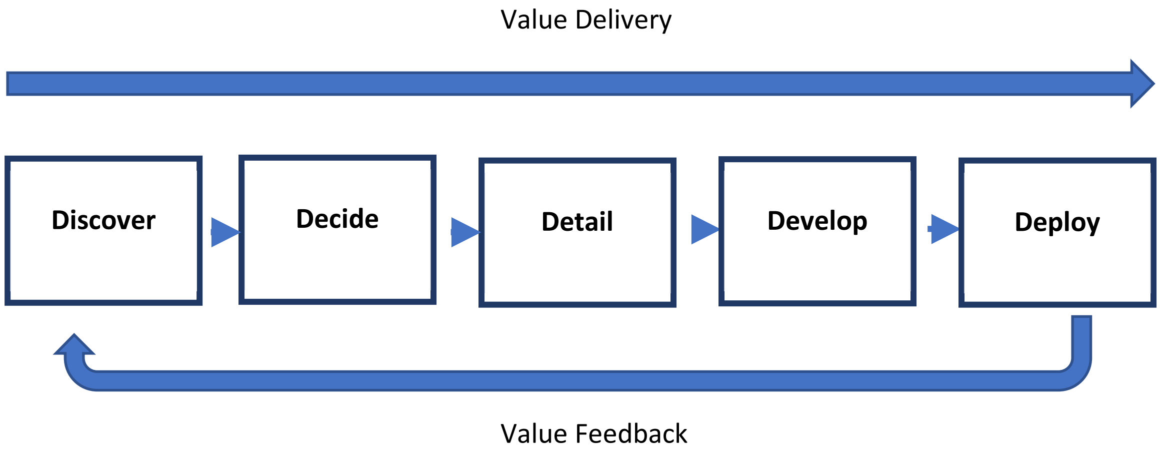
One Model of an Effective Team Process

Ken Pugh, https://kenpugh.com ken@kenpugh.com

Following the ideas in *One Model for Effective Software Development*, this article presents one model for an Effective Team Process. It provides an overview of the value delivery process (e.g., value stream) for software.

Depending on the context, value delivery

- May be done within minutes to hours to days to weeks
- Be by a small group, one team, or a group of teams



The steps within this flow are shown as linear, but there may be cycles between steps or combinations of steps (e.g. Detail and Develop). The steps may involve experimentation to explore alternatives.

- Discover
  - What is important – what potential value delivery elements (e.g. features, stories)
  - 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)
- Detail
  - The value elements (e.g., stories, scenarios)
- Develop

    - Decide what new or changed implementation components are needed for the value elements and make them
- Deploy
    - Deliver the value element so that value is available

Quality includes:

- Was it the right element?
    - The feedback arrow can track whether the value is delivered
- Was the element build correctly?
    - May use Test-first (TDD / BDD / ATDD)

Cadence / Schedule includes:

- Discovery / Decide / Detail / Develop / Deploy may have same or different cadences
- Periodic synchronization (e.g., daily standup)

Lean aspects of delivery include:

