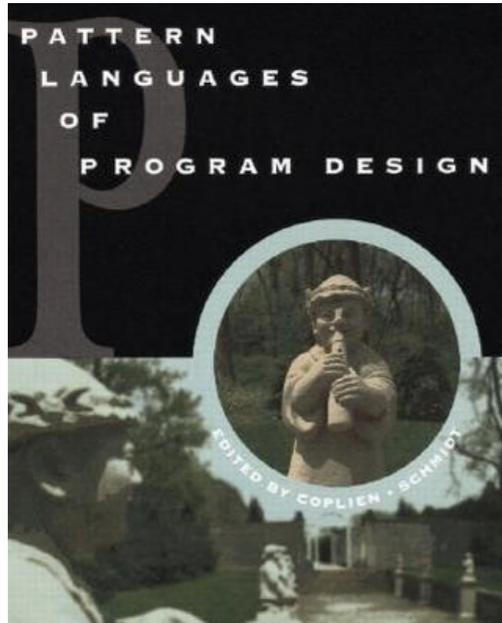


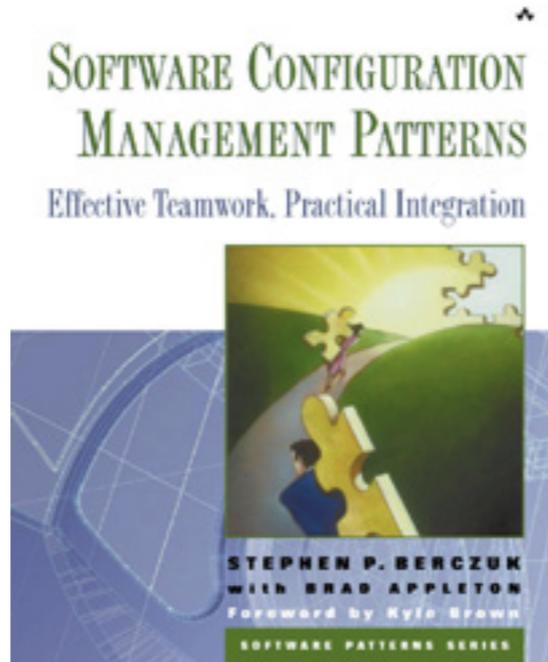
Using Testing Specialists to Grow Testing Skills in Agile Teams

Steve Berczuk
(TBA)

About Me



1995



2002

Software
Development

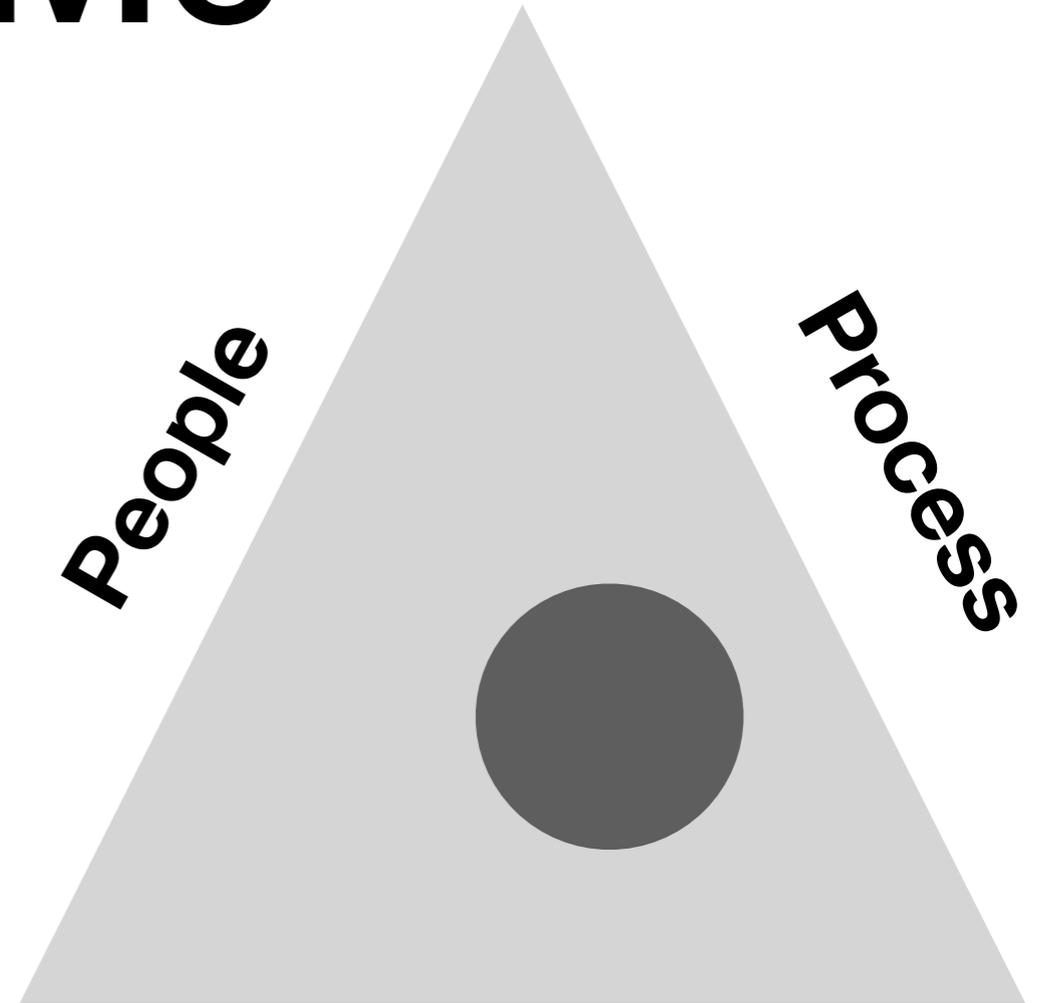
Writing



2005



????

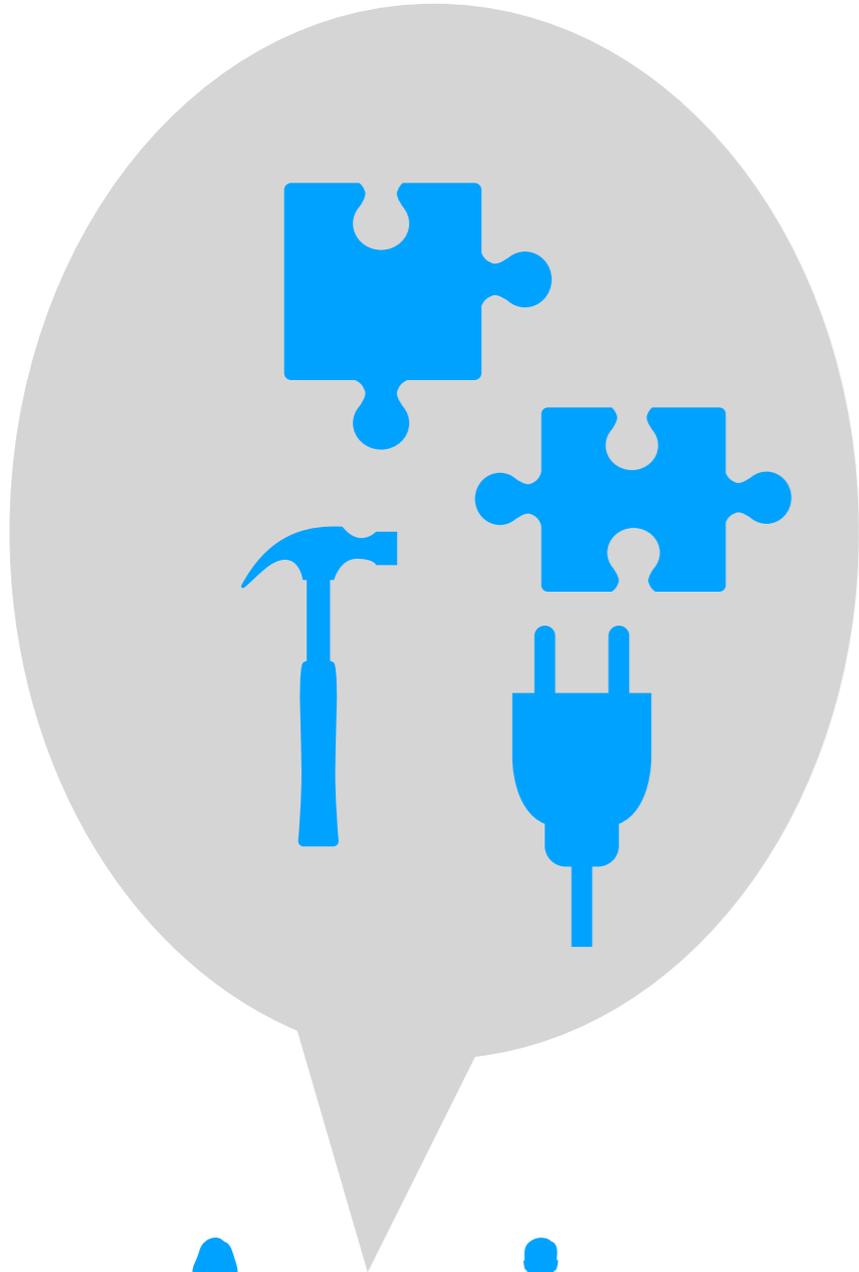


Technology

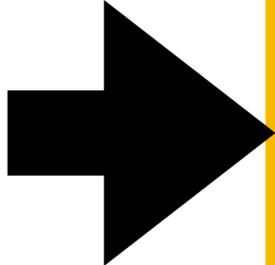
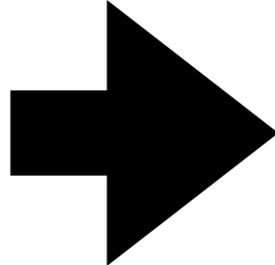
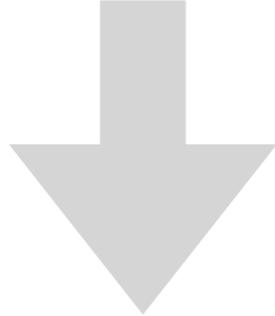
Agenda

- Scrum Process & Principles
- Scrum and Scrum Teams, and Testing
- Dealing with Growth and Specialization
- Questions/Discussion

