

# One Model of an Effective Inter-Team Process

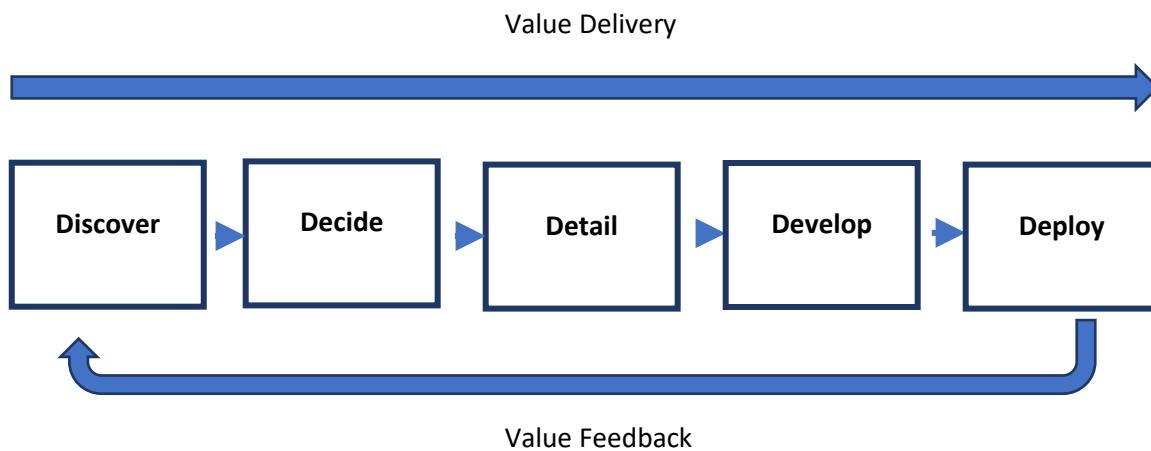
Ken Pugh, kenpugh.com [ken@kenpugh.com](mailto:ken@kenpugh.com)

Following the ideas in [One Model for Effective Software Development](#), this article presents one model for an Effective Inter-team Process. It provides an overview of the value delivery process (e.g., developmental value stream) for a software project that requires multiple teams.

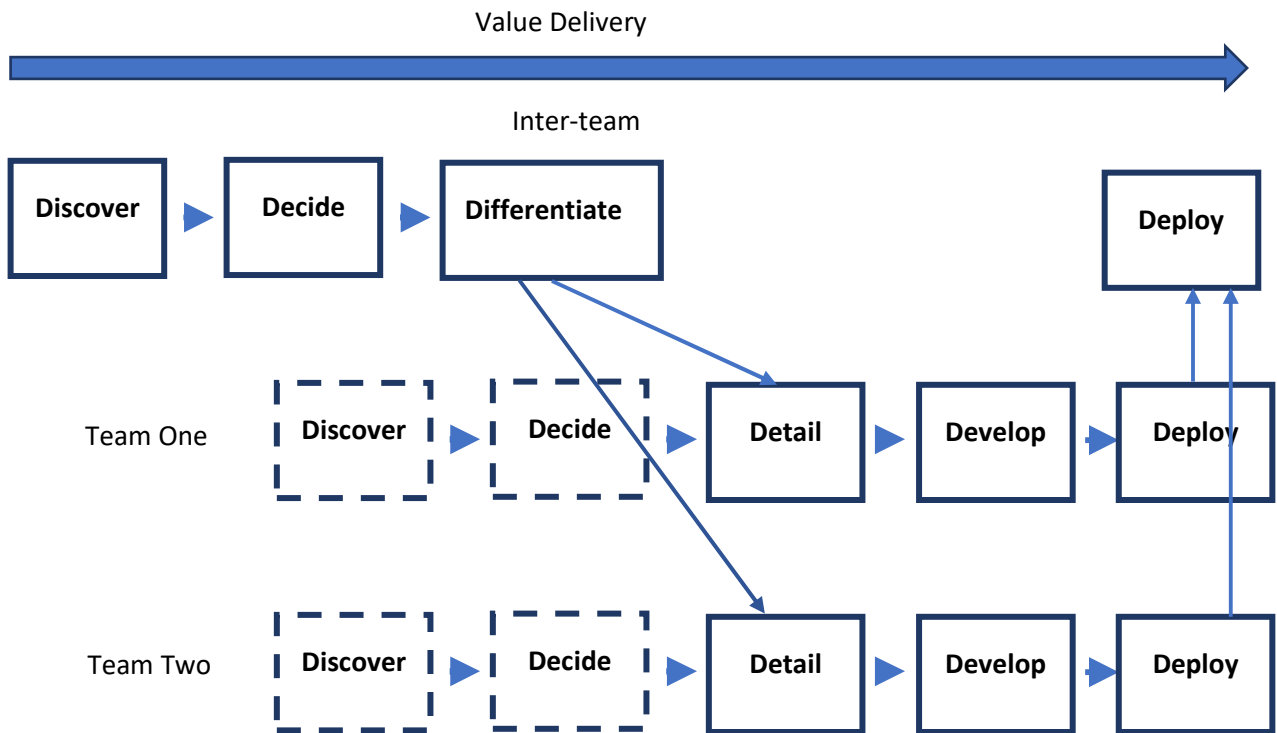
Why does some software development require multiple teams?

