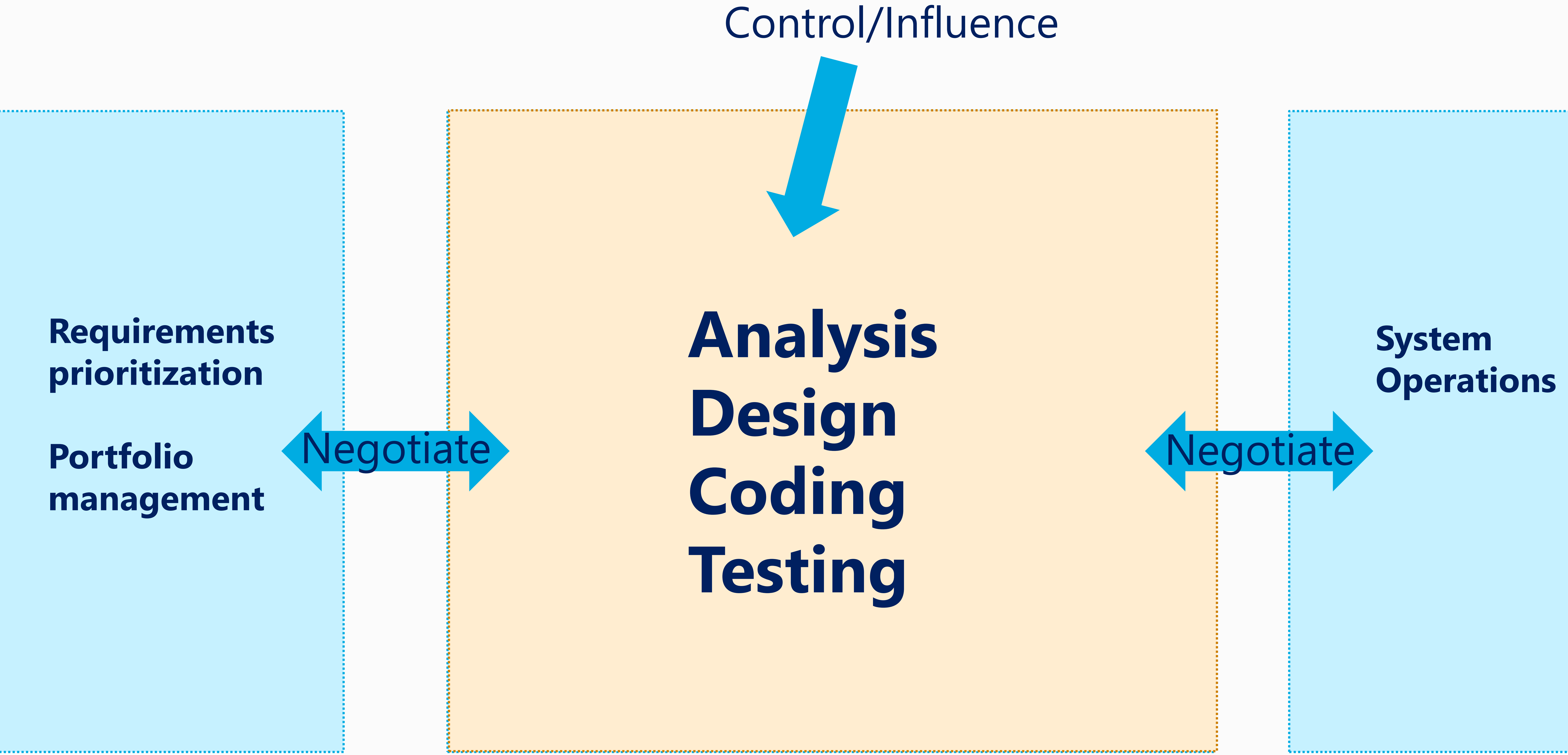
Implementing Kanban

9



Control/Influence

Requirements prioritization

Portfolio management

Negotiate

Analysis

Design

Coding

Testing

Negotiate

System Operations

Kanban

is an approach that drives change

by optimizing

your existing process.