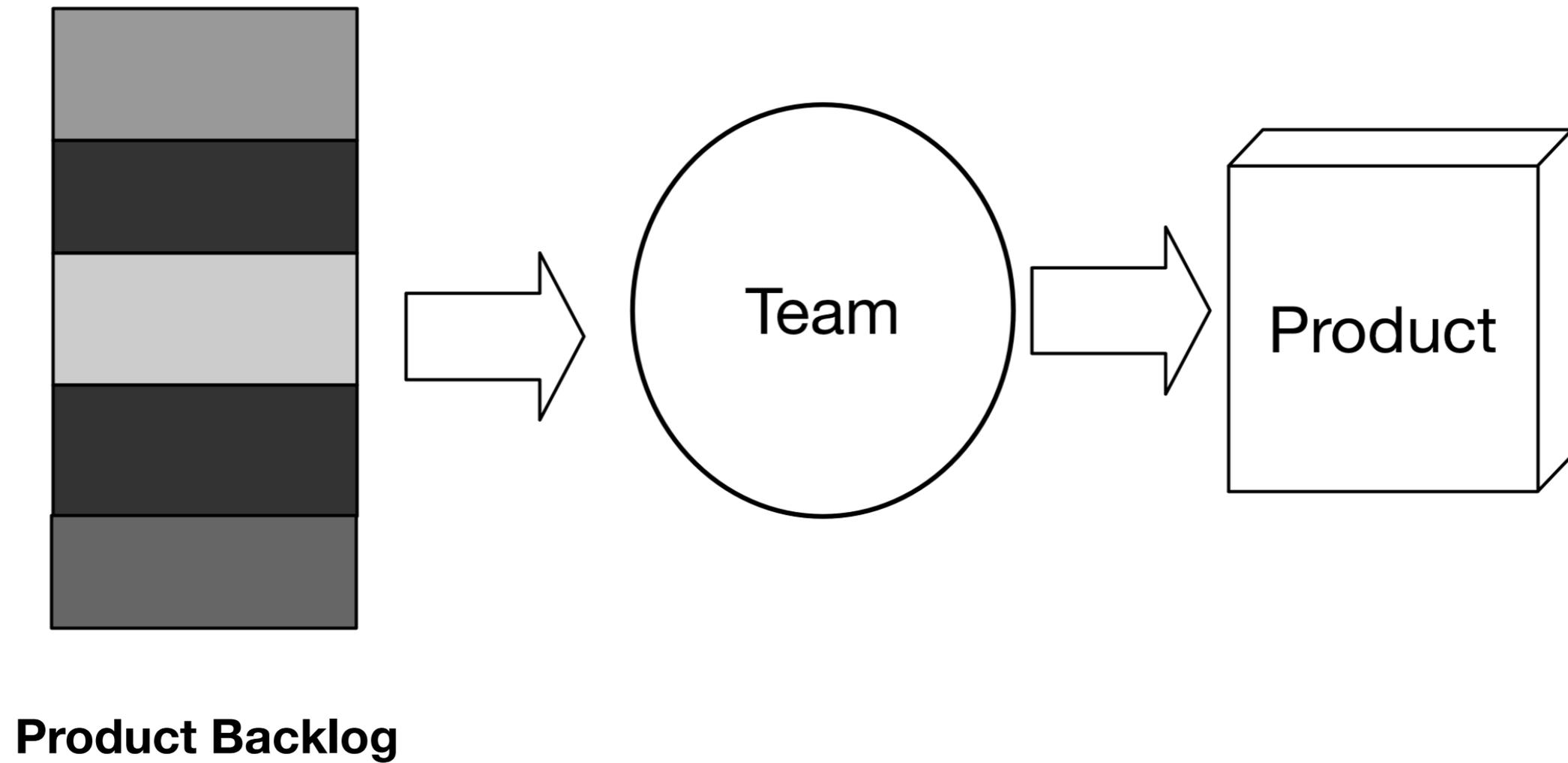
Scrum Process & Principles



UNCERTAINTY

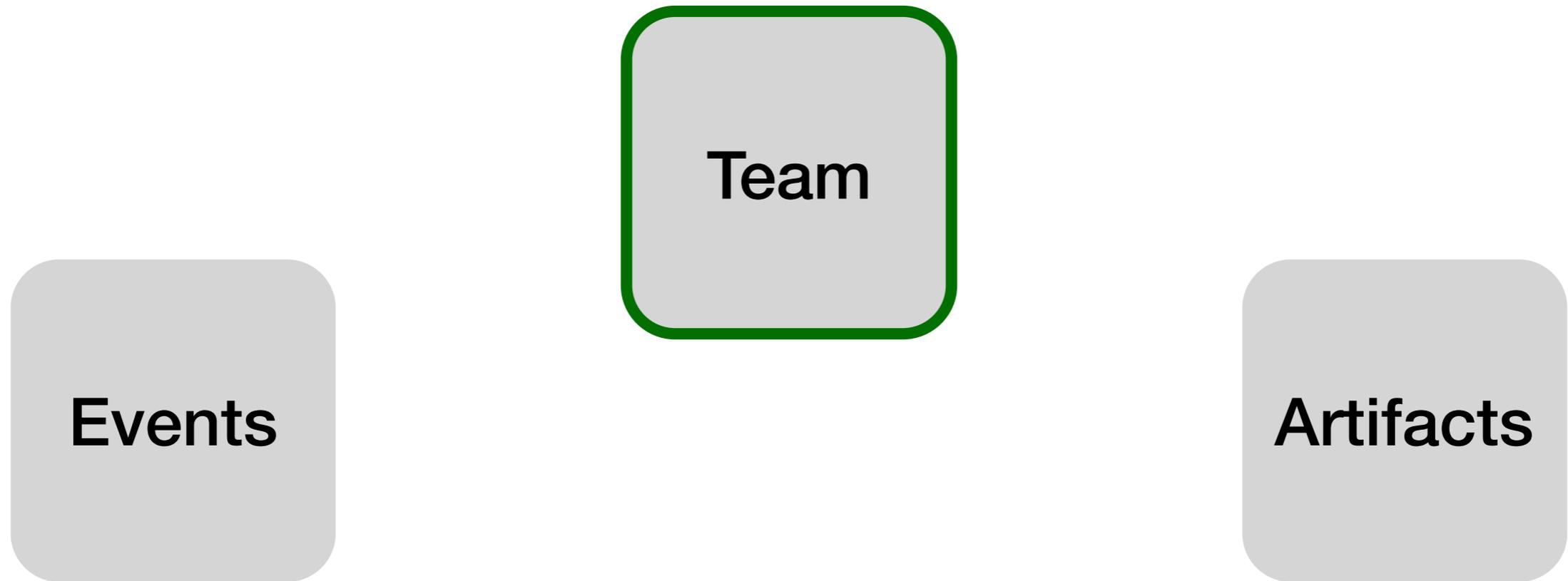


The Scrum Team



Product Backlog

Scrum



Scrum



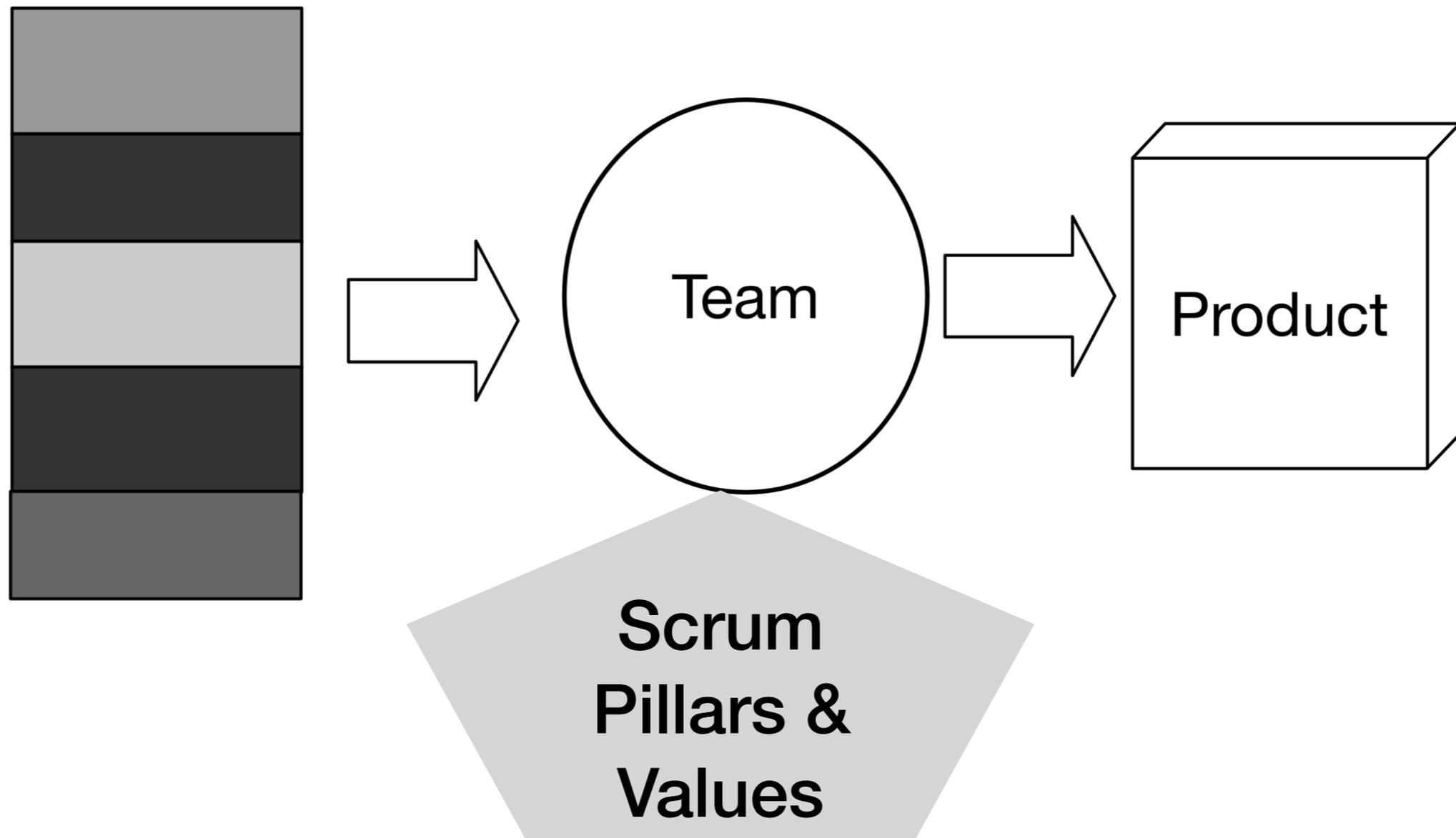
Events

Team

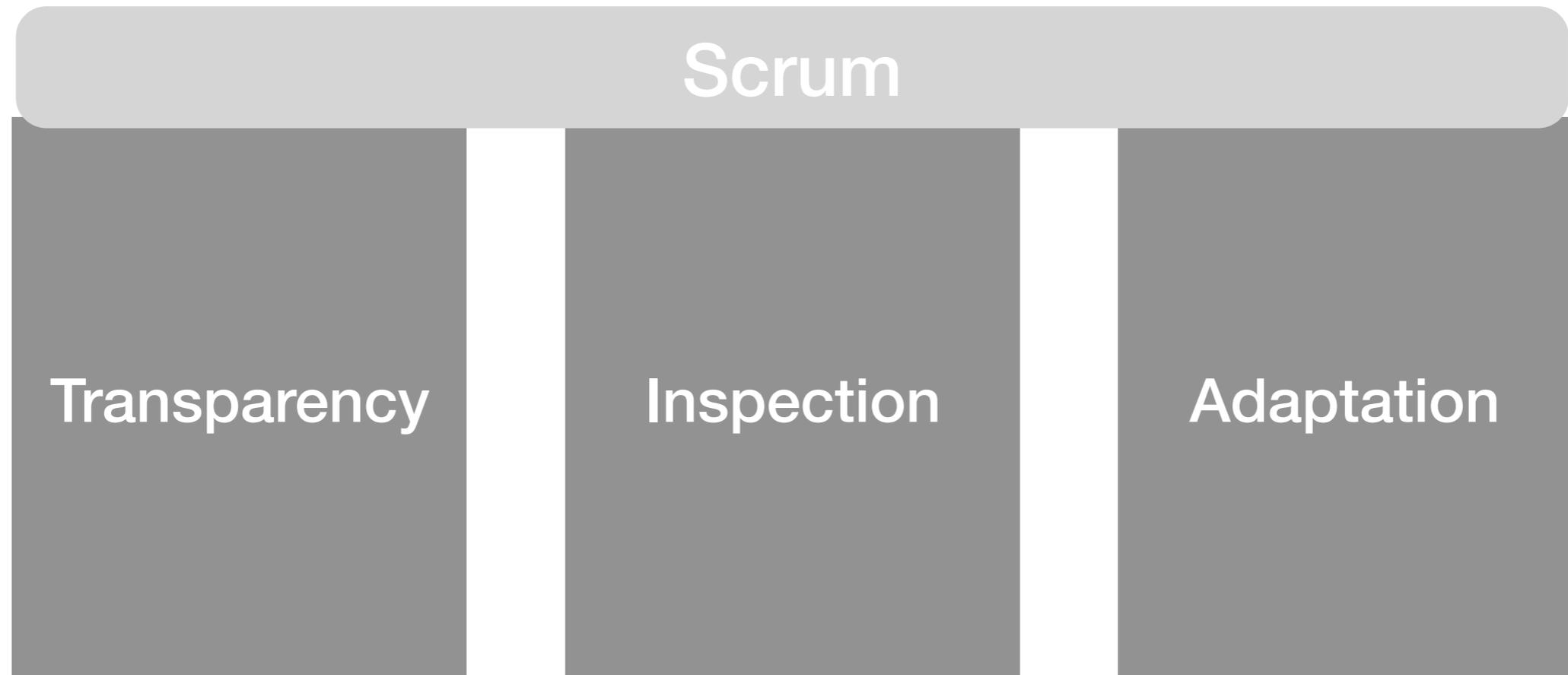
**Self Organizing
Self Contained
Accountable**

