Dependency Management in Agile Teams

Santoshanand Adkar,
Agile Thought Leader

Abstract: With more and more teams working towards a common goal and common product, it becomes imperative that the dependencies within these teams are managed well and in advance making sure that the dependencies are addressed in timely manner thus avoiding the inevitable delays in delivering the project. Dependencies which can be seen early in the life cycle can be taken care of, however the dependencies that come on the fly or during the execution, becomes challenge. Is there any way we can manage these dependencies in a structured manner? Can all the dependencies be taken care of? How to resolve the conflicts that arise because of these dependencies?

Here is a paper which talks about dealing and managing with Dependencies within the teams and across the teams who are working on a common product and who are getting benefited by that product. The reason to mention regarding the beneficiaries is that they too have dependencies on the product in direct or indirect way. Example can be of futuristic dependencies which can get addressed in timely manner so that once the feature is launched, the teams can get benefit from it.

I. INTRODUCTION

With the increasing complex nature of the work and the products that we are building for our customers, dependencies within and across the teams has been playing a very important role. There has been a substantial number of delays in delivering a specific product even in its Minimal Viable form because the dependencies that we have within the teams are not getting addressed in timely manner. Some known dependencies are addressed early in the cycle however we are not able to address the unknown dependencies because of the nature of the work and product. With the “Just Enough Design (JED)” approach, we can identify the probable dependencies.

Type of Dependencies

Dependencies are generally categorized as:

Internal Dependencies: Dependencies between teams within the ART if you are using SAFe Framework, Dependencies between tasks within the same Scrum team if you are using only Scrum Framework, Dependencies within a single team, often related to tasks, resources, or knowledge sharing, dependencies within and between internal teams. This includes resource dependencies, task dependencies, and inter-team coordination needs.

External Dependencies: Dependencies on teams or services outside the ART, including vendors or external systems.Dependencies on other Scrum teams, external teams, or third-party services, Dependencies between different teams within the same organization or project, Dependencies involving external stakeholders, vendors, third-party services, or systems outside the immediate project scope, dependencies on external vendors, partners, or other departments outside the immediate project team.

Role wise Dependency Management

The dependency management depends on the various roles that are getting played within the agile teams. If we consider SAFe then as an **Release Train Engineer** below are the steps for managing the dependencies:

1. Identify the Dependencies: Identify both the internal and external dependencies.
2. Categorize the Dependencies: Categorize the dependencies in the below categories:
    a. Mandatory vs. Discretionary: Identify which dependencies must be addressed first before proceeding ahead and which can be marked as optional or may not be addressed immediately.
    b. Critical Path Dependencies: When we are in the process of delivering something which is critical, we need to address the dependencies first which are on critical path as this might turn into a showstopper. Timelines are utmost important here and needs to have urgent attention.
    c. Risk-Based Dependencies: Identify the dependencies that could introduce significant risks if not managed in timely manner and with proper plan in place.
3. Visualize Dependencies: It’s important to visualize the dependencies as these need to be addressed in timely manner.
    a. Create a Dependency Board: Use tools like Jira, Trello, or Miro to visualize your dashboard and discuss this during your regular events.
    b. Program Board: During PI Planning create the program board and address the dependencies and get alignment from the team.
4. Plan & Prioritize: We need to plan and prioritize the dependencies based on the product and customer necessities.
    a. During PIPlanning: Use the PI Planning event to identify, discuss, and plan for dependencies. Prioritize dependencies based on their impact on the ART's objectives.

   b. Milestones and Checkpoints: Establish key milestones and checkpoints to monitor dependency progress and resolve issues proactively.