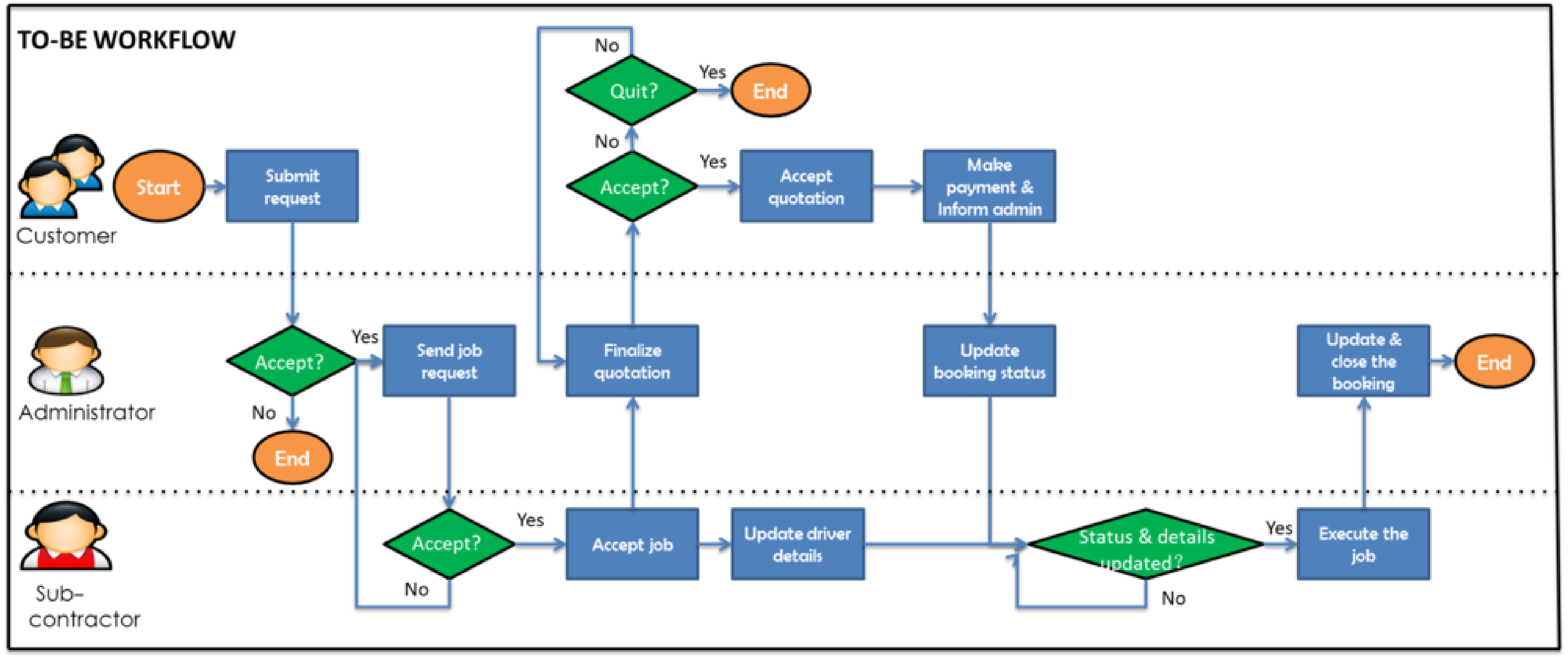
Step 1
Mapping the Value Stream

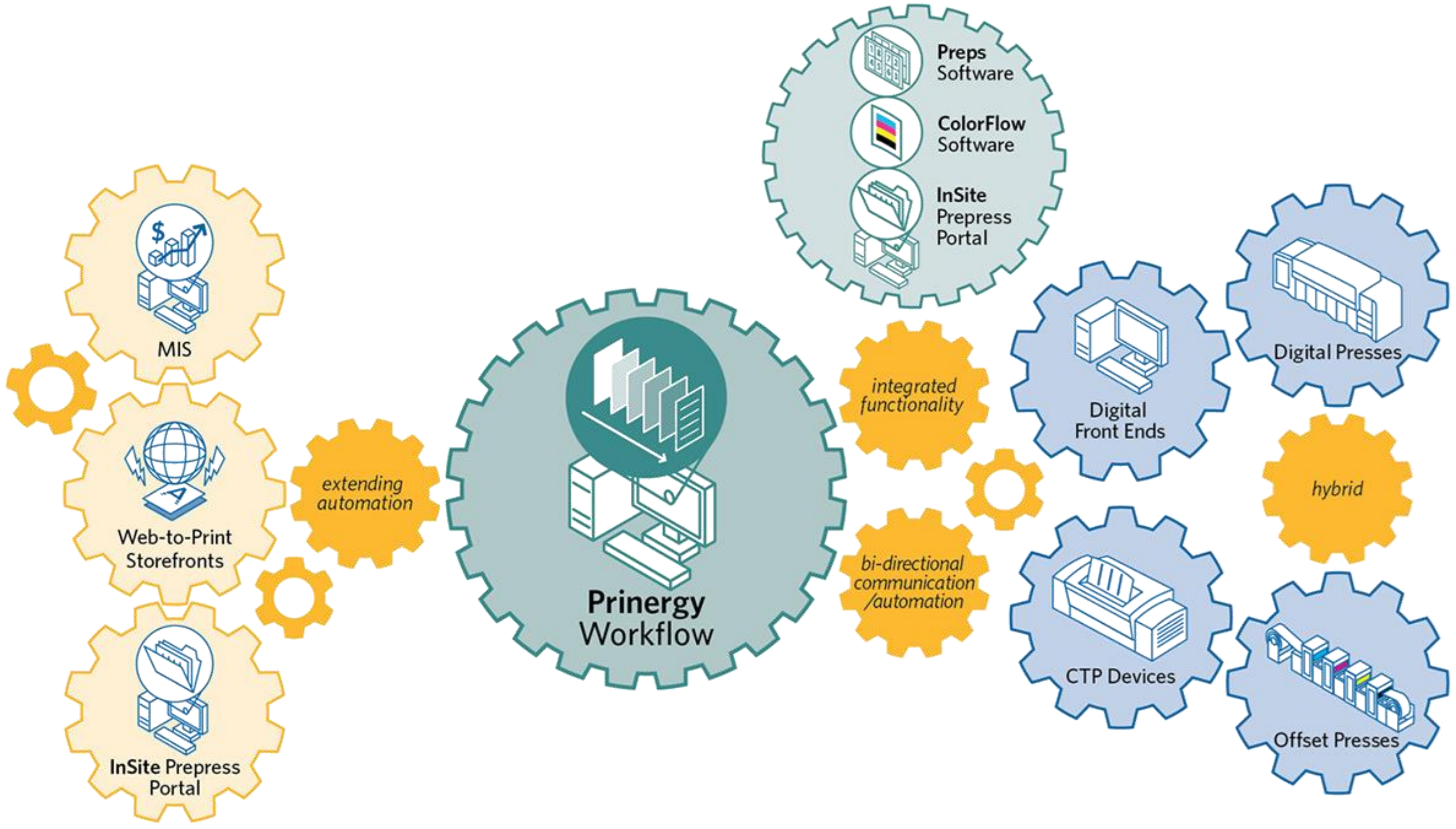
1. Mapping the Value Stream