Artifacts

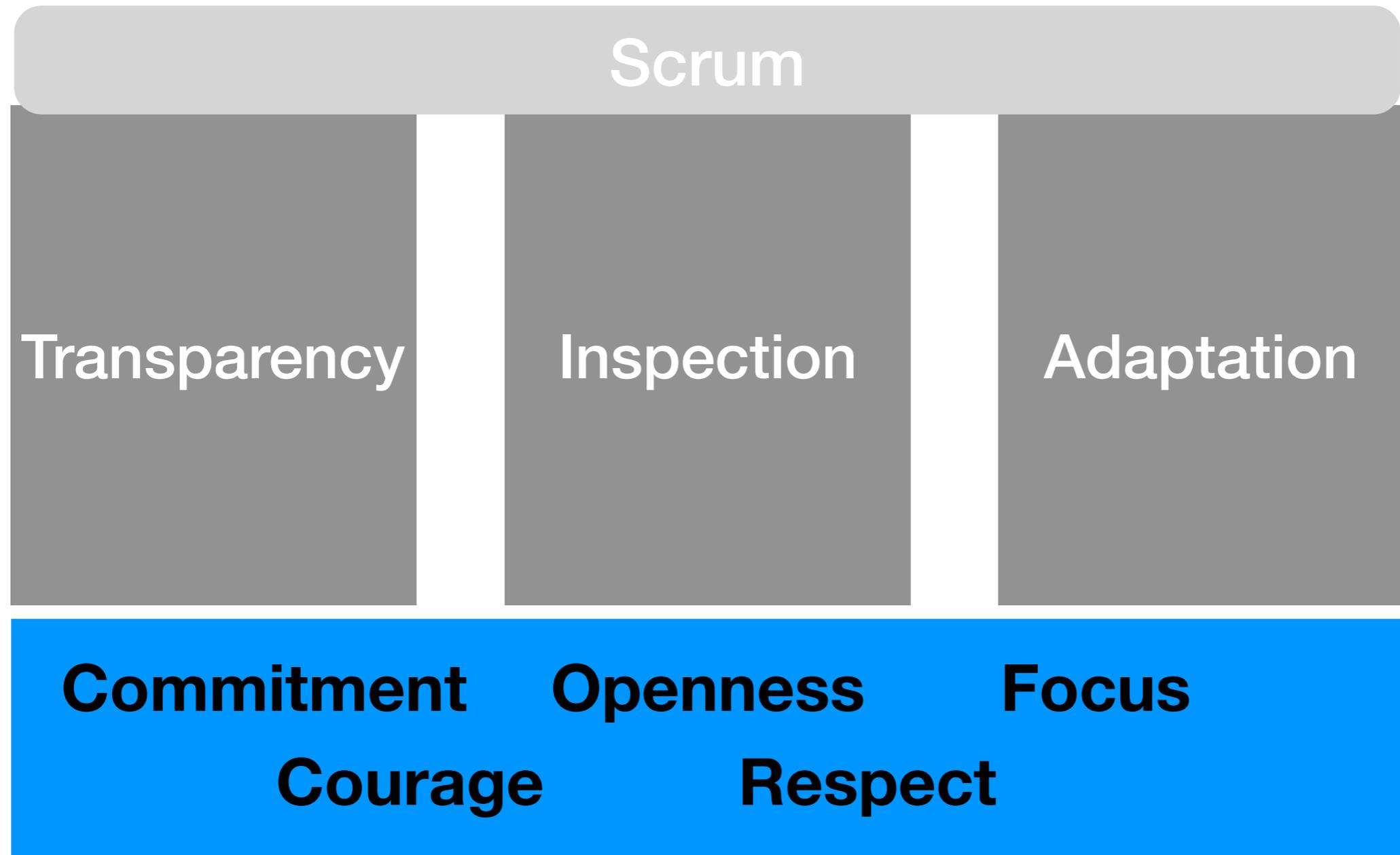
The Scrum Team



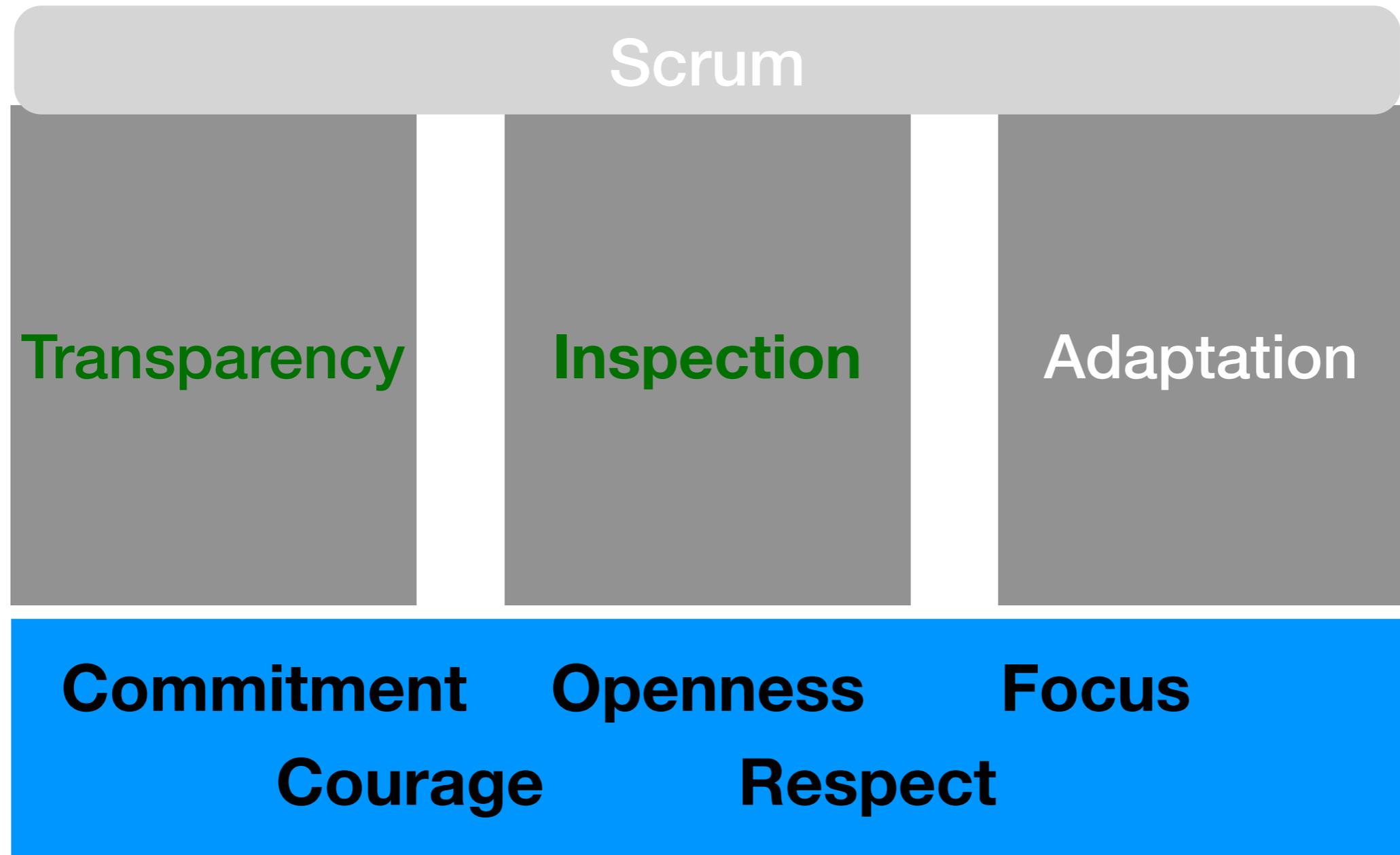
Scrum Pillars and Values



Scrum Pillars and Values



Scrum Pillars and Values



Scrum



Events

Team

Self Organizing
Self Contained
Accountable

Artifacts

Scrum

Team

Self Organizing

Self Contained

Accountable

Cross-
Functional

-Dev

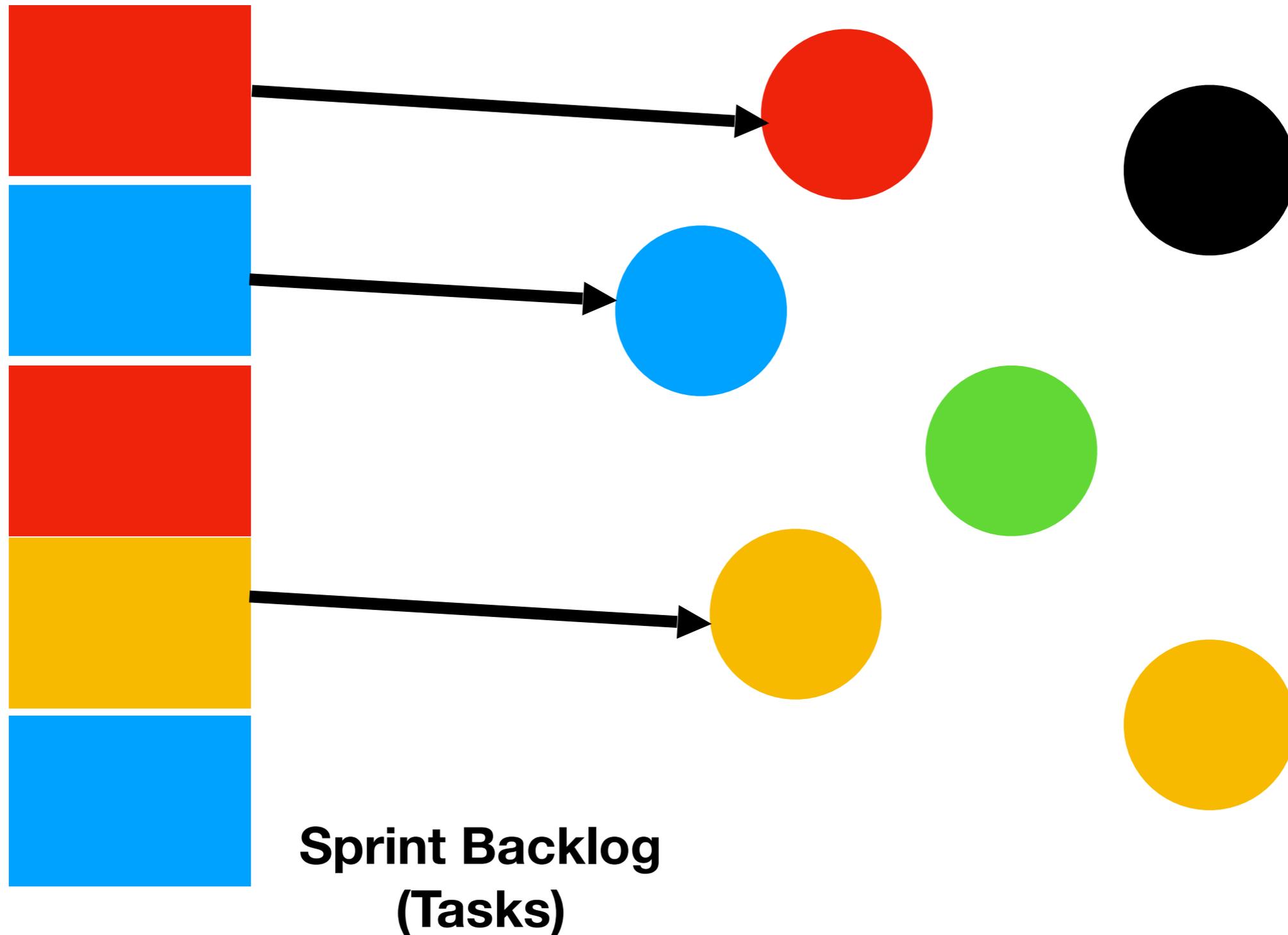
-Testing

-QA

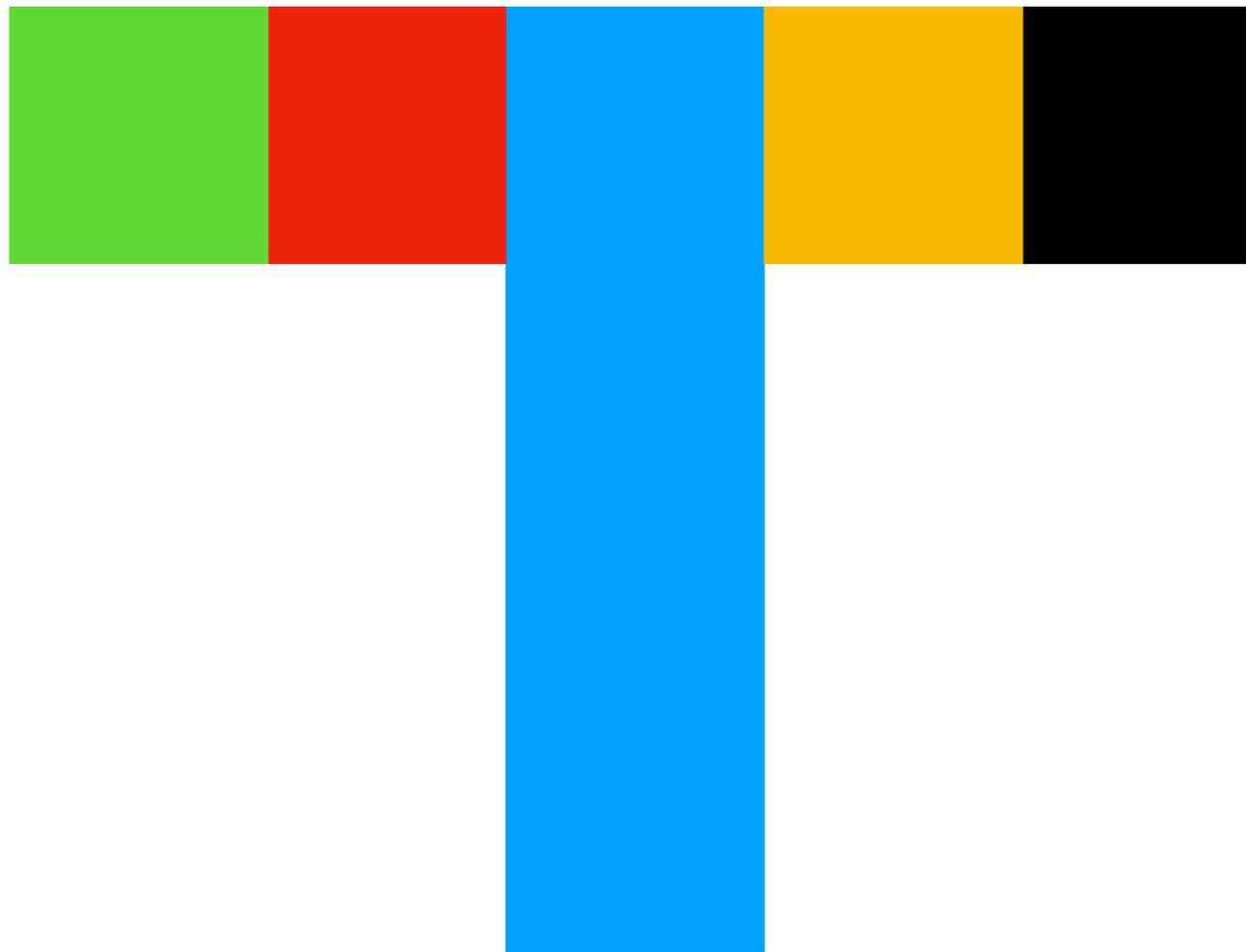
-Deployment

Why Cross Functional?