5. Communicate and Coordinate: All the dependencies must be communicated and coordinated to make sure the dependencies are addressed in timely manner.
   a. Regular Sync Meetings: Conduct regular sync meetings with Scrum Masters, Product Owners, and relevant stakeholders to review and address dependencies.
   b. Dependency Management Team: If possible, create a dedicated team or group to manage and track dependencies across the ART.
6. Dependency Risk Management: Identify the mitigation plan for the risks associated because of the dependencies.
   a. Risk Identification: Identify risks associated with dependencies, including delays or blockers.
   b. Mitigation Plans:Develop mitigation plans for high-risk dependencies and ensure these plans are communicated and understood by all relevant teams.
7. Dependency Tracking and Reporting: Dependencies need to be tracked and reported to the respective stakeholders making sure the dependencies are highlighted at the right level.
   a. Tracking Tools:Use tracking tools to monitor the status of dependencies. Jira, Azure DevOps, and other project management tools can be customized for this purpose.
   b. Reporting:Provide regular updates on dependency status during ART syncs, Scrum of Scrums, and other relevant meetings.
8. Improvisation of Dependency management: We need to look for ways to improvise the process of dependency management with the team.
   a. Iteration Retrospectives: Conduct retrospectives to reflect on how well dependencies were managed and identify areas for improvement.
   b. Feedback Loop: Establish a feedback loop to continuously refine the dependency management process based on learnings from each PI.
9. Leverage the SAFe Principles: Make sure the team understands the SAFe principles with respect to the Dependencies.
   a. Alignment: Ensure all teams are aligned with the ART's objectives and understand the importance of managing dependencies.
   b. Built-in Quality: Focus on quality to prevent issues related to dependencies.
   c. Transparency: Maintain transparency across the ART regarding the status and impact of dependencies.

If you are not using SAFeor any other scaling framework, but just Scrum framework, then as a **Scrum Master**, you can follow the below steps for dependency management:

1. Identify the Dependencies: Identify both the internal and external dependencies within the scrum team or other scrum teams, external teams, or third-party vendor services.
2. Categorize the Dependencies: Categorize the dependencies in the below categories:
   a. Critical vs. Non-Critical: Determine which dependencies are critical for the delivery of key features or user stories.
   b. Blocking vs. Non-Blocking: Identify dependencies that can block progress and those that are less likely to cause delays.
3. Visualize & Map Dependencies: It’s important to visualize the dependencies as these need to be addressed in timely manner.
   a. Create a Dependency Board: Use tools like Jira, Trello, or Miro to visualize your dashboard and discuss this during your regular events.
   b. Dependency Matrix:Create a matrix to map dependencies, showing the relationship between different tasks and teams.
4. Plan & Prioritize: We need to plan and prioritize the dependencies based on the product and customer necessities.
   a. Sprint Planning:Discuss and plan for dependencies during Sprint Planning. Ensure that the team is aware of dependencies and plans accordingly.
   b. Backlog Refinement: Regularly refine the backlog to identify and address dependencies early. Collaborate with Product Owners to prioritize tasks with dependencies.
5. Communicate and Coordinate: All the dependencies must be communicated and coordinated to make sure the dependencies are addressed in timely manner.
   a. Daily Stand-ups: Use daily stand-ups to discuss and track the status of dependencies. Ensure team members communicate any issues or blockers.
   b. Cross-Team Coordination: If dependencies involve other teams, schedule regular coordinationmeetings, or establish a Scrum of Scrums to manage cross-team dependencies.
6. Dependency Risk Management: Identify the mitigation plan for the risks associated because of the dependencies.
   a. Risk Identification: Identify risks associated with dependencies, including delays or blockers.
   b. Mitigation Plans:Develop mitigation plans for high-risk dependencies and ensure these plans are communicated and understood by all relevant teams.
7. Dependency Tracking and Reporting: Dependencies need to be tracked and reported to the respective stakeholders making sure the dependencies are highlighted at the right level.