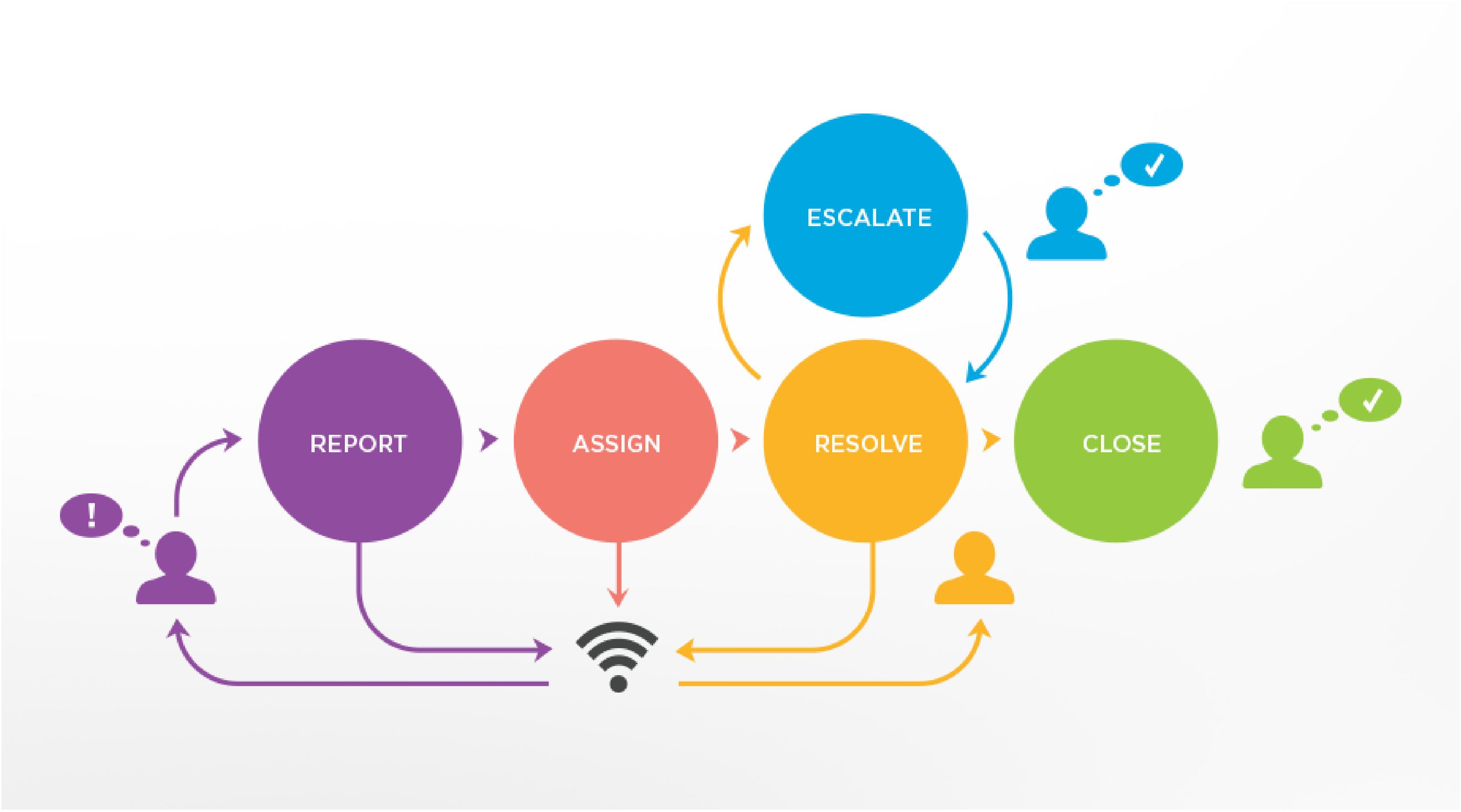
Identify Work Item Types

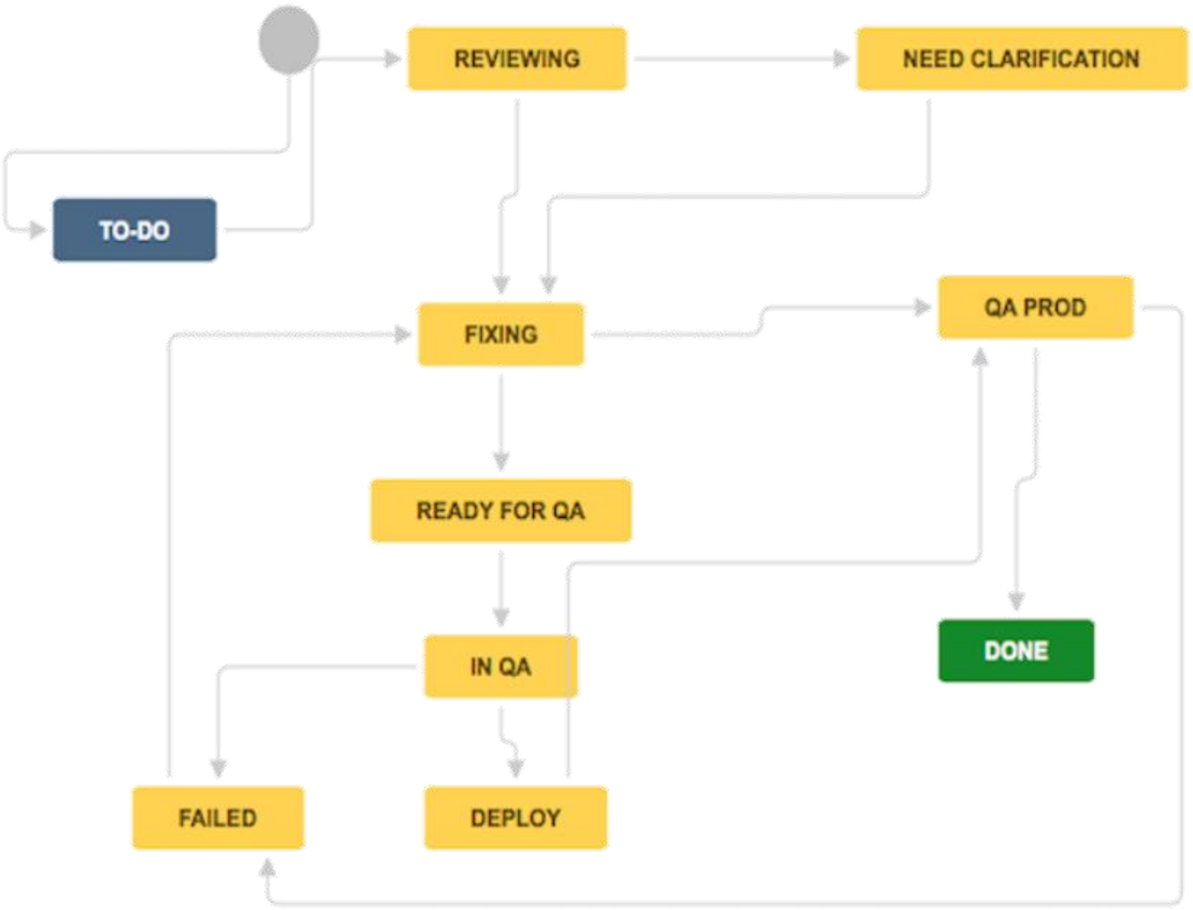
- Requirement
- Feature
- User Story
- Use Case
- Change Request
- Production Defect
- Maintenance
- Refactoring
- Bug Improvement
- Suggestion
- Blocking Issue