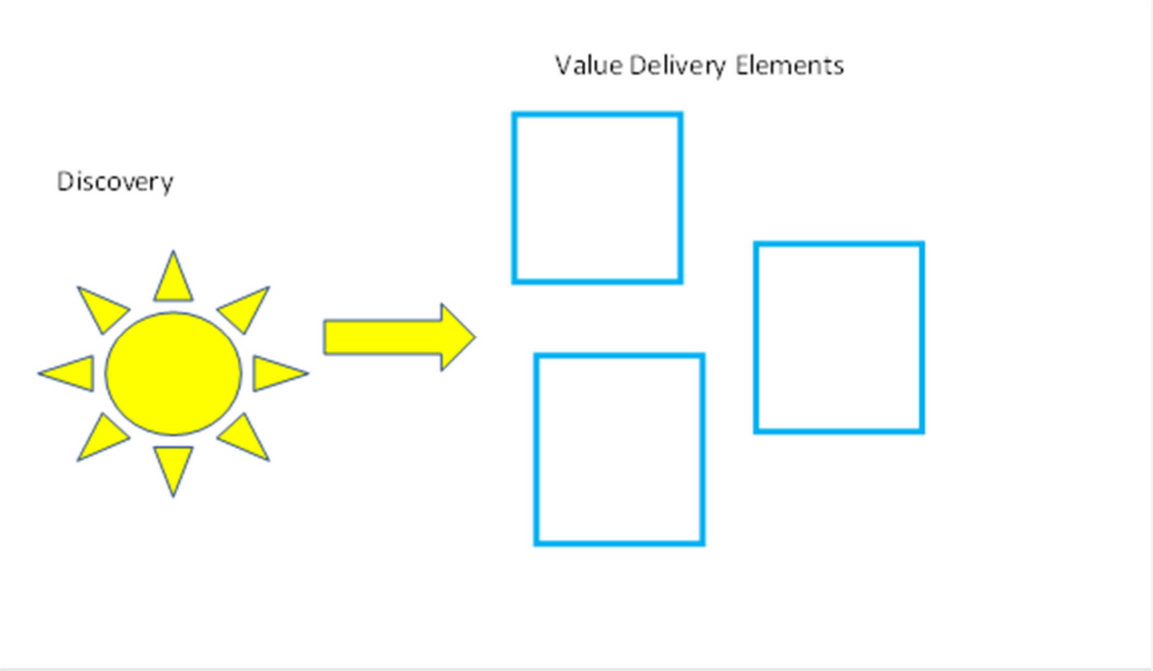
- Minimize the handoffs between the steps
- Minimize the effort (e.g., automated deployments)

Discover

Find out what value elements (e.g. features, stories, defect removals) might be delivered