   a. Tracking Tools:Use tracking tools to monitor the status of dependencies. Jira, Azure DevOps, and other project management tools can be customized for this purpose.
   b. Reporting:Provide regular updates on dependency status during ART syncs, Scrum of Scrums, and other relevant meetings.
8. Improvisation of Dependency management: We need to look for ways to improvise the process of dependency management with the team.
   a. Iteration Retrospectives: Conduct retrospectives to reflect on how well dependencies were managed and identify areas for improvement.
   b. Feedback Loop: Establish a feedback loop to continuously refine the dependency management process based on learnings from each PI.
9. Foster a collaborative Culture: Make sure the team members maintain the collaborative culture towards dependency resolution.
   a. Encourage Transparency: Foster an environment where team members feel comfortable sharing information about dependencies and potential issues.
   b. Build Trust: Encourage collaboration and trust among team members and across teams, ensuring that dependencies are addressed openly and proactively.
10. Leverage Agile Practices: As a Scrum Master leverage the agile practices and principles within the teams.
   a. Agile Principles:Apply Agile principles such as delivering working software frequently and embracing change to manage dependencies effectively.
   b. Flexibility:Be flexible and adaptive in managing dependencies, as they may change during the course of the project.

As a **Delivery Manager** it becomes more imperative to take care of the dependency management as it leads to the delivery management and if dependencies not identified in time can lead to delays thus impacting the customer deliverables.

Below are the steps that can be taken by the delivery manager to manage dependencies:

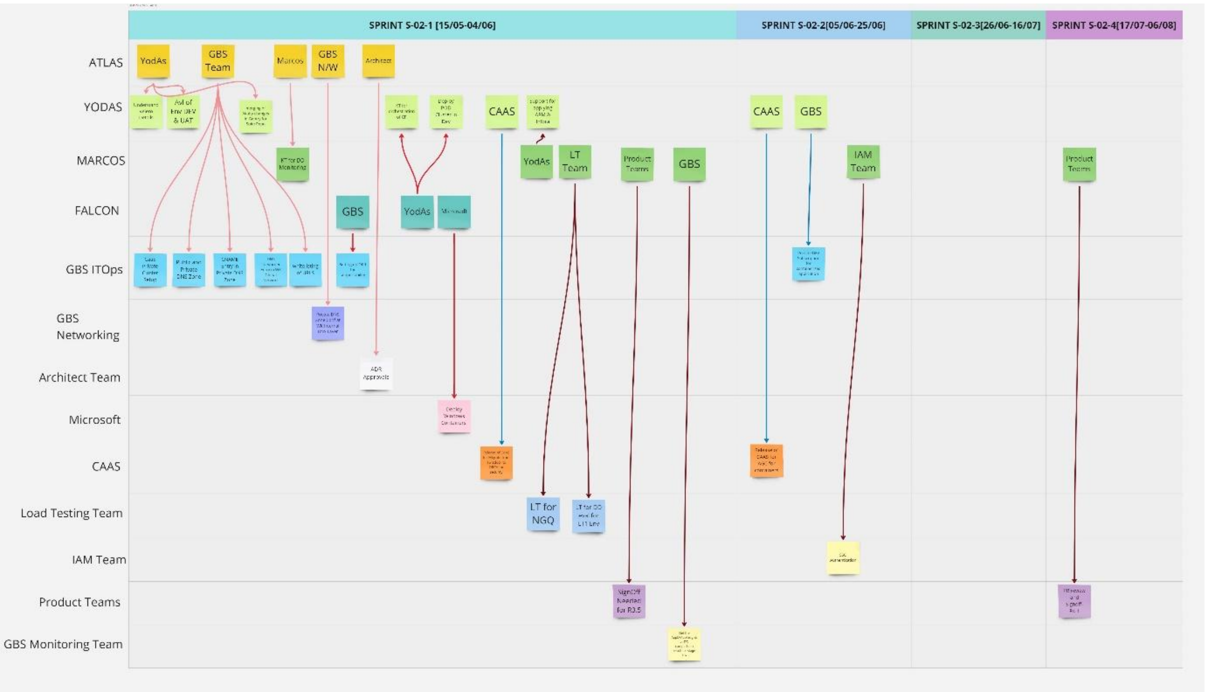
1. Understand the Scope and Context: It becomes important for the delivery manager to understand the clear definitionof the scope of the project or product, including key deliverables, timelines, and stakeholders. Additionally, the Delivery Manager need to take cognizance of the organizational context with respect to the organizational structure, context the teams involved, their roles and the interdependencies between them.
2. Identify the Dependencies: Identify the internal dependencies within and between internal teams which includes resource dependencies, task dependencies, and inter-team coordination needs. and external Dependencies on external vendors, partners, or other departments outside the immediate project or product team(s).