Sketch the workflow



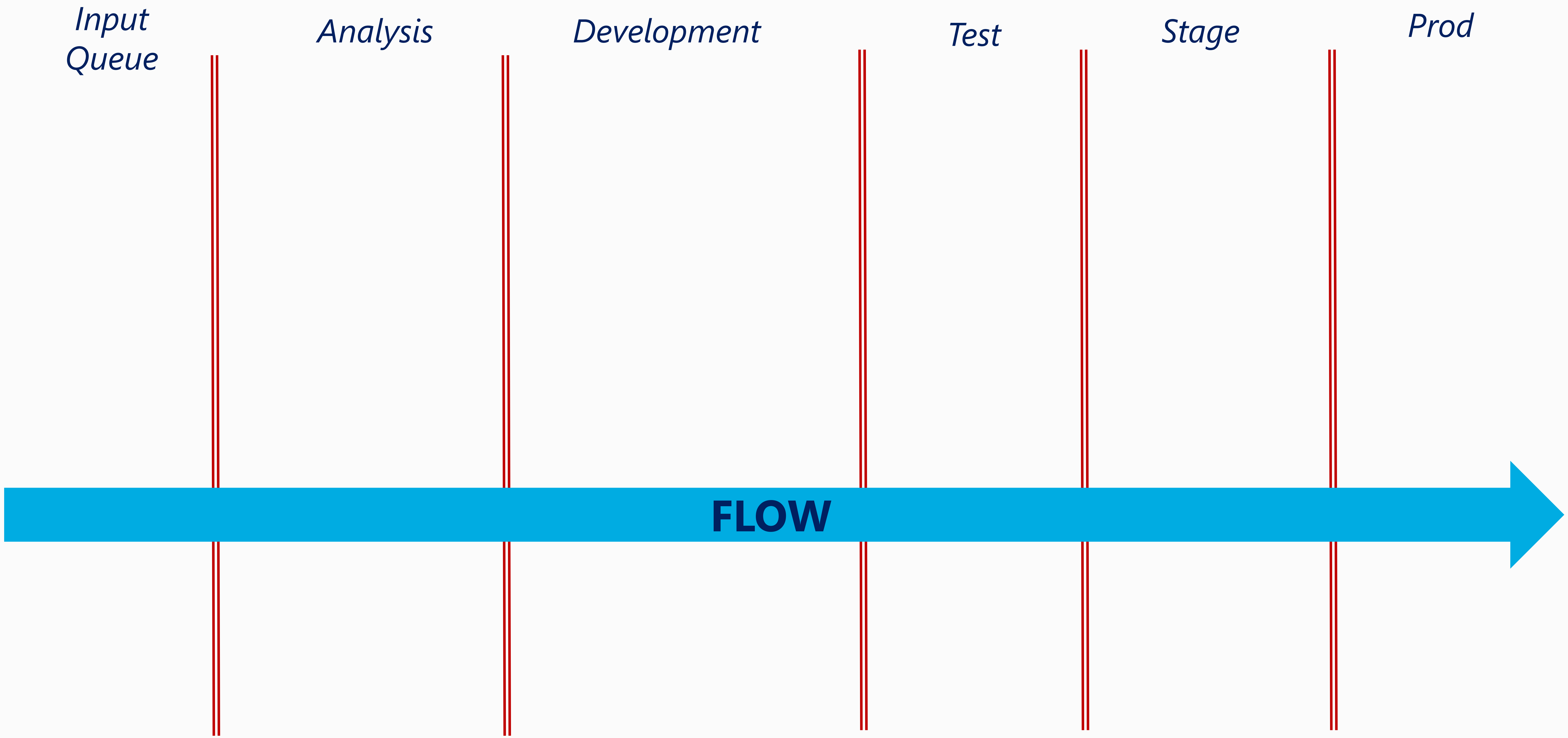






1. Mapping the Value Stream

Drawing a Card Wall



Input Queue

Analysis

Development

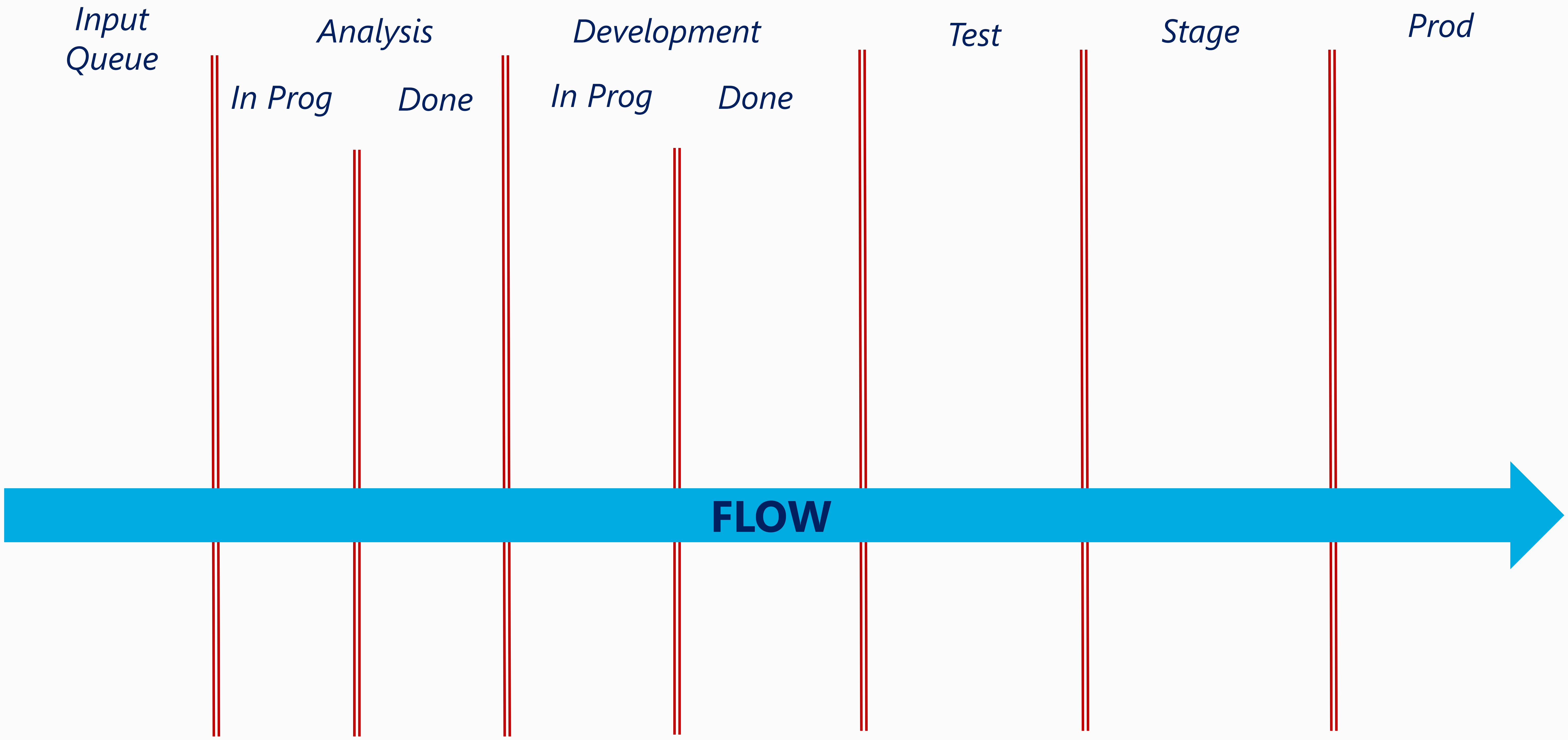
Test

Stage

Prod

FLOW

1. Mapping the Value Stream



Input Queue

Analysis

In Prog *Done*

Development

In Prog *Done*

