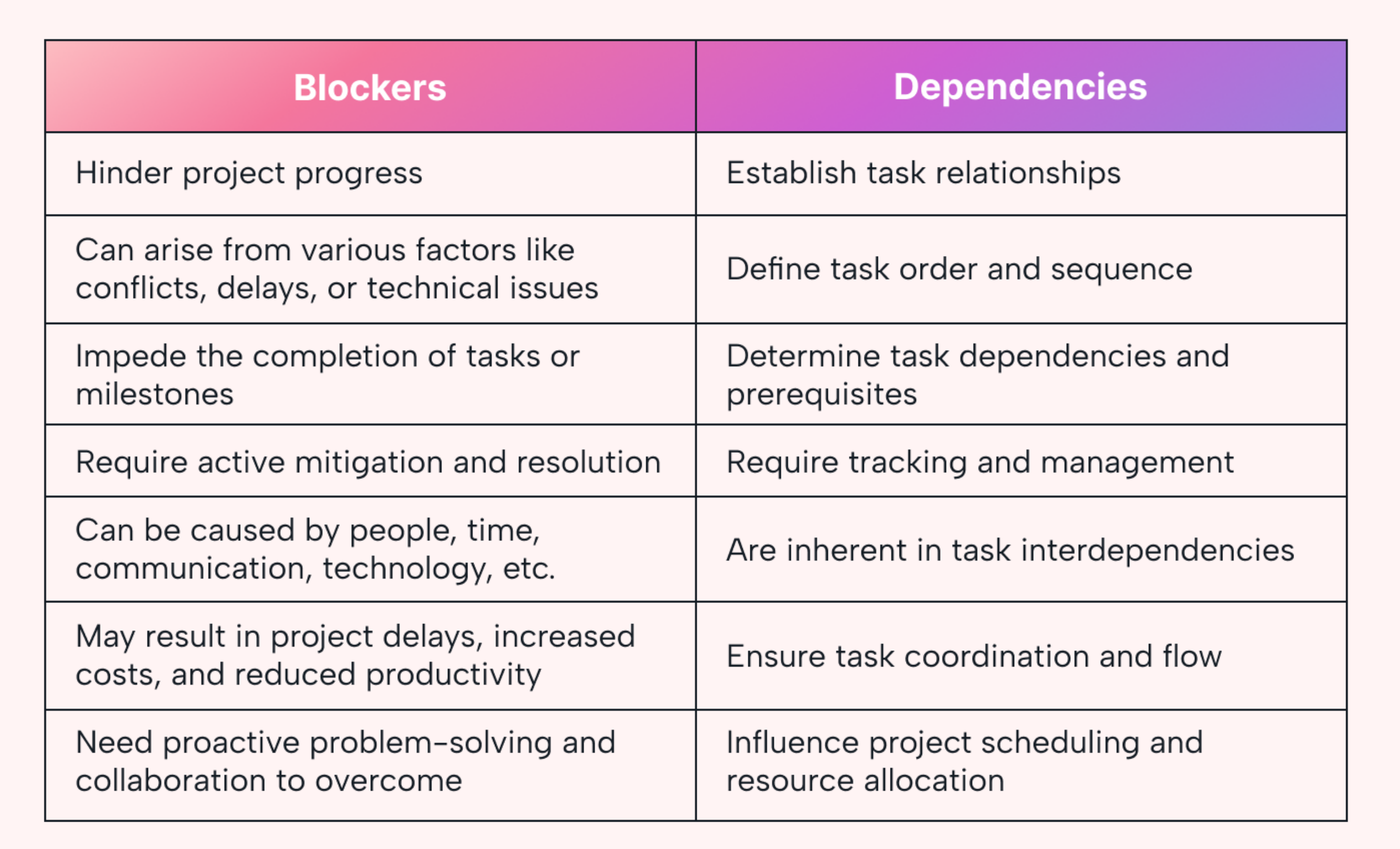
- **Intra-team dependencies** - The least challenging dependency scenario is where a single team has all the skills needed to produce a completed and potentially deployable product backlog item.

- **Inter-team dependencies** - Next, and more challenging is the multi-team environment where more than one team is working from the same product backlog. Each team can consume and deliver product backlog items to their completion, but sequencing or technical coordination across teams might be required.

- **Extra-team dependencies** - Finally, the most challenging case is where a scrum team cannot complete all of the required work necessary to deliver a product backlog item. They need assistance or support from a) another scrum team, b) another department, or c) an external vendor.

https://resources.scrumalliance.org/Article/managing-dependencies-scrum

2. **Dependencies** are relationships between activities while **blockers** are current work stoppages. Blockers are obstacles or challenges that arise and prevent progress on a project. They can sometimes be created by dependencies, but often are caused by other events. In contrast, dependencies refer to relationships between tasks or activities, similar to those we discussed in the baking example. Dependencies can impact not only the dependent tasks themselves, but also can affect resources, schedules, and cost.



Blockers	Dependencies
Hinder project progress	Establish task relationships
Can arise from various factors like conflicts, delays, or technical issues	Define task order and sequence
Impede the completion of tasks or milestones	Determine task dependencies and prerequisites
Require active mitigation and resolution	Require tracking and management
Can be caused by people, time, communication, technology, etc.	Are inherent in task interdependencies
May result in project delays, increased costs, and reduced productivity	Ensure task coordination and flow
Need proactive problem-solving and collaboration to overcome	Influence project scheduling and resource allocation

https://www.usemotion.com/blog/project-blockers

3. **Dependencies** are needs upon other teams or parties, **blockers** are current stoppers or obstructions, and **impediments** are temporary obstacles, delays, or slowdowns (like traffic congestion or accidents).

• **Dependency** – State of being dependent or subordinate; appurtenance. A dependency puts another party on the line for taking responsibility or accountability for something tangible. If a project is dependent on something, the project cannot be completed without that dependency being delivered. You can layer risk over this dependency by stating, the risk would be unsure of the delivery date of this dependency simply because the delivery date is out of your control and in the control of whoever this dependency falls on.

• **Blocked** – To act as to stop or obstruct someone or something, obstacle in the way. Preventing flow. To be blocked is to state that whoever you are dependent on is greatly impacting your ability to meet a commitment, deadline or ultimately stops your flow. The criticality of something that is blocked is much higher when you are closer to a commitment date or deadline. To be blocked means, you cannot work on something you have in your possession until the “dependent” blocker has been delivered. A risk could be raised early in a project to say, “We could be blocked on this project if this dependency is not delivered in time.”

• **Impediment** – Obstruction; hindrance; obstacle; Impede – To delay or slow down. Impediment means to simply obstruct or hinder someone or something. An impediment can happen before something is even blocked if we honor the true spirit of the definition. As an Agilist or on a project, you’d want to know when something is obstructed so you can work toward mitigating this obstruction before it’s officially blocked.

https://www.linkedin.com/pulse/big-project-impacts-commonly-confused-terminology-up-jeffrey