



INNOLUTION DEPENDENCIES ARE KILLING YOUR AGILITY: LEARN TO FIGHT BACK!



www.innolution.com



hello@innolution.com



303 827-3333



[@krubinagile](https://twitter.com/krubinagile)



[kennethrubin](https://www.linkedin.com/in/kennethrubin)

Module	Description
Course Introduction	<ul style="list-style-type: none"> • Introductions • Overview of the course agenda • Collection of student discussion topics
What Are Dependencies	<ul style="list-style-type: none"> • Definition of dependencies • Scope of coordination • Shared dependencies • Introducing the concept of N (the cardinality of the dependence set)
Why Are Dependencies Important	<ul style="list-style-type: none"> • Misaligned dependencies block the flow of work • Blocked work is inventory that accumulates • Large amounts of inventory (WIP) cause increasing level of delay cost • Large cost of delay will cause you to fail or substantially underperform your target outcomes
Dependencies At Scale	<ul style="list-style-type: none"> • Dependency permutations (possible outcomes) • The math really is that bad! • How does any organization get things done? • A single team with a few dependencies on other unburdened teams is a solvable problem • At scale, however, there can be a staggering number of dependencies • Prioritization is the killer problem • Trying to prioritize across multiple teams each with multiple competing requests is nearly impossible • A visible representation of your dependencies does not solve the prioritization problem at scale
Need to Solve for Dependencies in the Large	<ul style="list-style-type: none"> • Creating a dependency solution for a small subset of your organization doesn't really solve the flow problem • Need to take a more end-to-end business perspective • At the larger perspective, projects prove to be a poor unit of focus • We need something more long-lived like products, capabilities, value streams



Module	Description
Dependency Myths	<ul style="list-style-type: none"> Identifying the dependencies mostly solves the problem Better project management will solve the “when” problem Centralized demand management is the solution Escalation will solve the dependency prioritization problem Feature teams are the full solution All team / product interaction models will address dependencies No need to address system or architectural issues
Dependency Improvement Patterns	<ul style="list-style-type: none"> Overview of pattern concepts Pattern – Create feature teams Pattern – Intra coordinated cluster Pattern – Build using (self service) Pattern – Don’t count nice-to-have dependencies Pattern – Dedicated capacity Pattern – Named individual Pattern – Named and expedited Pattern – Normal SLA Pattern – Expedited SLA
Analysis of Example Solution at Scale	<ul style="list-style-type: none"> Addressing dependencies at scale requires the use of multiple patterns Example of a company’s solution Analysis of the solution
Closing	<ul style="list-style-type: none"> Review of the class Final Q&A