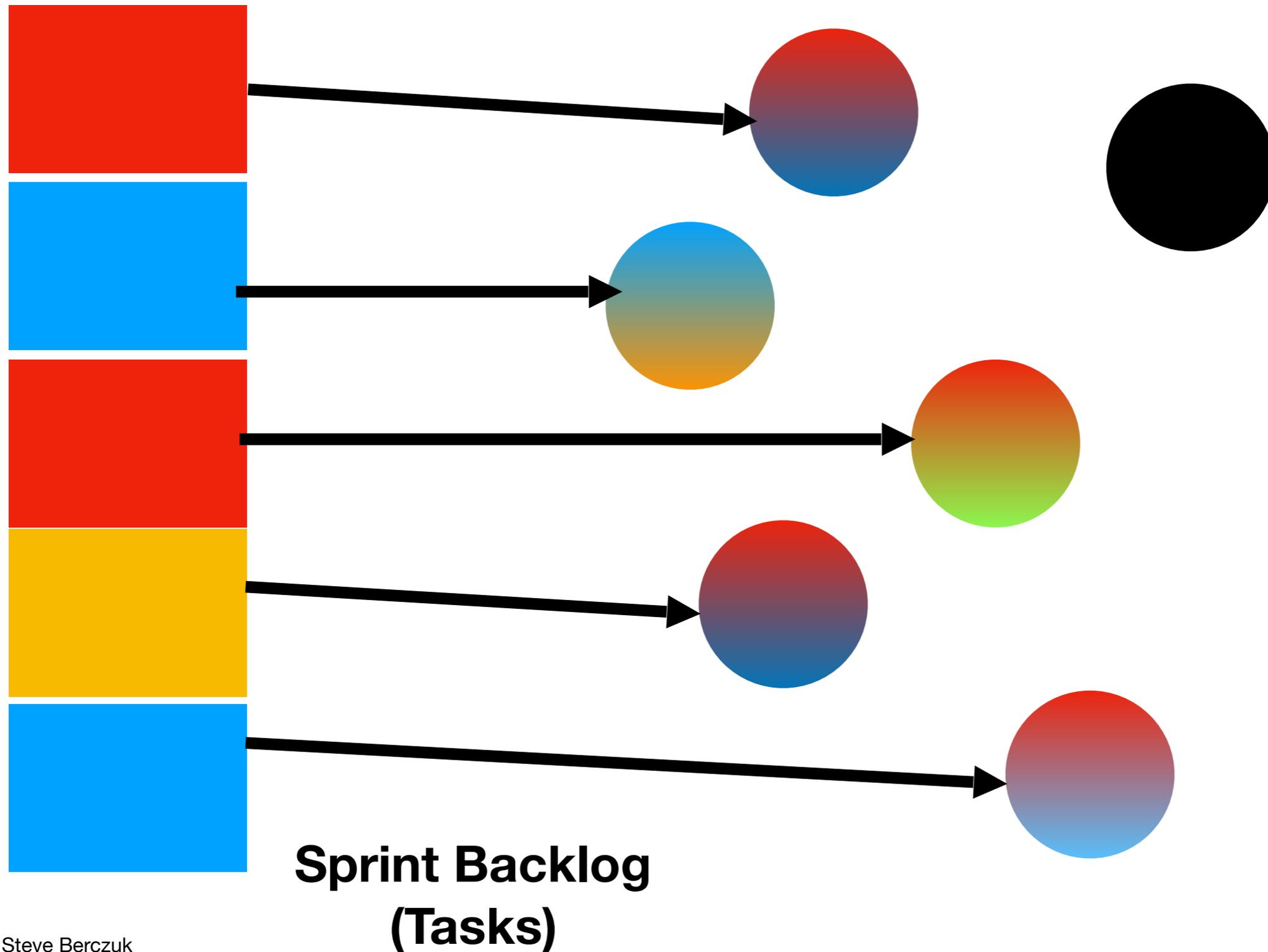
Experts Only: Blocking



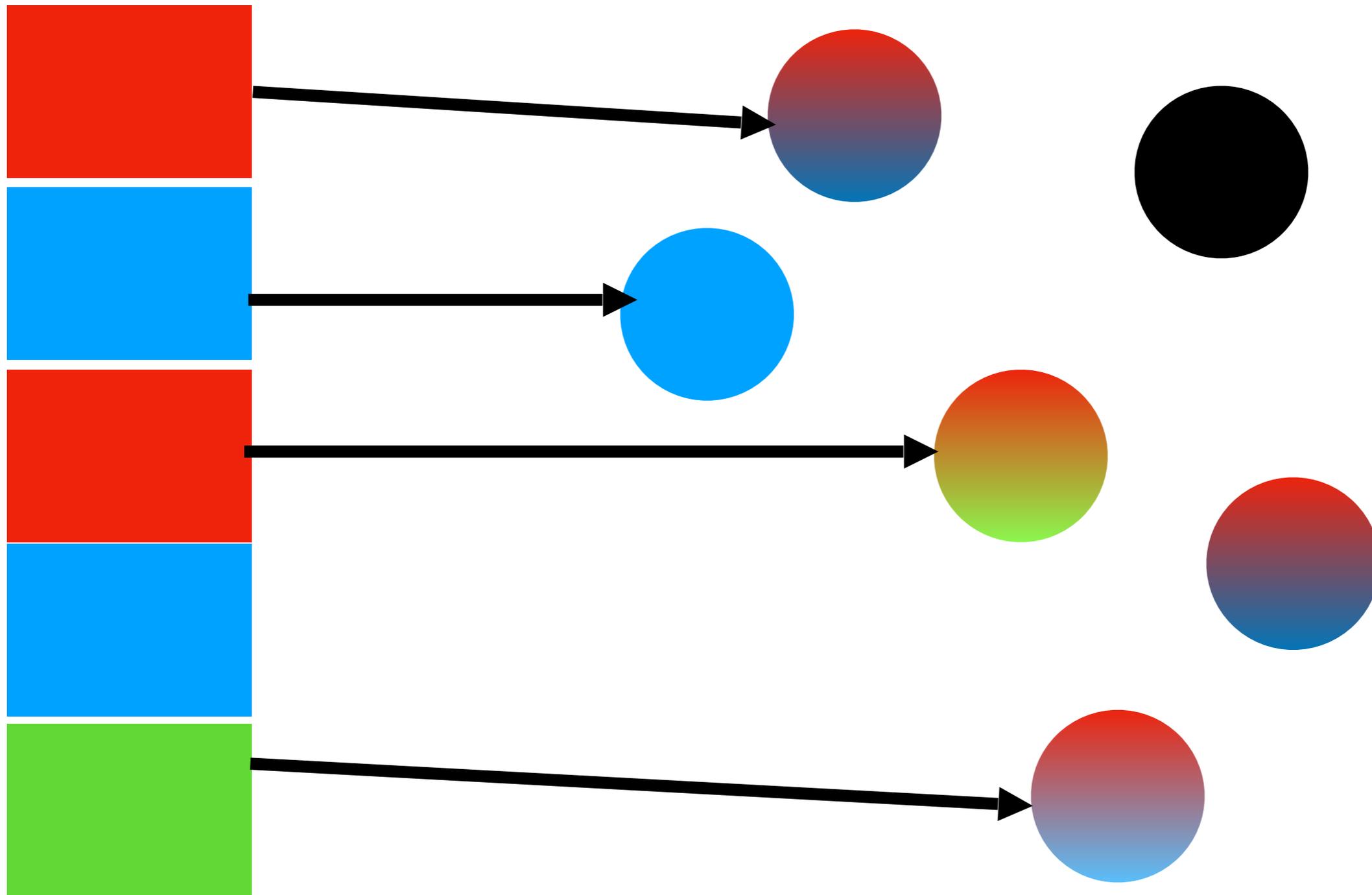
T Shaped Skills



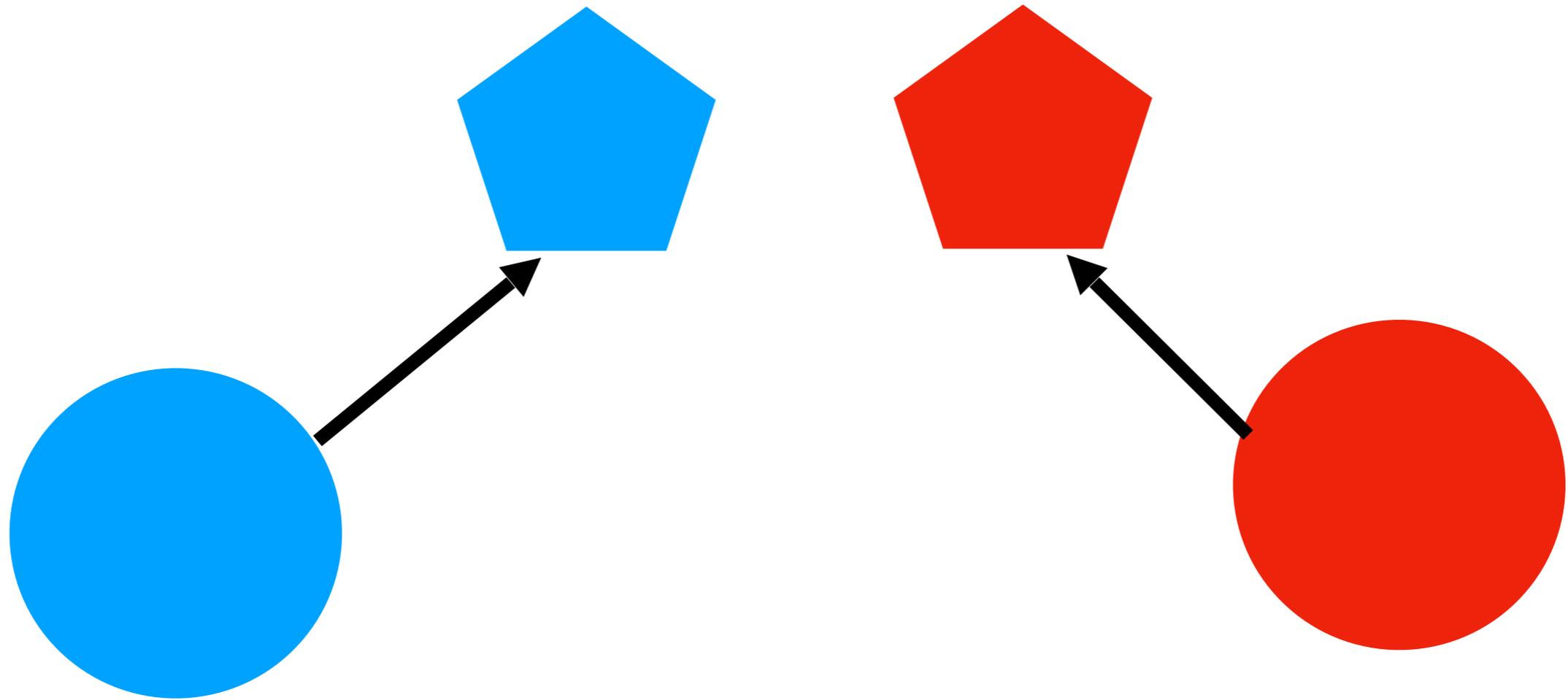
T-Shaped Skills



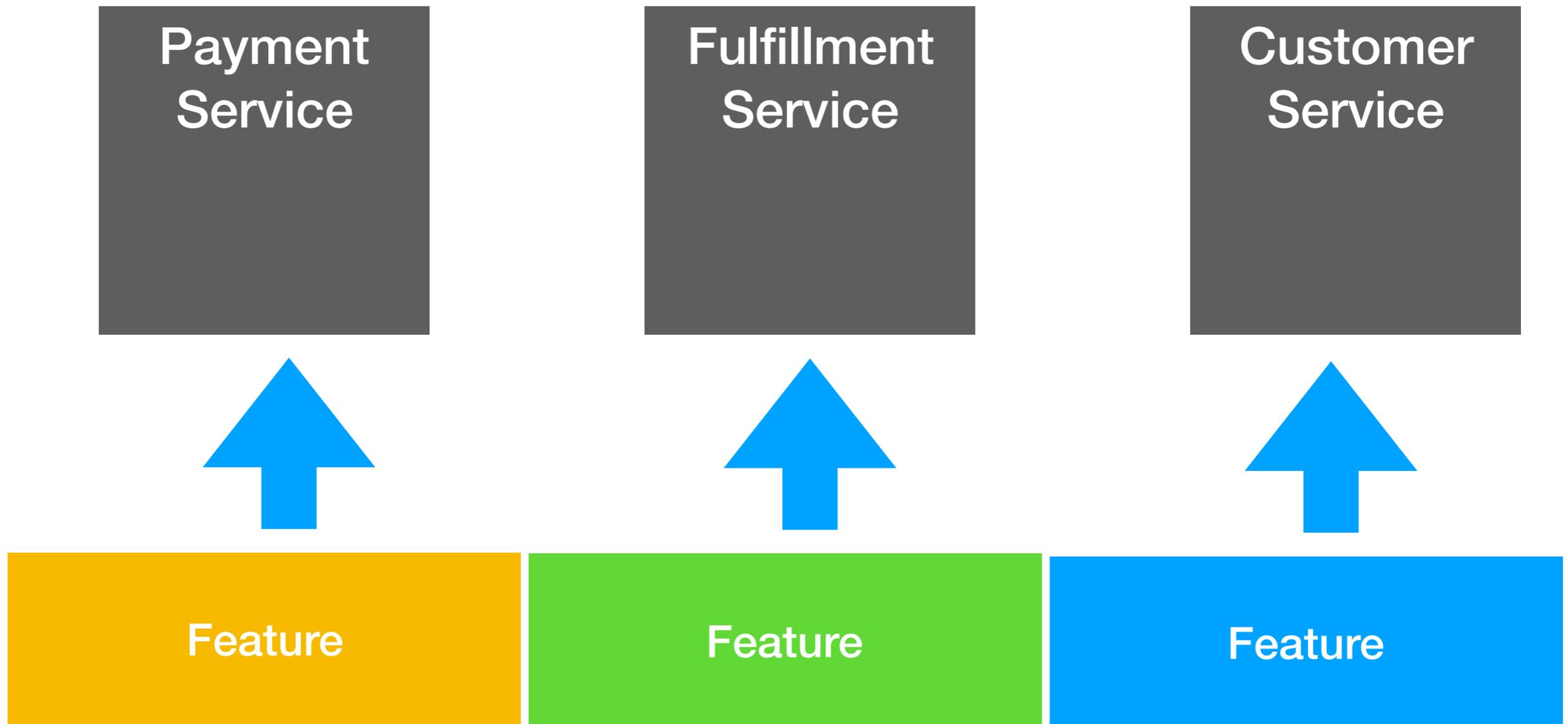
Reality



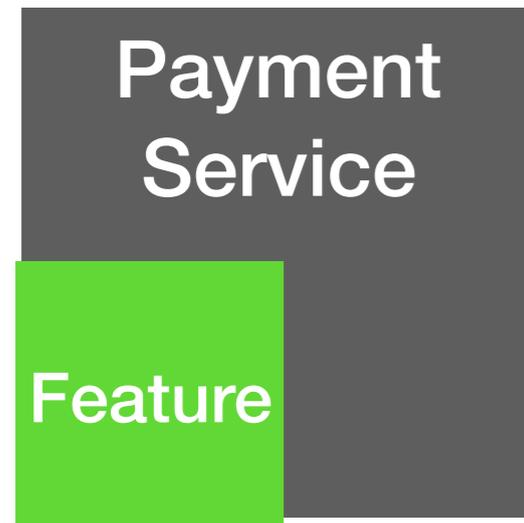
Teams and Features



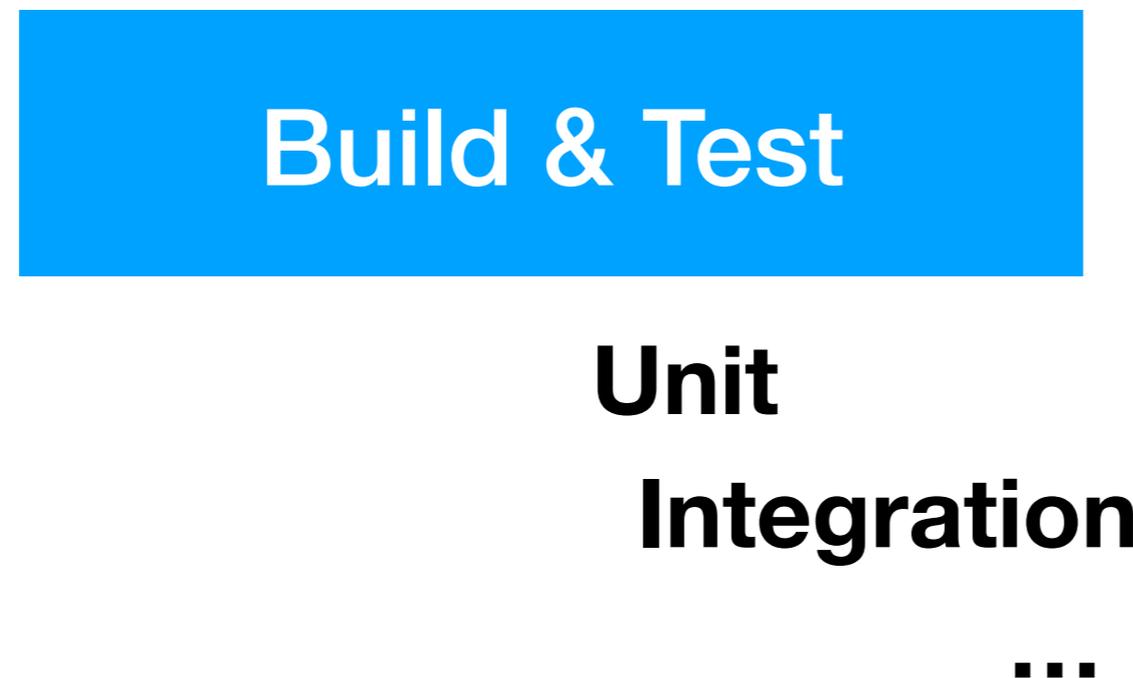
Aside: Feature v Application



Aside: Feature v Application



Testing in the Lifecycle



Automated Integration Testing is Different

DEVOPS

QUALITY

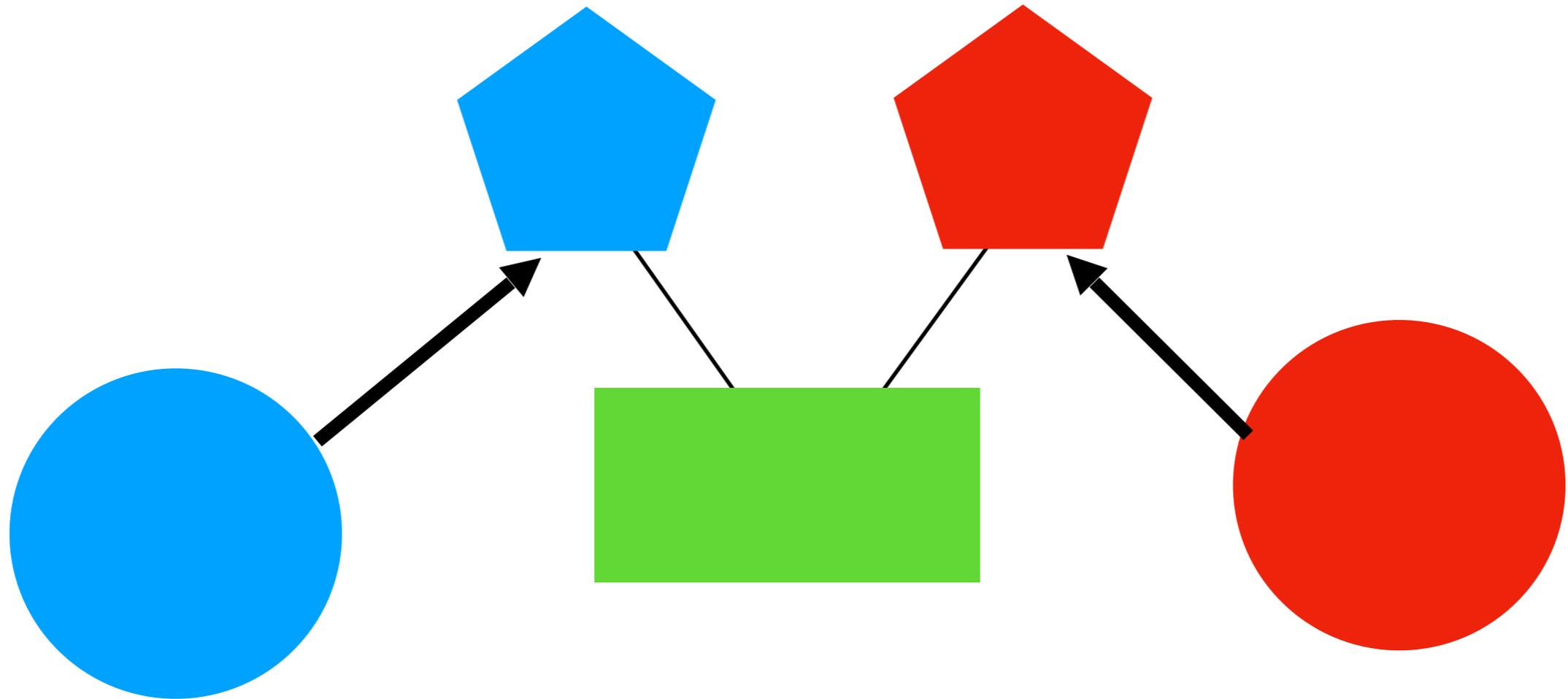
Automated Integration Testing is Different

Coding

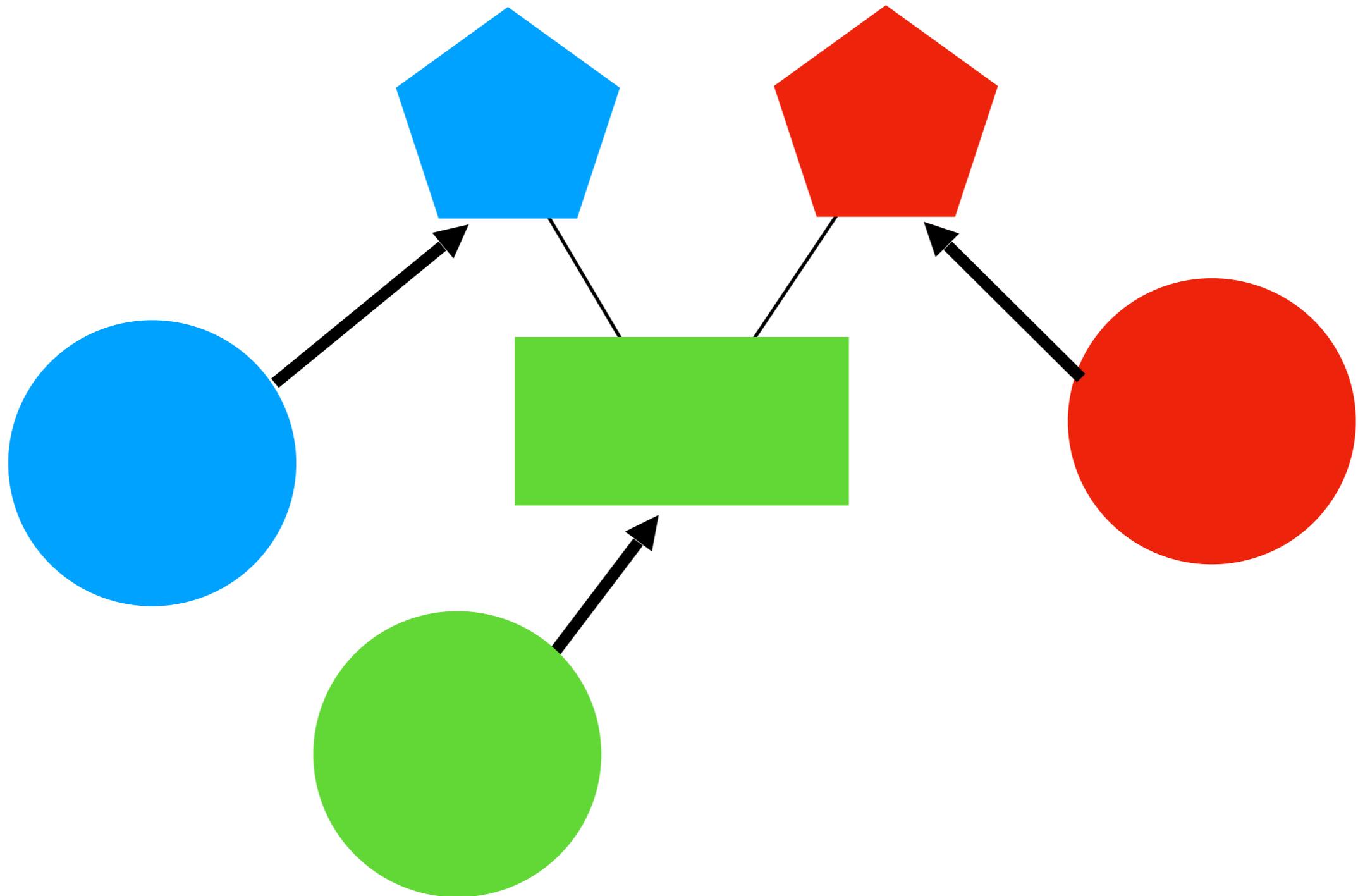
....