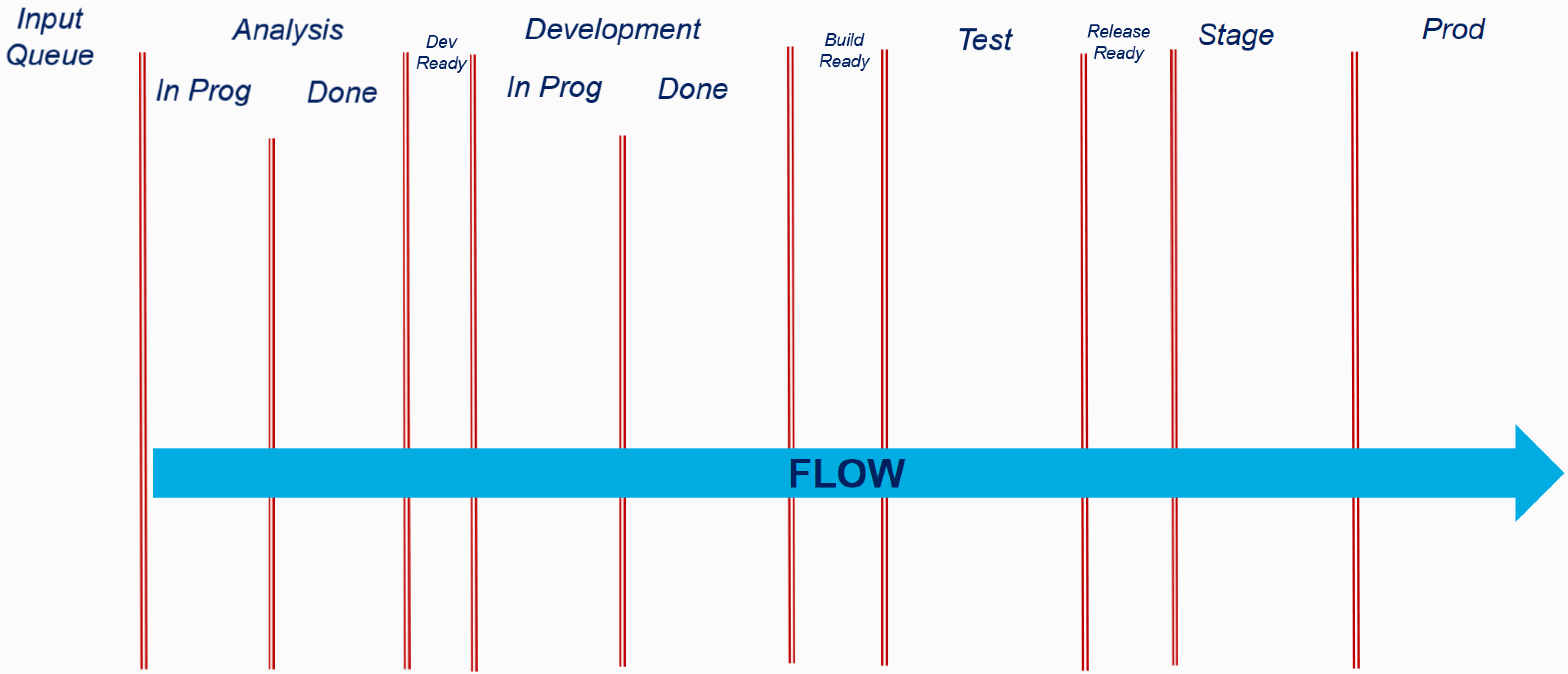
Test

Stage

Prod

FLOW

1. Mapping the Value Stream



Input Queue

Analysis

In Prog

Done

Dev Ready

Development

In Prog

Done

Build Ready

Test

Release Ready

Stage

Prod

FLOW

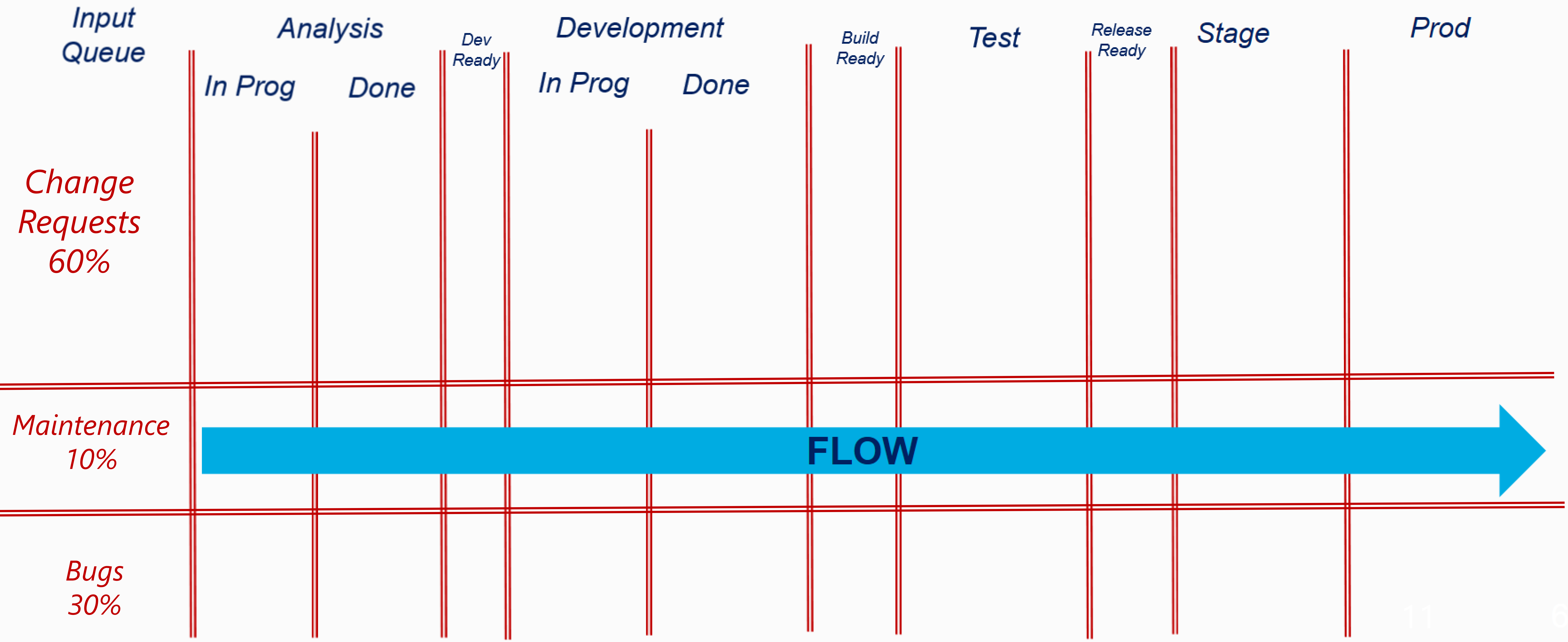
1. Mapping the Value Stream

How to put buffers?