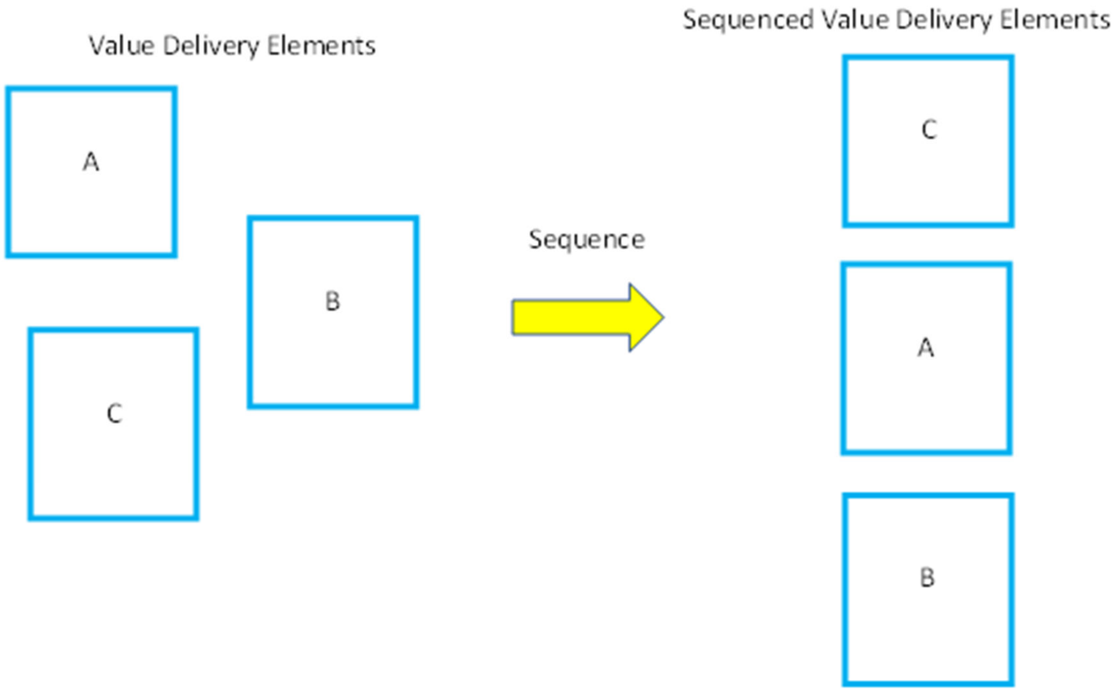
Value may be

- Business value
- Customer value
- Technical value



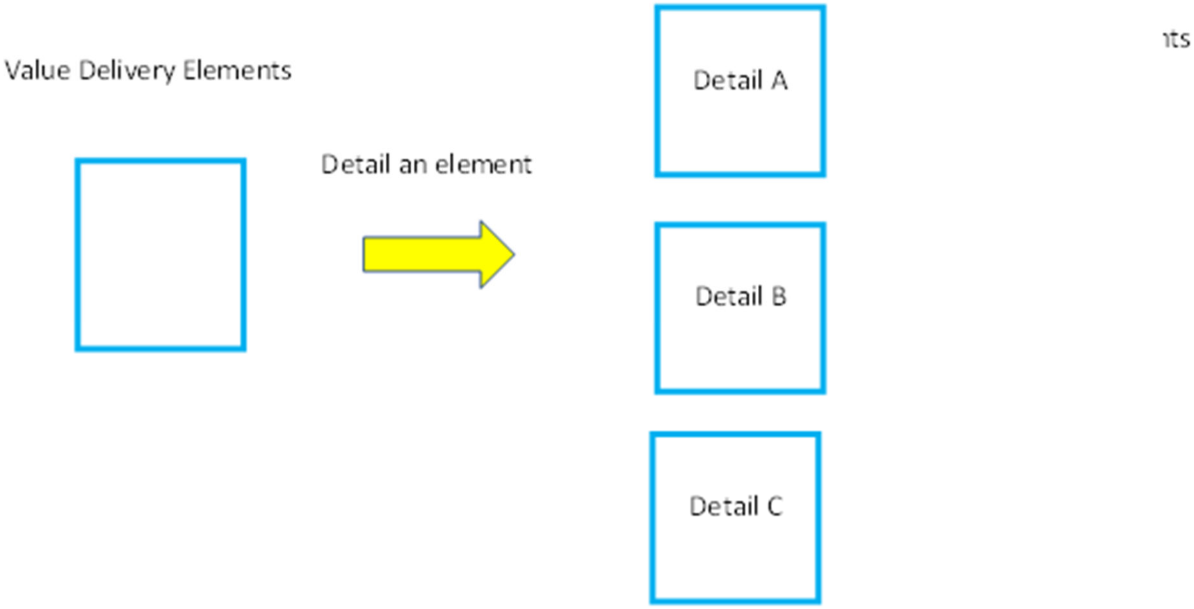
Decide

Determine what value elements (e.g. features, stories, defect removals) to deliver and in what sequence.



Detail

Decompose value elements into details (e.g., stories, scenarios)



Scheduling

As part of detailing, the details may be scheduled

- In iterations (e.g. Scrum)
- Iterationless (e.g. Kanban)