Scaling Development

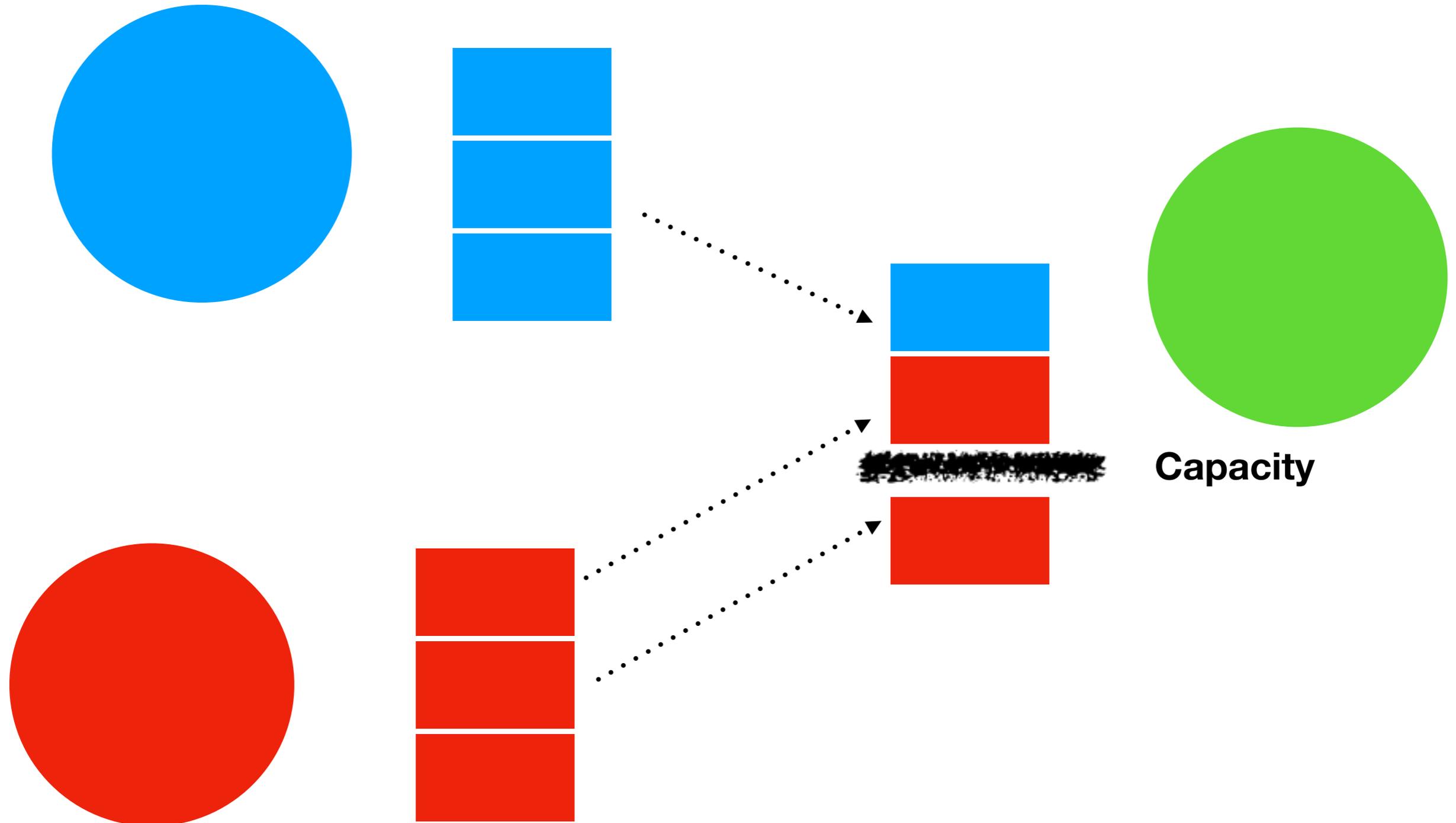
Teams and Components



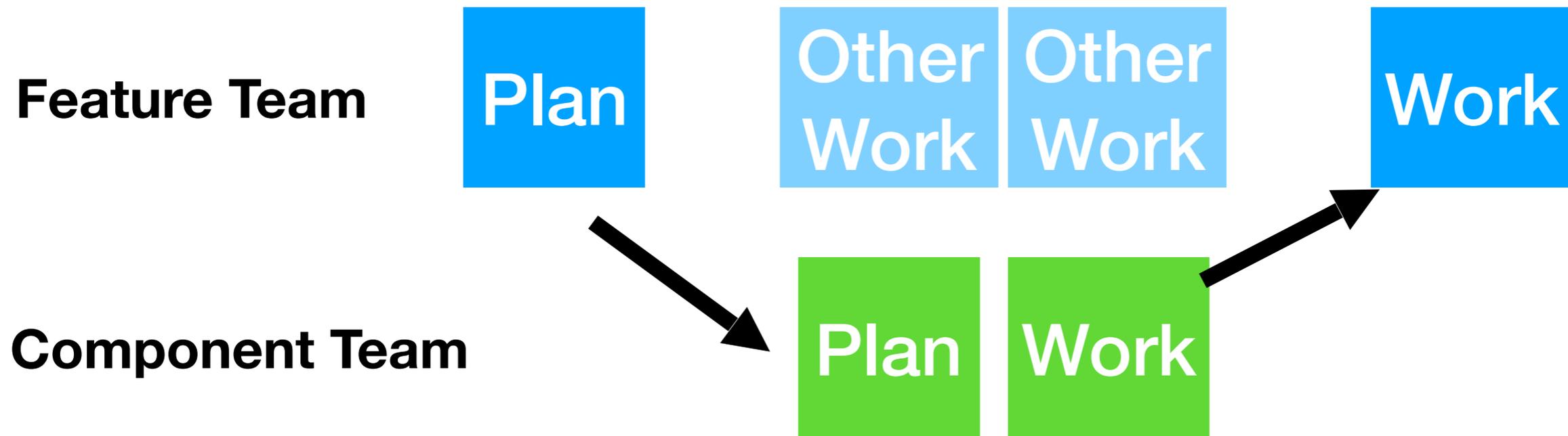
Components & Features



Component Team Backlogs

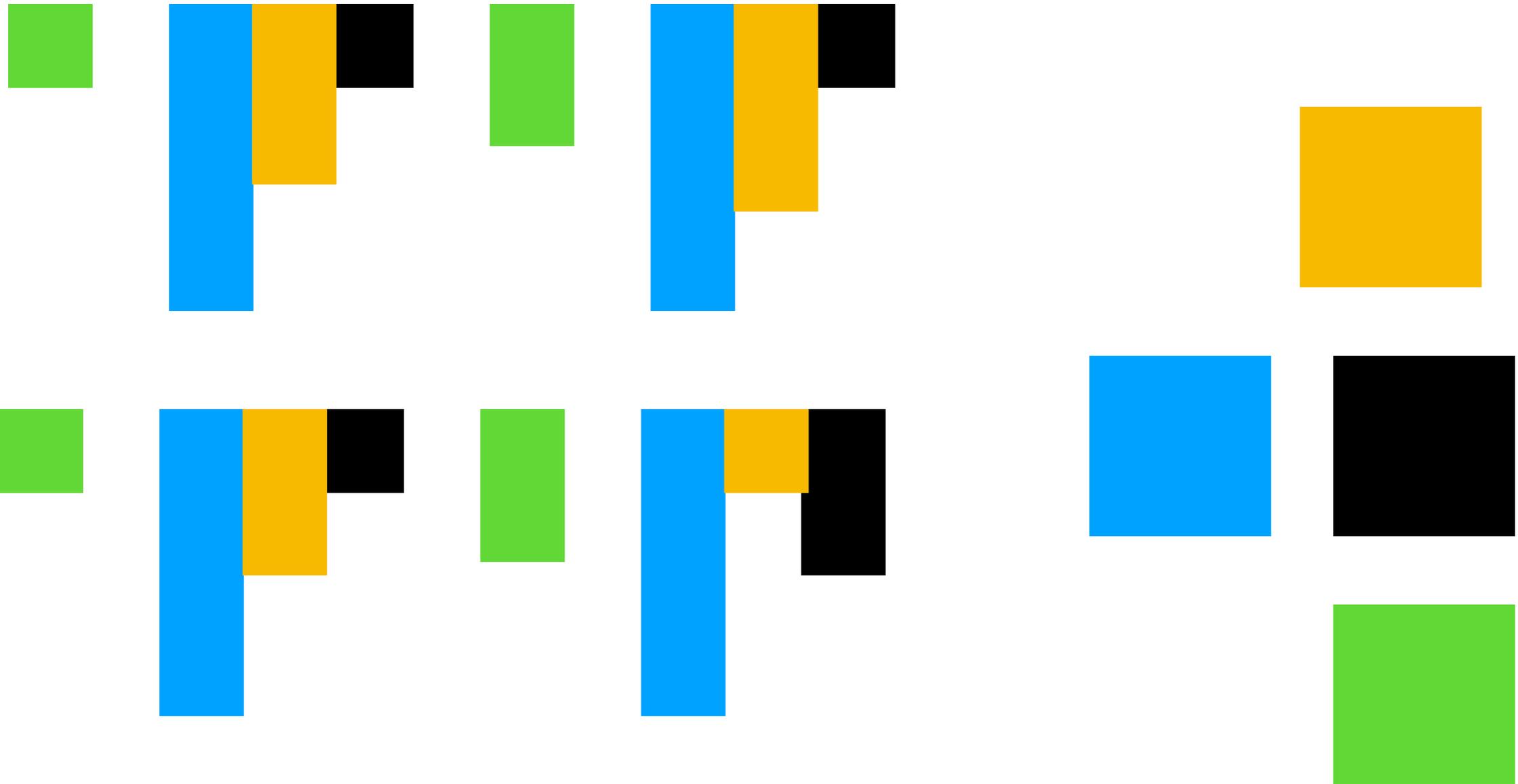


Dependencies in Time

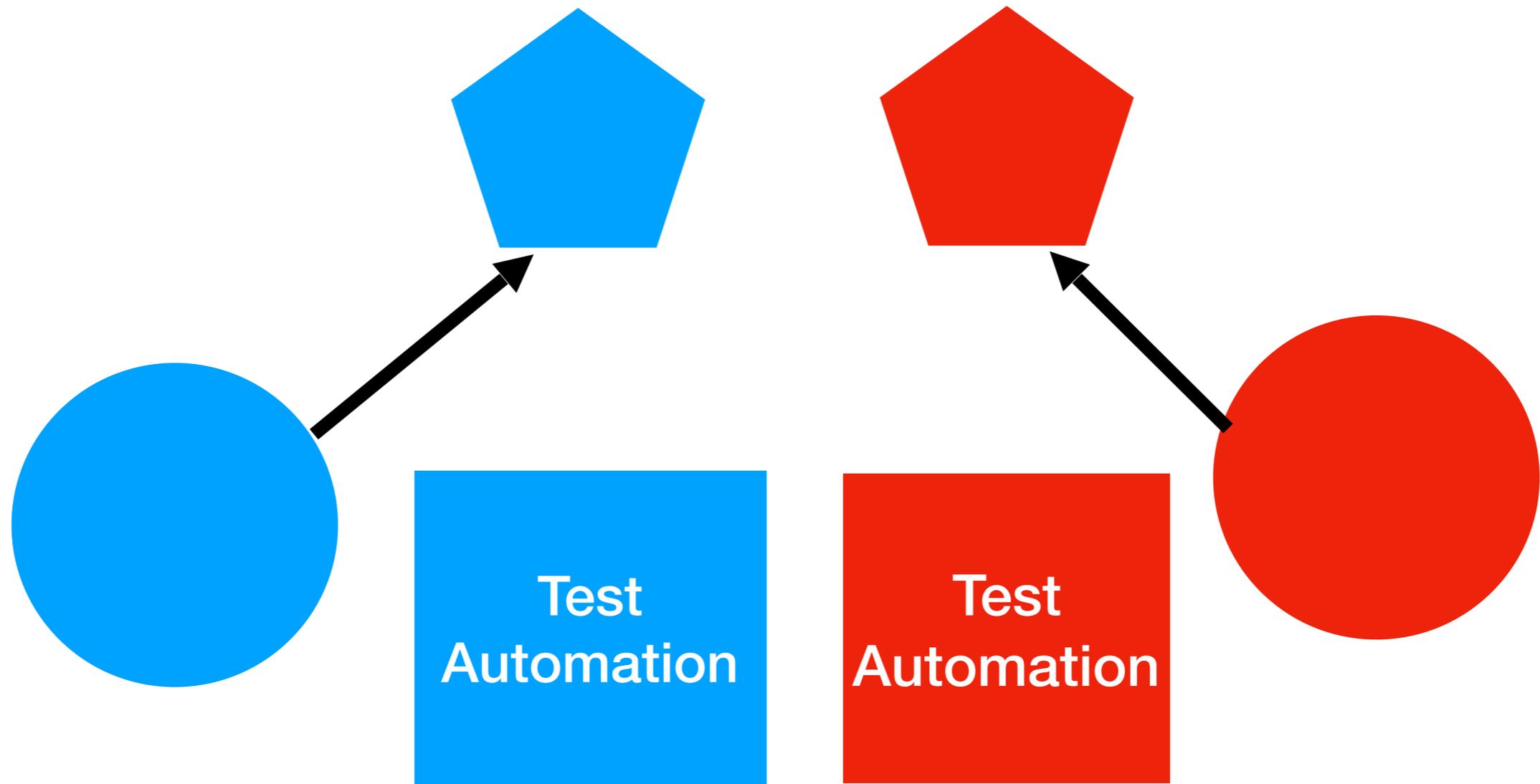


**Single + Self Contained
Teams are Better**

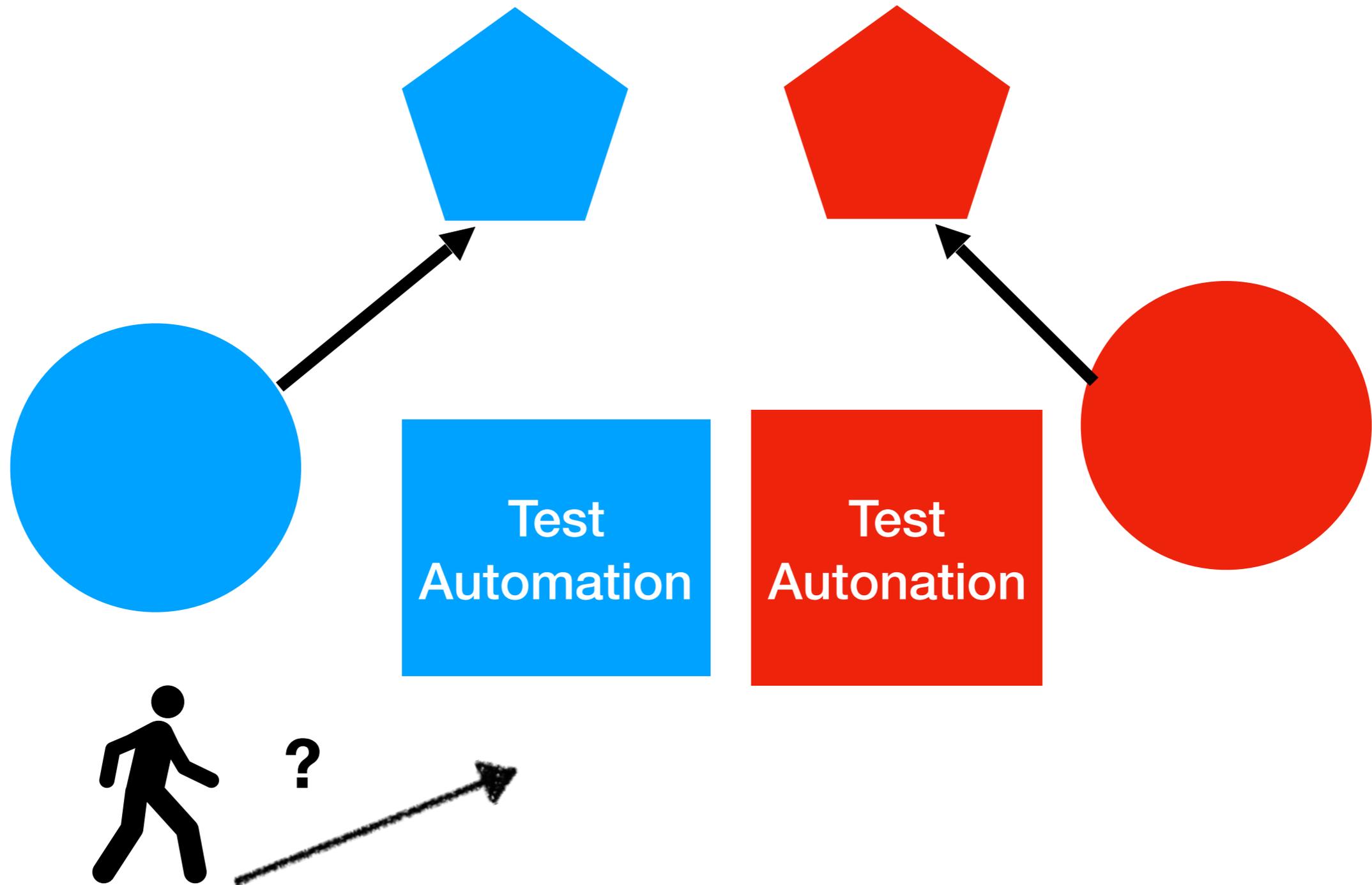
Missing Skills



Teams and Specialties

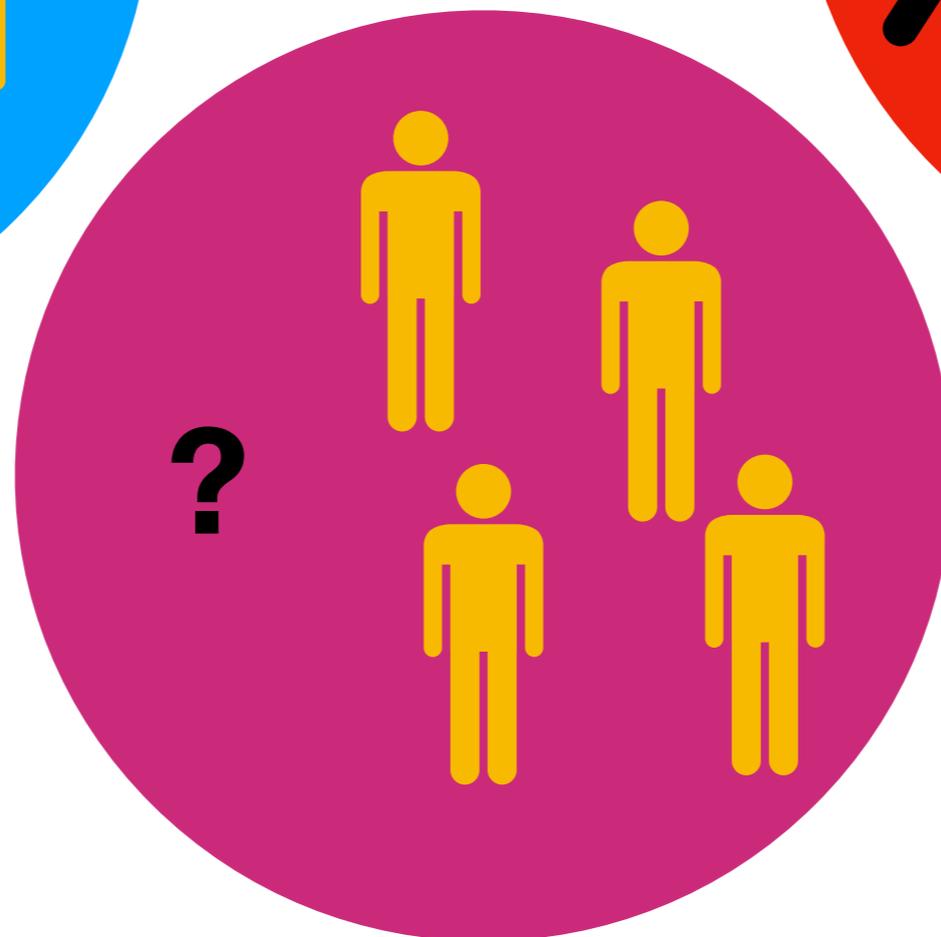
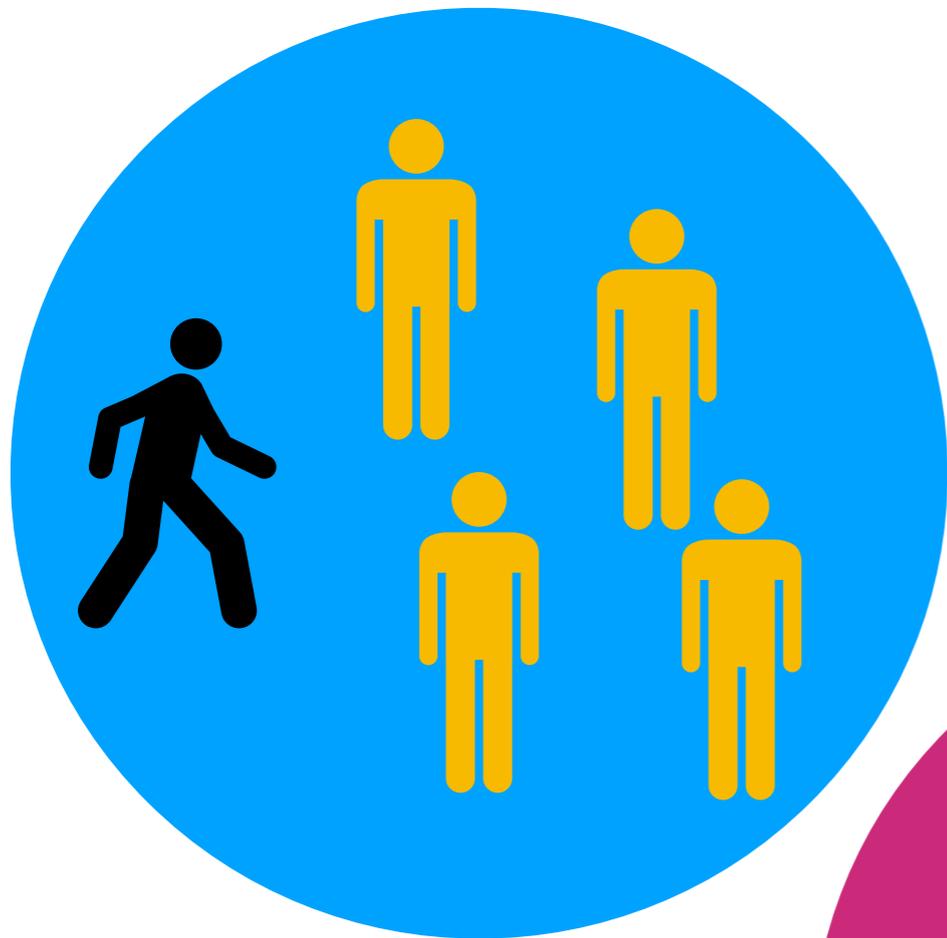