3. Categorize the Dependencies: Categorize the dependencies in the below categories:
   a. Critical vs. Non-Critical: Identify which dependencies are critical for project or product delivery and which are less critical.
   b. Blocking vs. Non-Blocking: determine which dependencies can block progress of the delivery of the project or product or teams to proceed ahead and need immediate attention.
4. Visualize & Map Dependencies: It’s important to visualize the dependencies as these need to be addressed in timely manner.
   a. Create a Dependency Board: Use tools like dependency matrices, Gantt charts, or network diagrams to map dependencies clearly.
   b. Visualize Dependencies:Employ project management software (like Jira, MS Project, or Asana) to create visual representations of dependencies, making it easier to track and manage them.
5. Plan & Prioritize: We need to plan and prioritize the dependencies based on the product and customer necessities.
   a. Product / Project Planning:Incorporate dependencies into the project plan, highlighting critical paths and key milestones.
   b. Risk Assessment: Prioritize dependencies based on their potential impact on project timelines and deliverables. Identify high-risk dependencies that require close monitoring.
6. Communicate and Coordinate: All the dependencies must be communicated and coordinated to make sure the dependencies are addressed in timely manner.
   a. Regular Meeting: Establish regular coordination meetings, such as project syncs, Scrum of Scrums, or program interval planning sessions, to discuss and manage dependencies.
   b. Stakeholder Management: Ensure open communication with stakeholders about dependency status, potential risks, and mitigation strategies.
7. Dependency Risk Management: Identify the mitigation plan for the risks associated because of the dependencies.
   a. Risk Identification: Recognize potential risks associated with dependencies, such as resource constraints, technical challenges, or delays.
   b. Mitigation Plans:Develop and implement mitigation plans for high-risk dependencies. This may involve contingency planning, resource reallocation, or timeline adjustments.
8. Dependency Tracking and Reporting: Dependencies need to be tracked and reported to the respective stakeholders making sure the dependencies are highlighted at the right level.