Scheduling can be based on

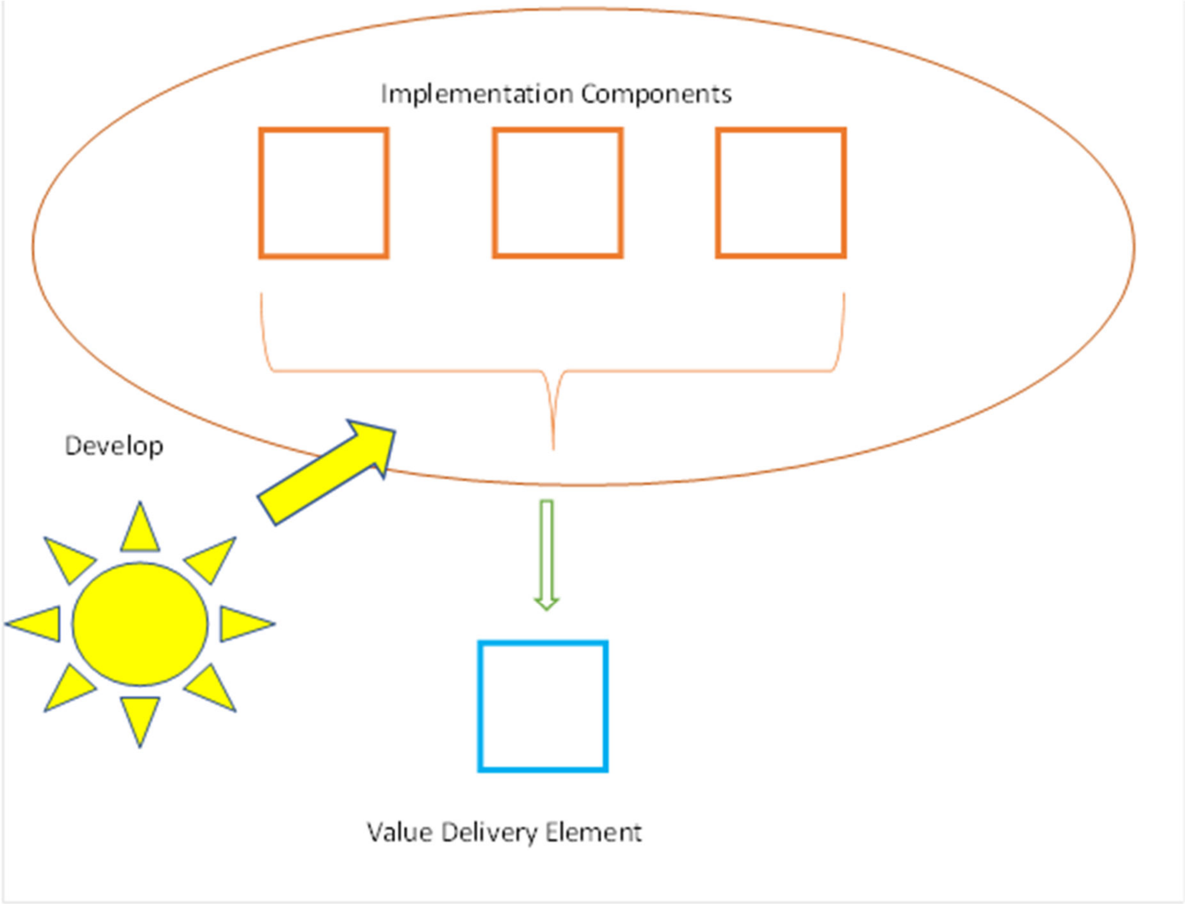
- Capacity of the team
- Allowance for sustaining team and implementation components

Develop

Design, code, and test the implementation components (e.g., modules, classes, methods, interfaces) and changes to existing components that together provide the value delivery element or a detail of it.

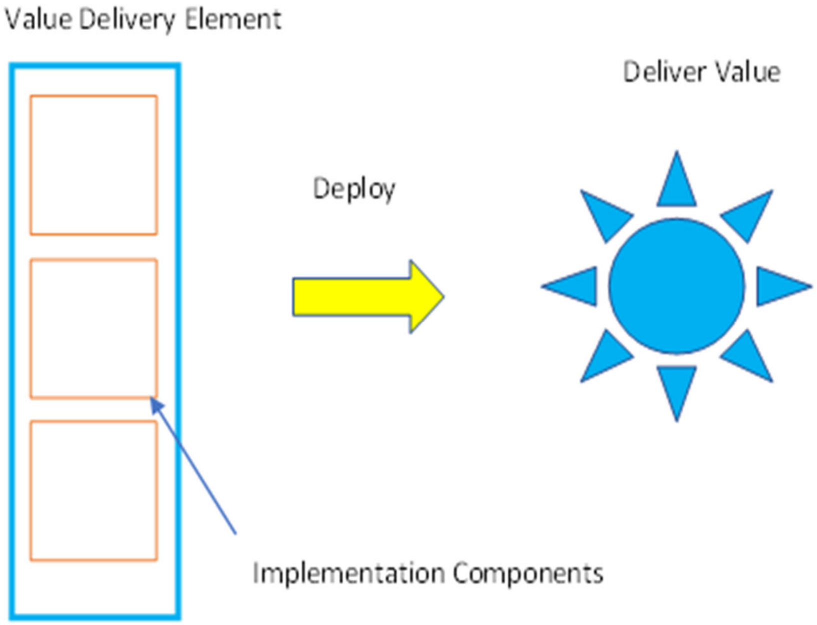
Delivering by details (e.g., stories, scenarios) can allow quicker feedback.

Focus on one element and one detail of that element.



Deploy

The value element is deployed to start delivering value


Value Delivery Element
Deliver Value
Deploy
Implementation Components