Teams and Specialties

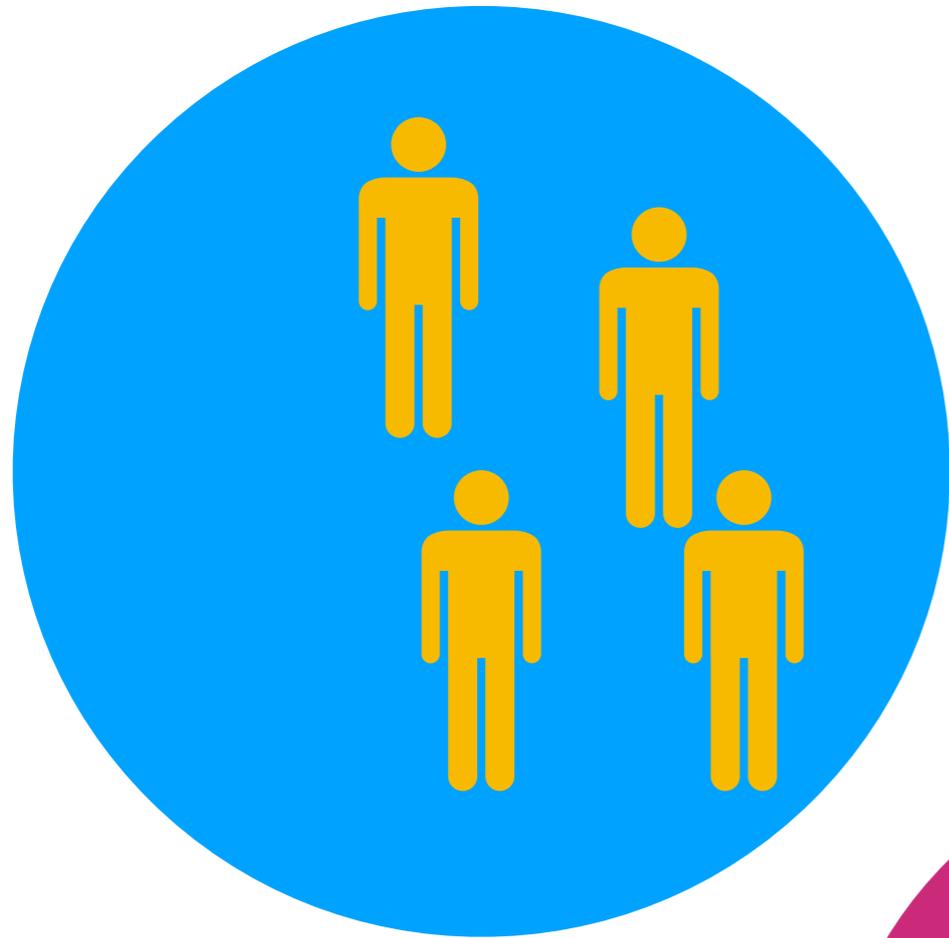


Specialists: Options

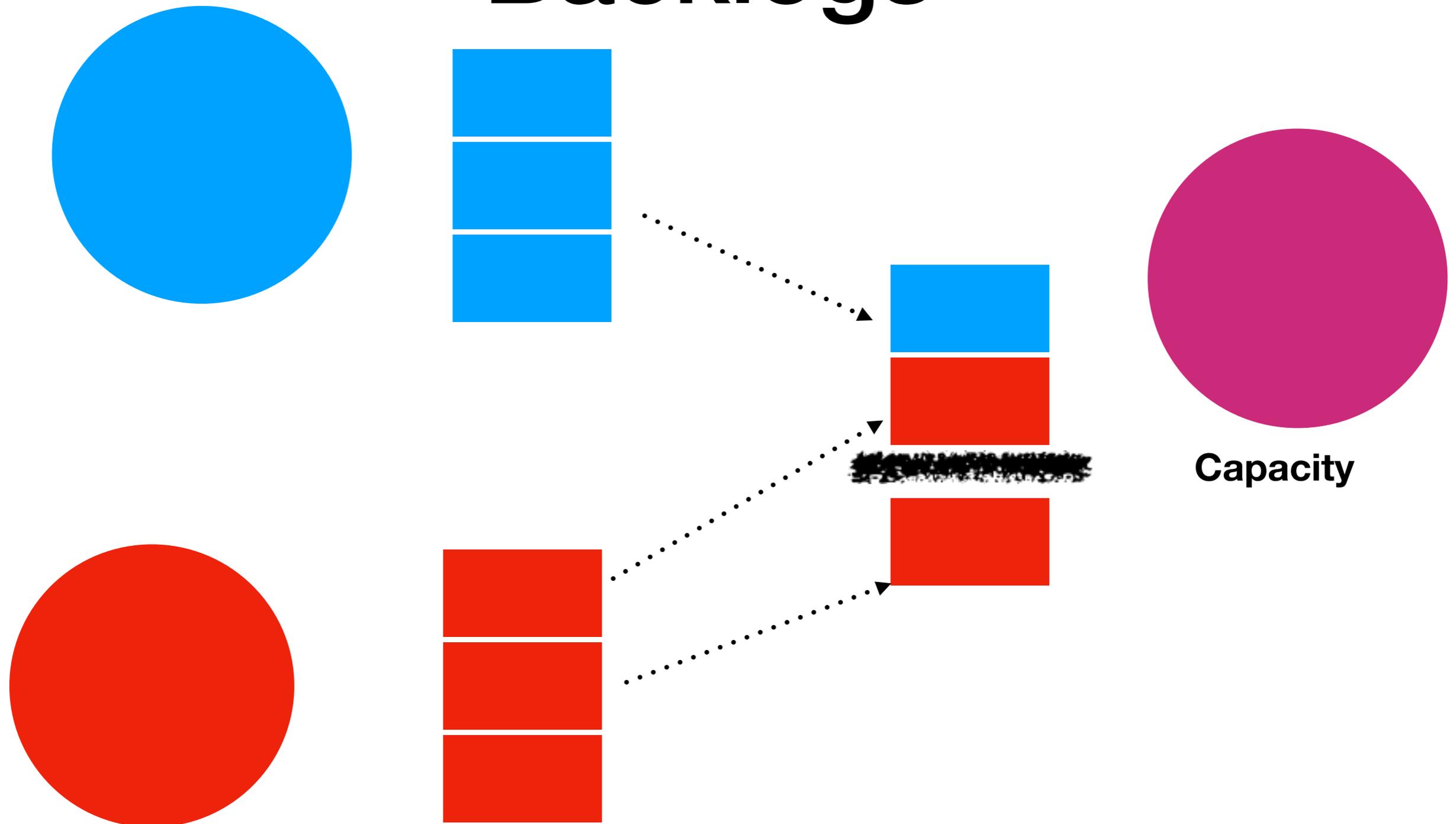
Embedded Specialist



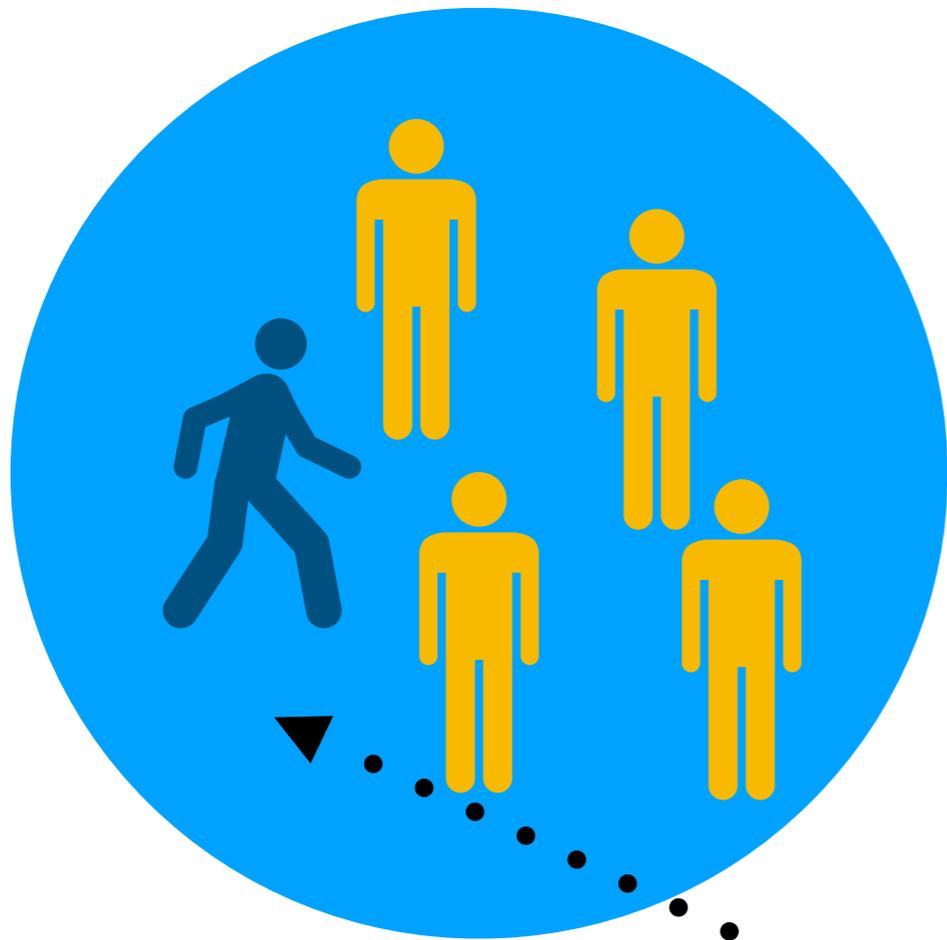
Specialist Component Team



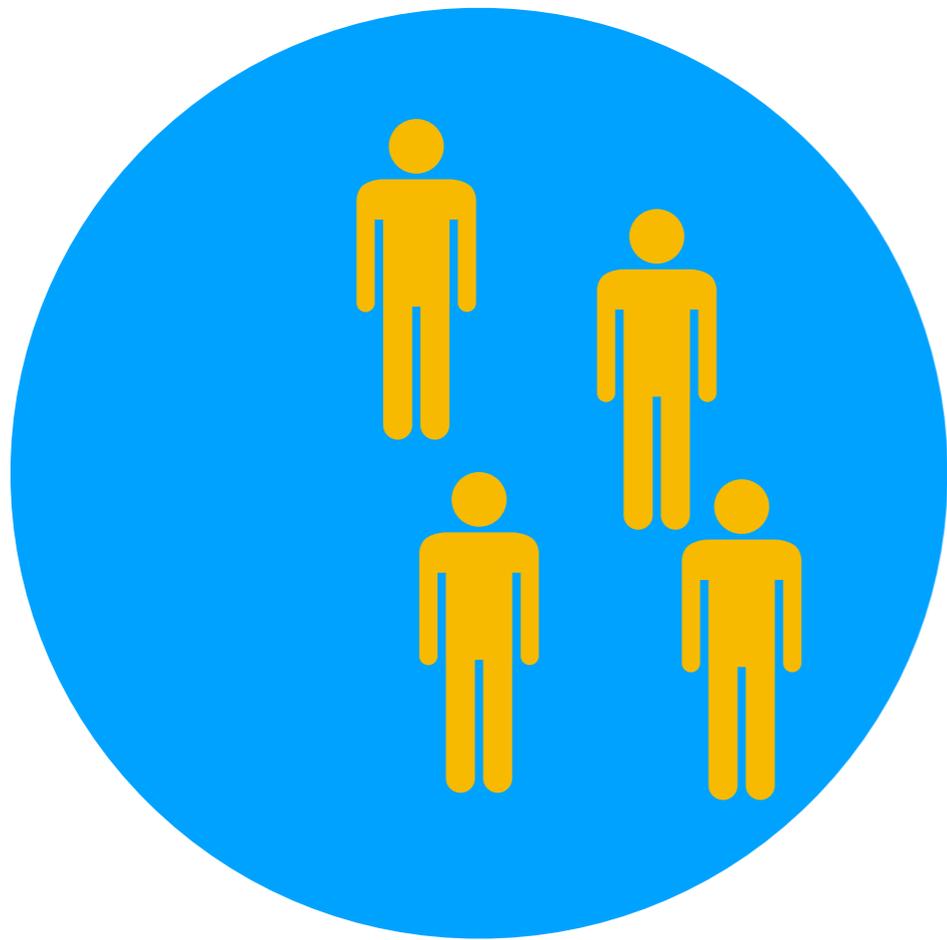
Specialist Component Team Backlogs



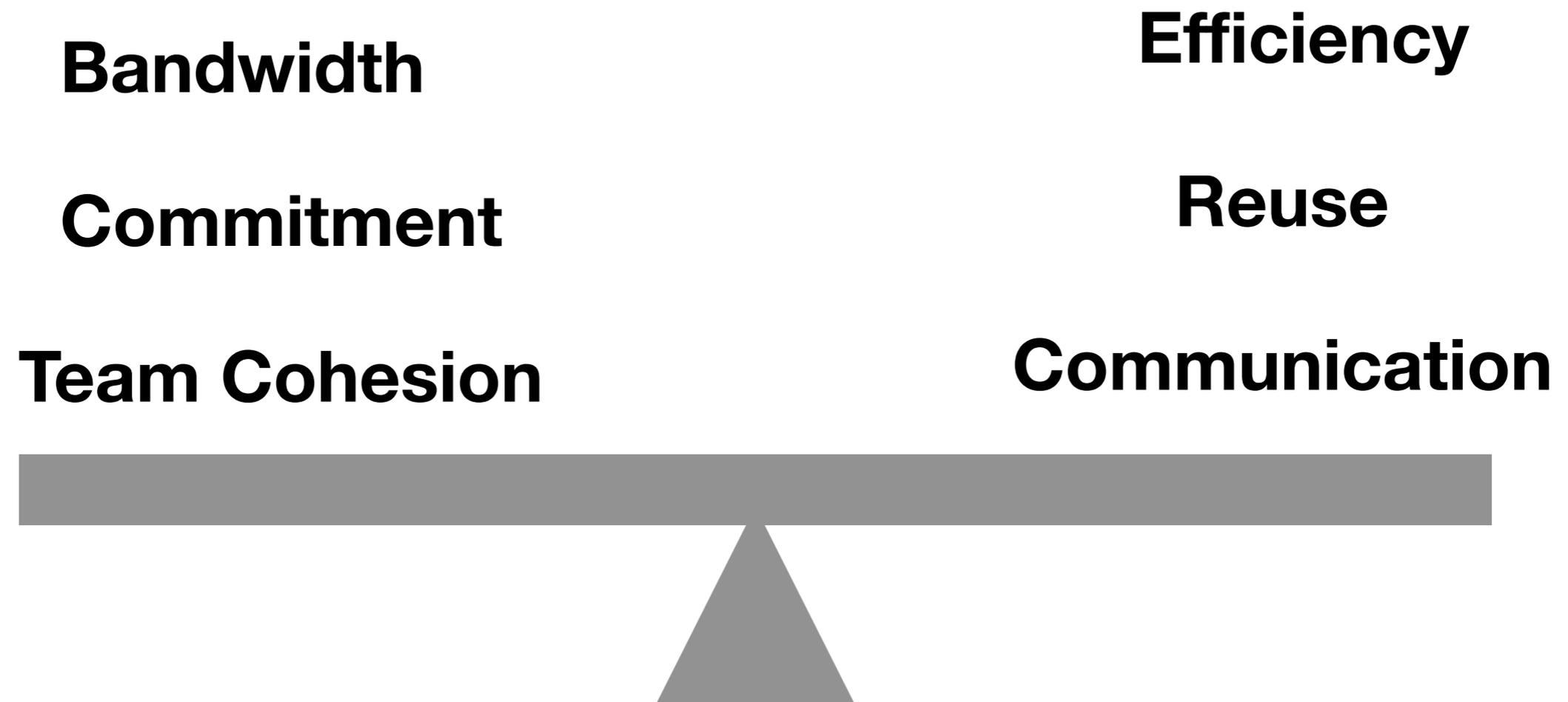
Consulting Model



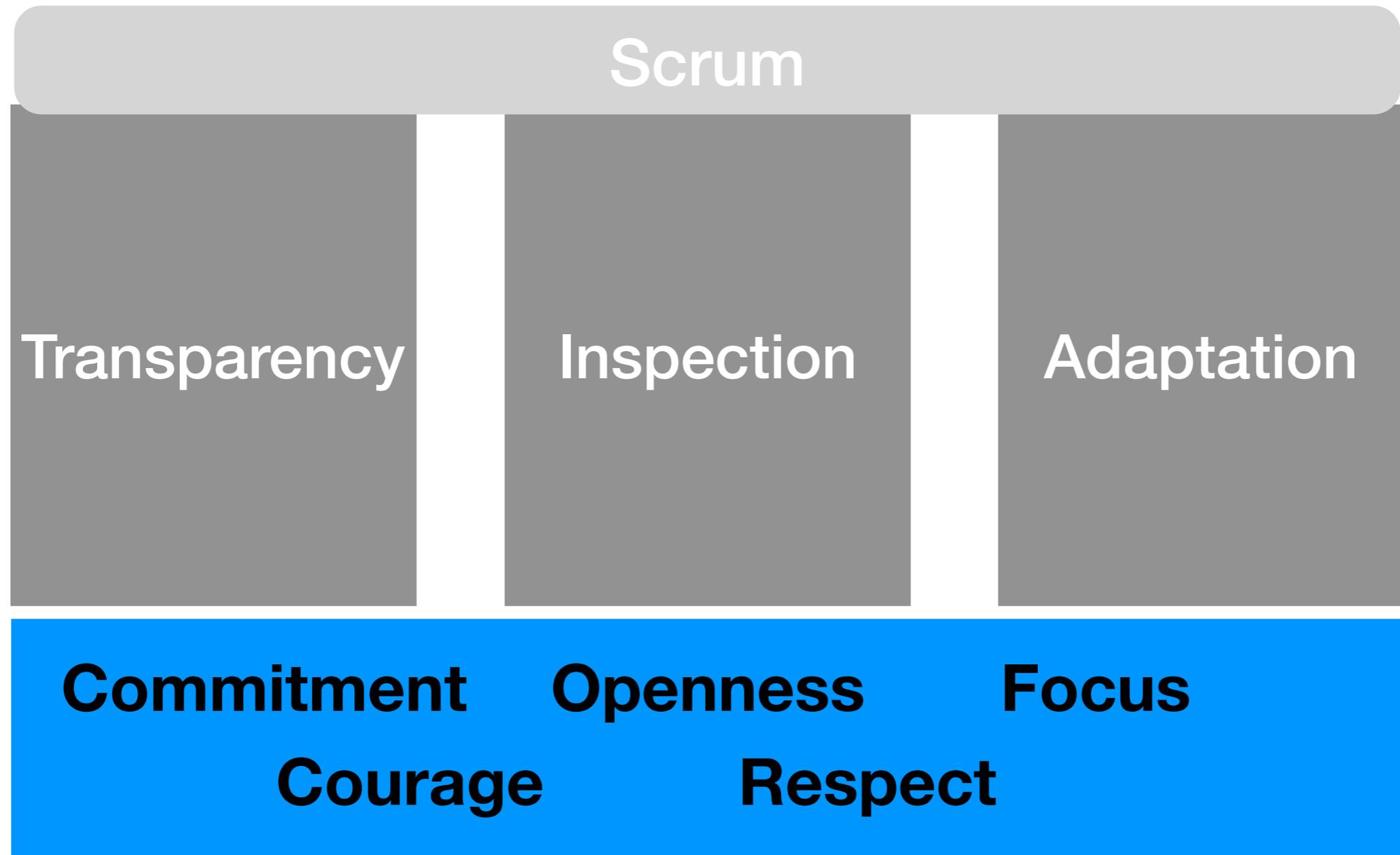
Consulting Model



Balance

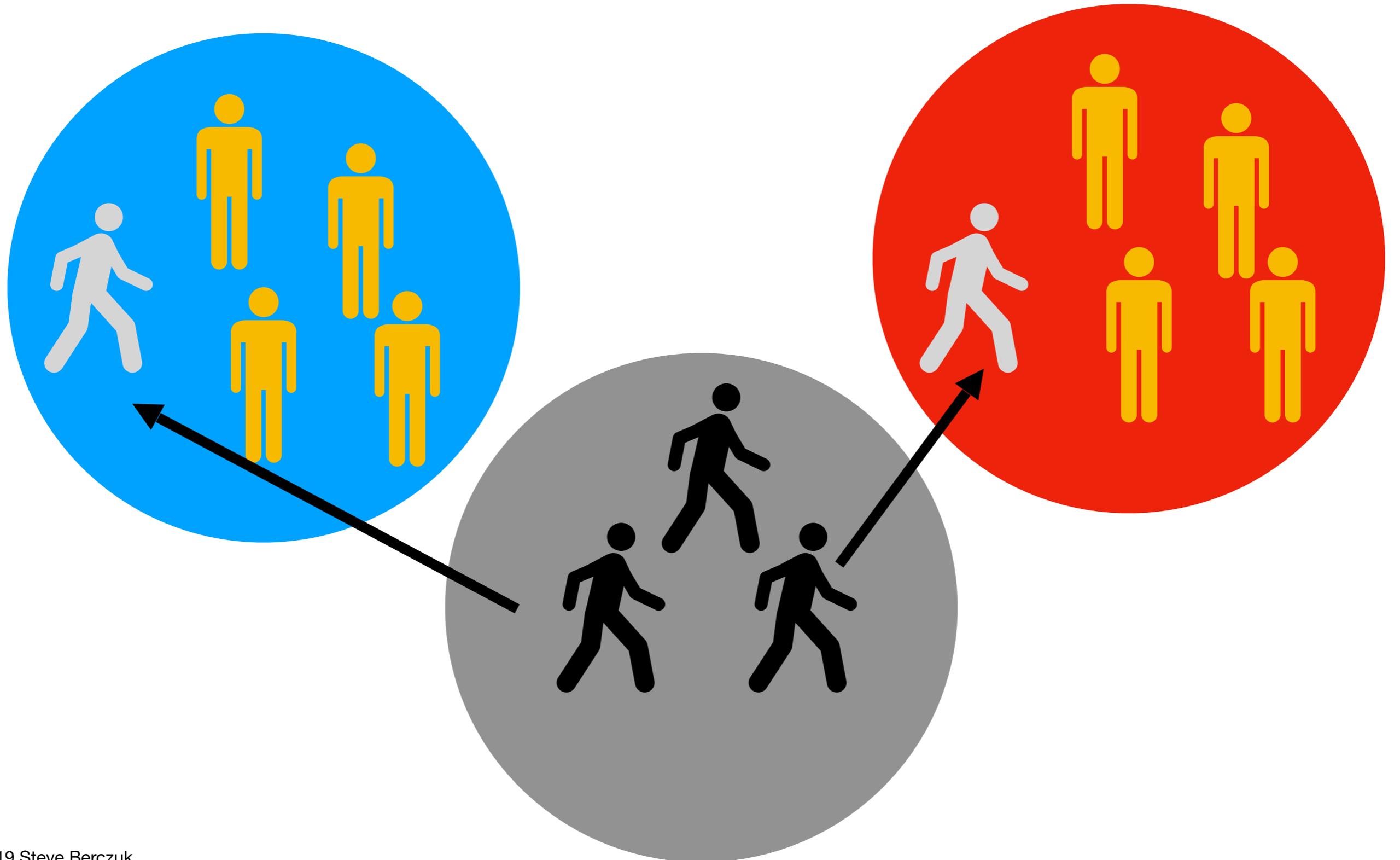


Scrum Pillars and Values

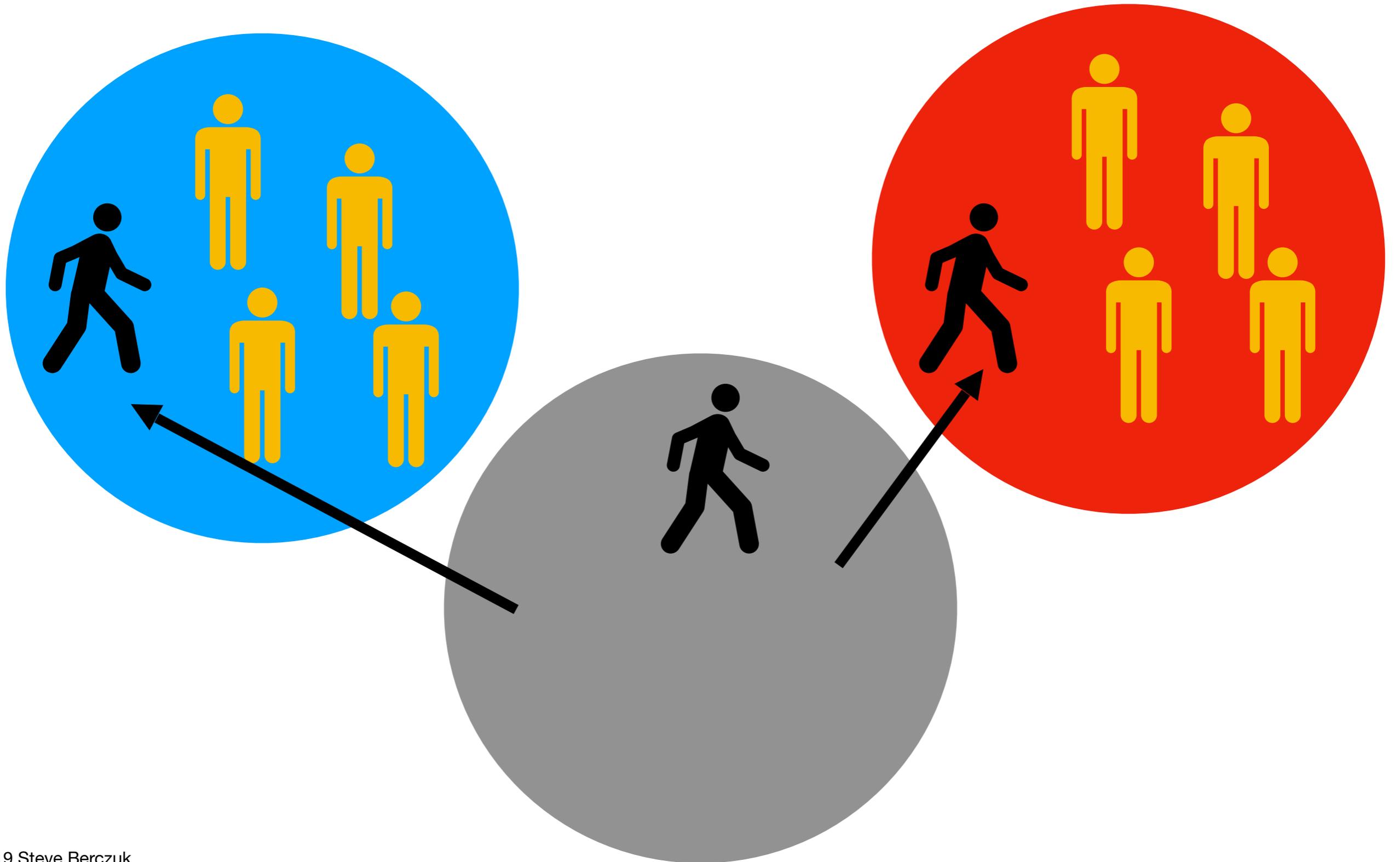


Specialist Teams

Specialist Teams



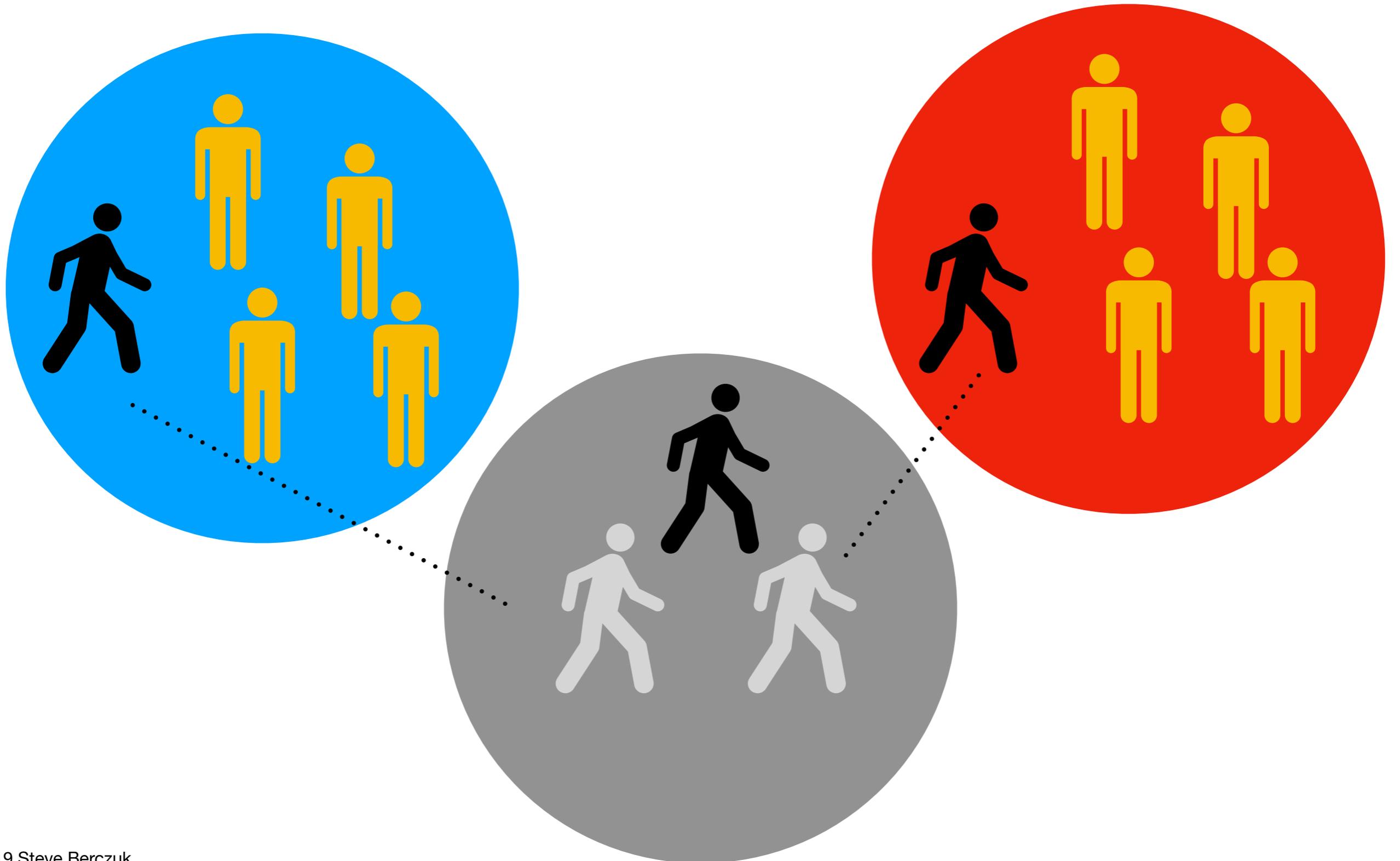
Specialist Teams



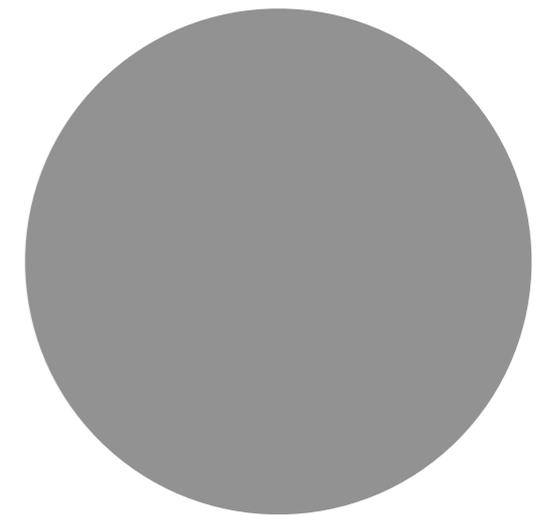
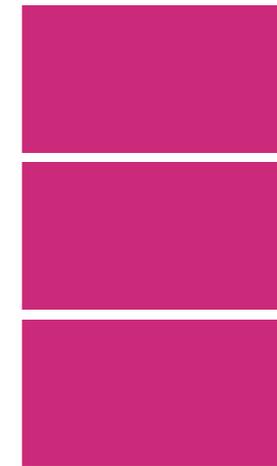
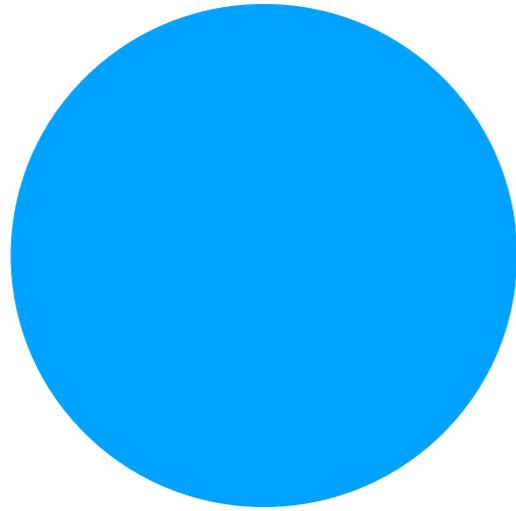
Specialist Teams



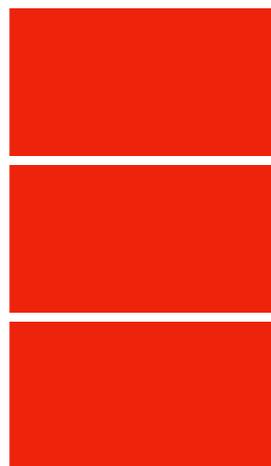