   a. Tracking Tools:Utilize project management tools to track the status of dependencies. Implement dashboards or status reports to maintain visibility.
   b. Monitoring Progress:Regularly review the status of dependencies during project review meetings and adjust plans as necessary.

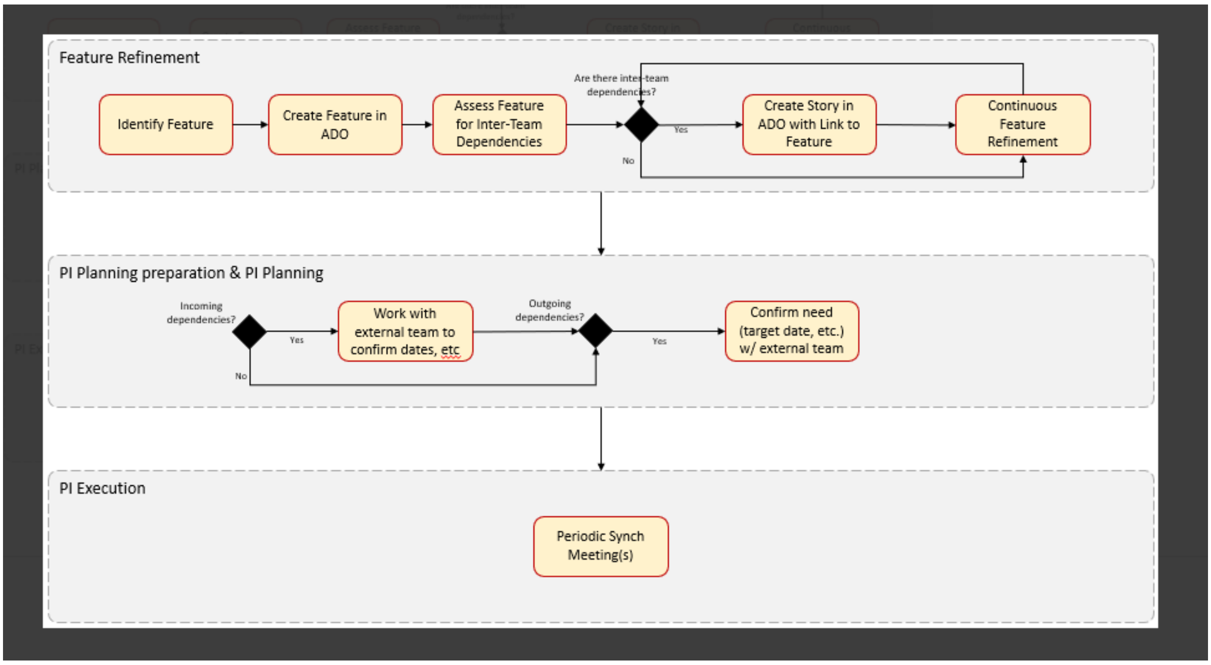
9. Improvisation of Dependency management: We need to look for ways to improvise the process of dependency management with the team.
   a. Root Cause Analysis: Conduct Root Cause Analysis and post-project reviews to evaluate how dependencies were managed. Identify lessons learned and areas for improvement.
   b. Feedback Loop: Implement a feedback mechanism to continuously refine and improve the dependency management process.
10. Foster a collaborative Culture: Make sure the team members maintain the collaborative culture towards dependency resolution.
   a. Build a Collaborative Environment:Promote a culture of transparency, open communication, and collaboration across teams.
   b. Empower Teams:Encourage teams to proactively identify and communicate dependencies and potential issues early in the process.
11. Leverage Technology and Best Practices: Implement the best practices that are followed within the organization or industry best practices by leveraging latest technology.
   a. Use Appropriate Tools:Utilize the right project management and collaboration tools to streamline dependency tracking and communication.
   b. Adopt Best Practices: Stay updated with industry best practices and frameworks, such as Agile, Lean, or SAFe, to optimize dependency management.
   c. Agile Frameworks:Utilize Agile frameworks like Scrum, Kanban, or SAFe to structure the dependency management process, aligning with the principles of iterative development and continuous delivery.

II. EXAMPLE OF DEPENDENCY MANAGEMENT

Using MIRO Tool:



Using Azure DevOps (ADO)


Feature Refinement
Identify Feature
Create Feature in ADO
Assess Feature for Inter-Team Dependencies
Are there inter-team dependencies?
Yes
No
Create Story in ADO with Link to Feature
Continuous Feature Refinement
PI Planning preparation & PI Planning
Incoming dependencies?
Yes
No
Work with external team to confirm dates, etc
Outgoing dependencies?
Yes
Confirm need (target date, etc.) w/ external team
PI Execution
Periodic Synch Meeting(s)

CONCLUSION

By implementing the above-mentioned dependency management strategies, we can ensure that all dependencies are identified, tracked, and managed effectively, minimizing risks and facilitating smooth project delivery.

This holistic approach helps mitigate risks, resolve issues proactively, and foster a culture of collaboration and continuous improvement.

We can help our Scrum teams deliver more efficiently and reduce the risk of delays or bottlenecks. This approach ensures that the team can effectively identify, plan for, and manage dependencies throughout the development process.

References

[1] Google
[2] "Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise" by Dean Leffingwell
[3] "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford
[4] "Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework" by Mik Kersten
[5] "Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum" by Craig Larman and Bas Vodde