- The amount of time to produce the software for a single team would not allow it to deliver value in a reasonable time frame
- Multiple different team capabilities are required, e.g., a product that has embedded software and a web interface

Here's the flow for one team, as shown in [One Model for an Effective Team Process](#).



For an Inter-team process, the discovery and deciding on the value delivery elements may occur at a higher level. Once the decision is made, the value element is differentiated into implementation components that are created by the separate teams. The flow is not necessarily linear. For example, there can be cycles where differentiation occurs when new information emerges.

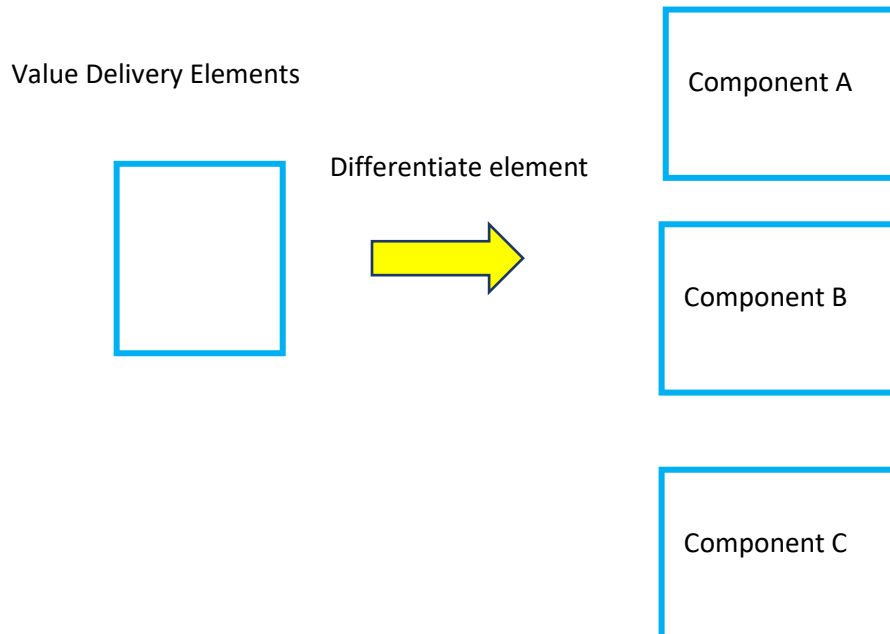


The first two steps are like ones for a single team. With inter-teams, they deal with elements that require multiple teams. Each team may discover and decide on elements that just involve themselves.

- Discover
  - What is important – what are the potential value delivery elements (e.g., capabilities, features)
  - May involve users (e.g., defect reports), Hypothesis Driven Development
- Decide
  - What value elements should be delivered and in what order
  - May use estimates of value and implementation (e.g. WSJF, ROI, BfB )
- Differentiate (on the inter-team level)
  - Delineate the responsibilities for the components that teams will develop
  - Determine which teams develop which components
- Deploy (on the inter-team level)
  - Delivers the combined components delivered by the individual teams so that value is available

## Differentiate

The value delivery elements are differentiated into components that developed by teams.



The inter-team development has aspects of:

- Cadence / Schedule includes
  - Periodic synchronization
- Quality includes:
  - Was the component build correctly?
- Lean aspects of inter-team delivery include:
  - Minimize the coupling between components
  - Minimize the effort (e.g., automated deployments)

## Deploy

Individual teams deploy their components and then the combination delivers the value

Component A

Component B

Component C

Implementation Components

Deploy



Deliver Value