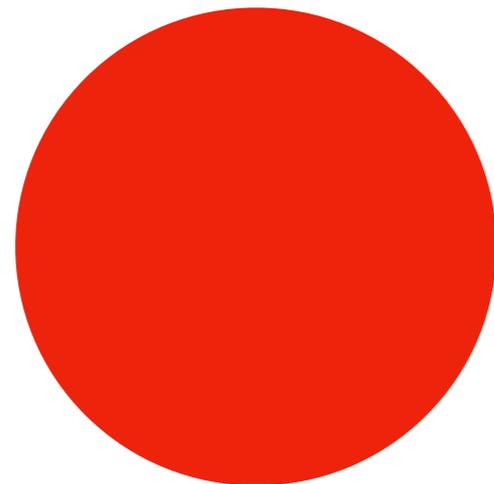
Specialist Teams



Specialist & Team Backlogs



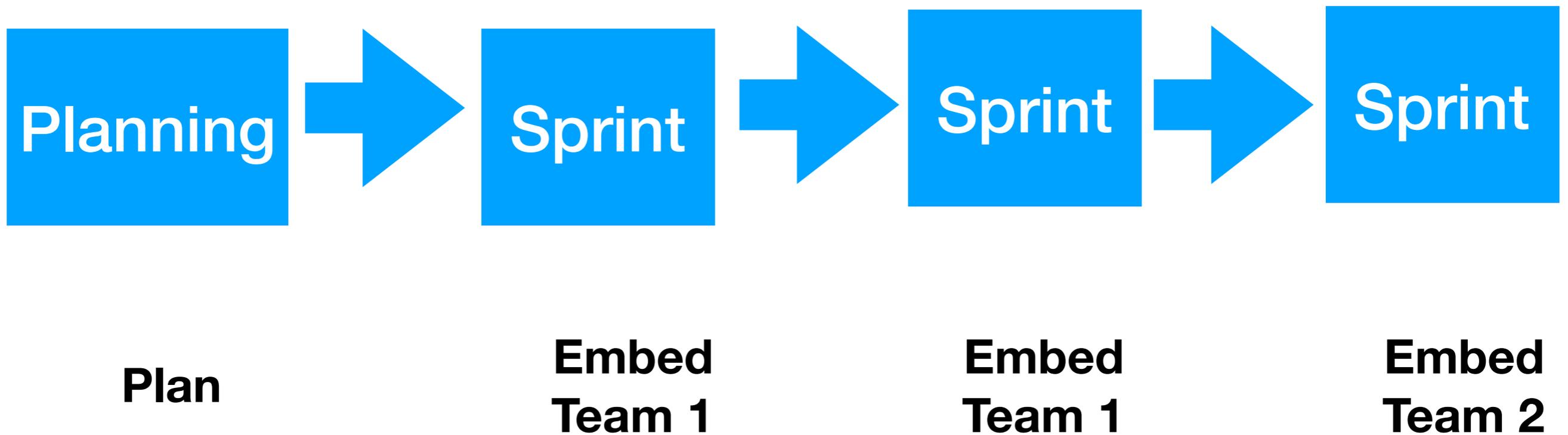
**Rotation or
% Allocation**



Specialist Teams

- **Embed** with Scrum Teams for **multiple sprints**
- Train, **Coach**, and Build Infrastructure
- Identify **Common Needs** Across Teams
- **Share** Knowledge Across Teams
- Build **Shared Tools** and Frameworks

When?



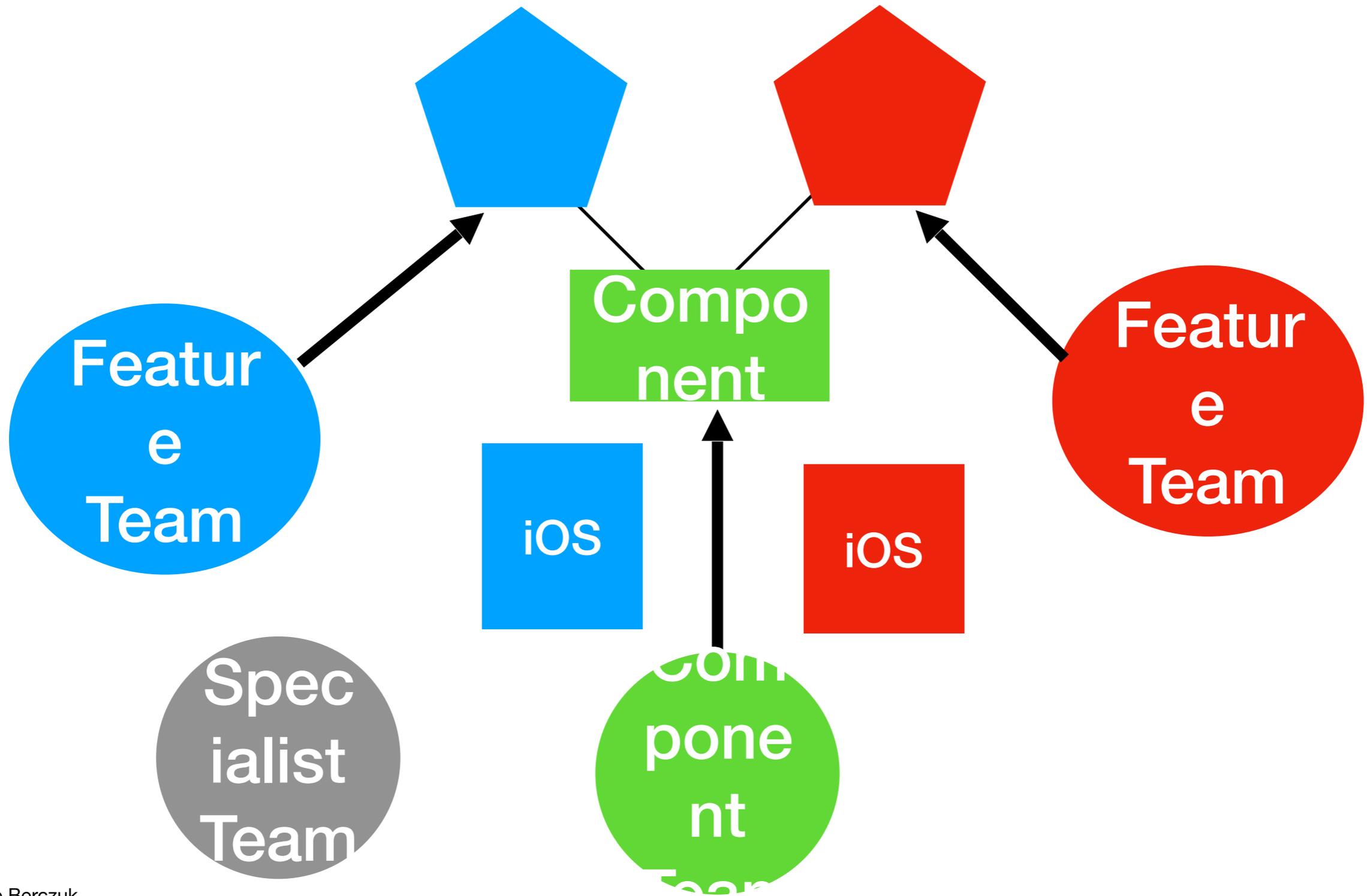
Benefits: Knowledge Sharing

- Embedding for a period of time -> Commitment
- Sharing of knowledge and techniques
- Creation of communication channels between teams

Other Approaches to shared Special Tools

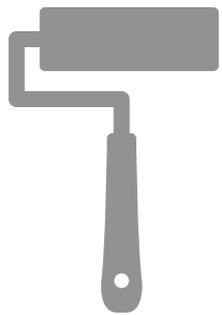
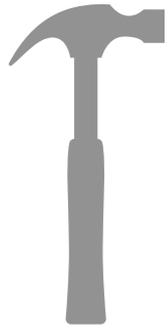
- Collective Code Ownership
 - Teams make changes as needed & Share (versioning)
 - Internal Open Source (Specialists as gatekeepers)
- Component Team
 - Need to stay close to needs/use

Types of Teams



Things to Do

- Visualize your team's skill set
 - Jan Beaver Visualizing Cross Functionality (https://www.greyrockagile.com/presos/Visualizing_CrossFunctionality_Jan_Beaver_SGATL.pdf)
- Identify gaps
- Consider how you organize teams

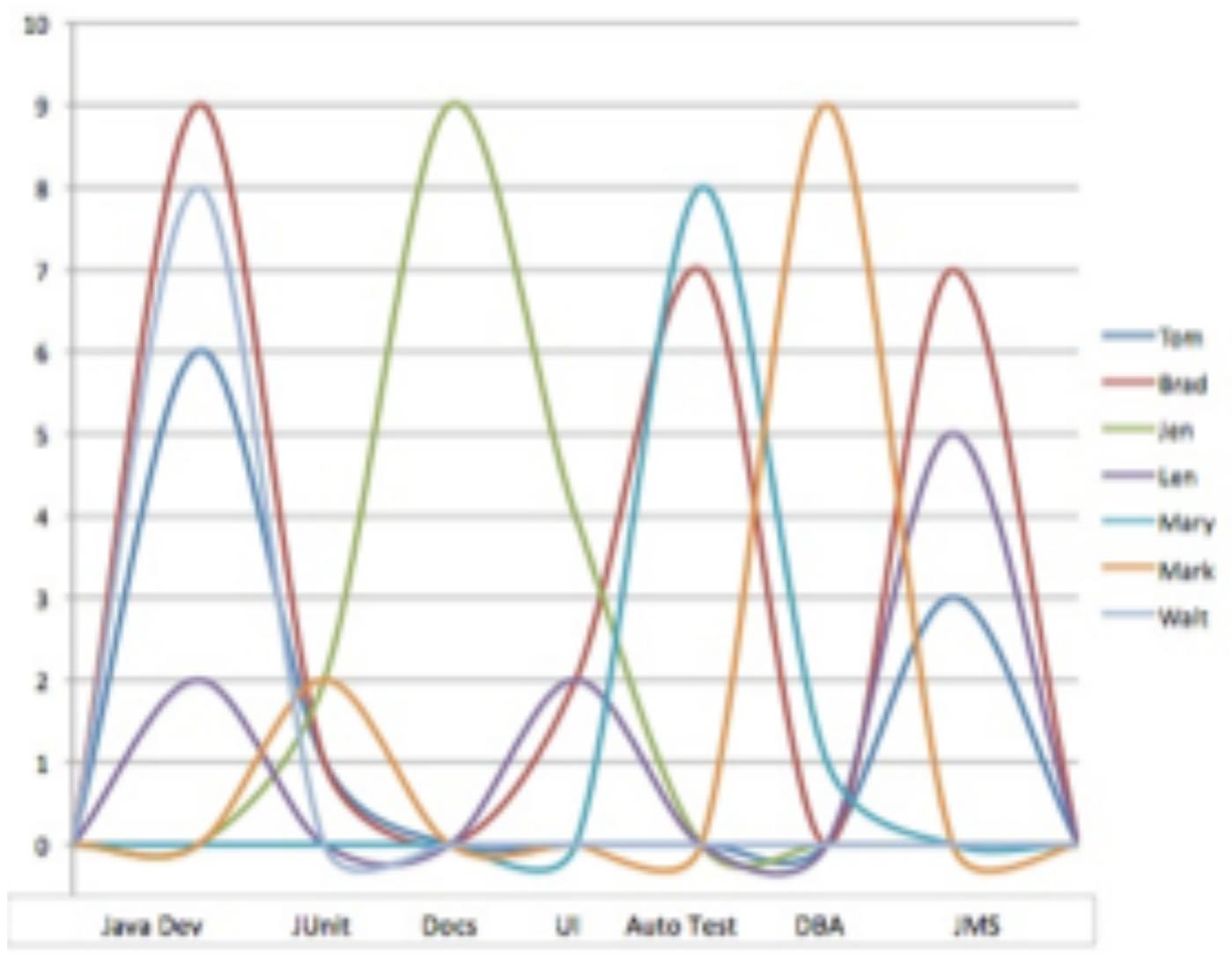


These are all Engineering Skills



Expertise, Training & Learning





Summary

- Feature Teams
 - Build Features (Customer Visible)
- Component Teams
 - Build Shared Components
- Specialist Teams
 - Embed with Feature and Component Scrum Teams
 - Most members are embedded in other Scrum Teams
 - Enable Teams to forecast more accurately with less overhead

Questions?

Contact

- Twitter: @sberczuk
- Linked In: steveberczuk
- steve@berczuk.com
- www.berczuk.com