Approach 1. Implement the system. Wait for the bottleneck to reveal itself. Make changes to introduce a buffer.

Approach 2. Each stage should be buffered. Bottlenecks and variability will reveal themselves by how full the buffers become.

1. Mapping the Value Stream



Input Queue

Analysis

In Prog

Done

Dev Ready

Development

In Prog

Done

Build Ready

Test

Release Ready

Stage

Prod

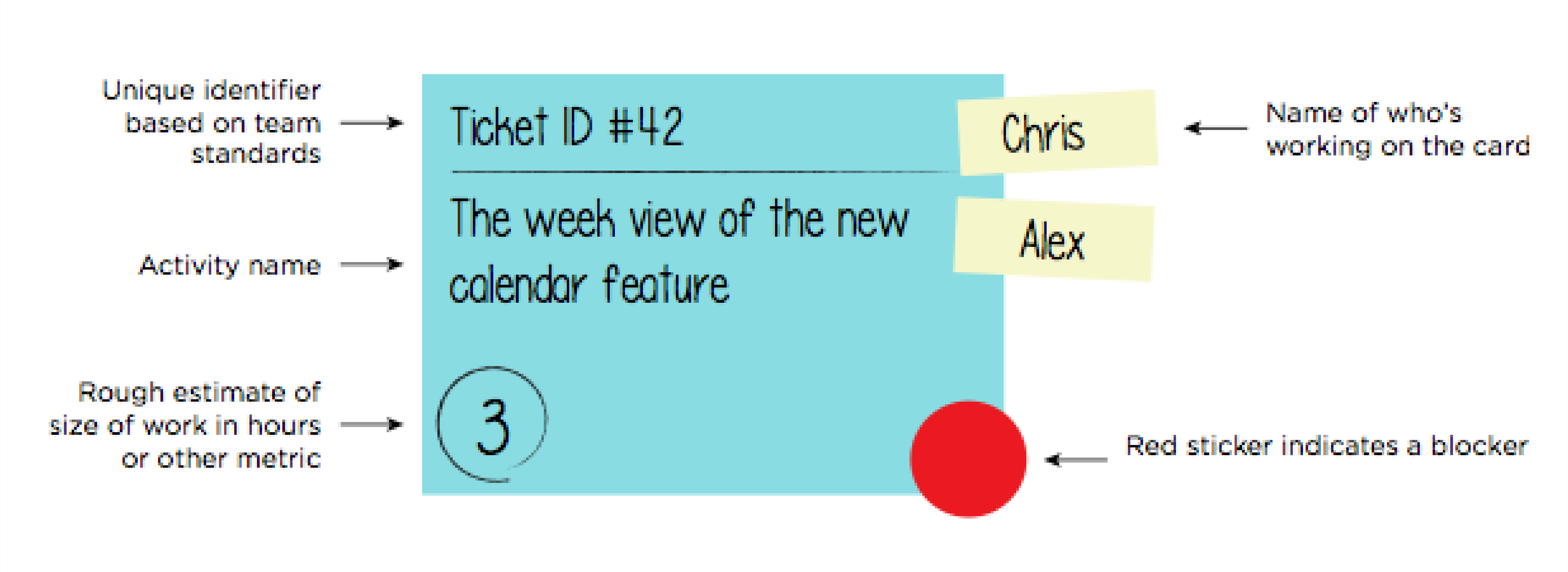
Change Requests 60%

Maintenance 10%

Bugs 30%

1. Mapping the Value Stream

Work Item Card Structure



Unique identifier based on team standards

Activity name

Rough estimate of size of work in hours or other metric

Ticket ID #42

The week view of the new calendar feature

3

Chris

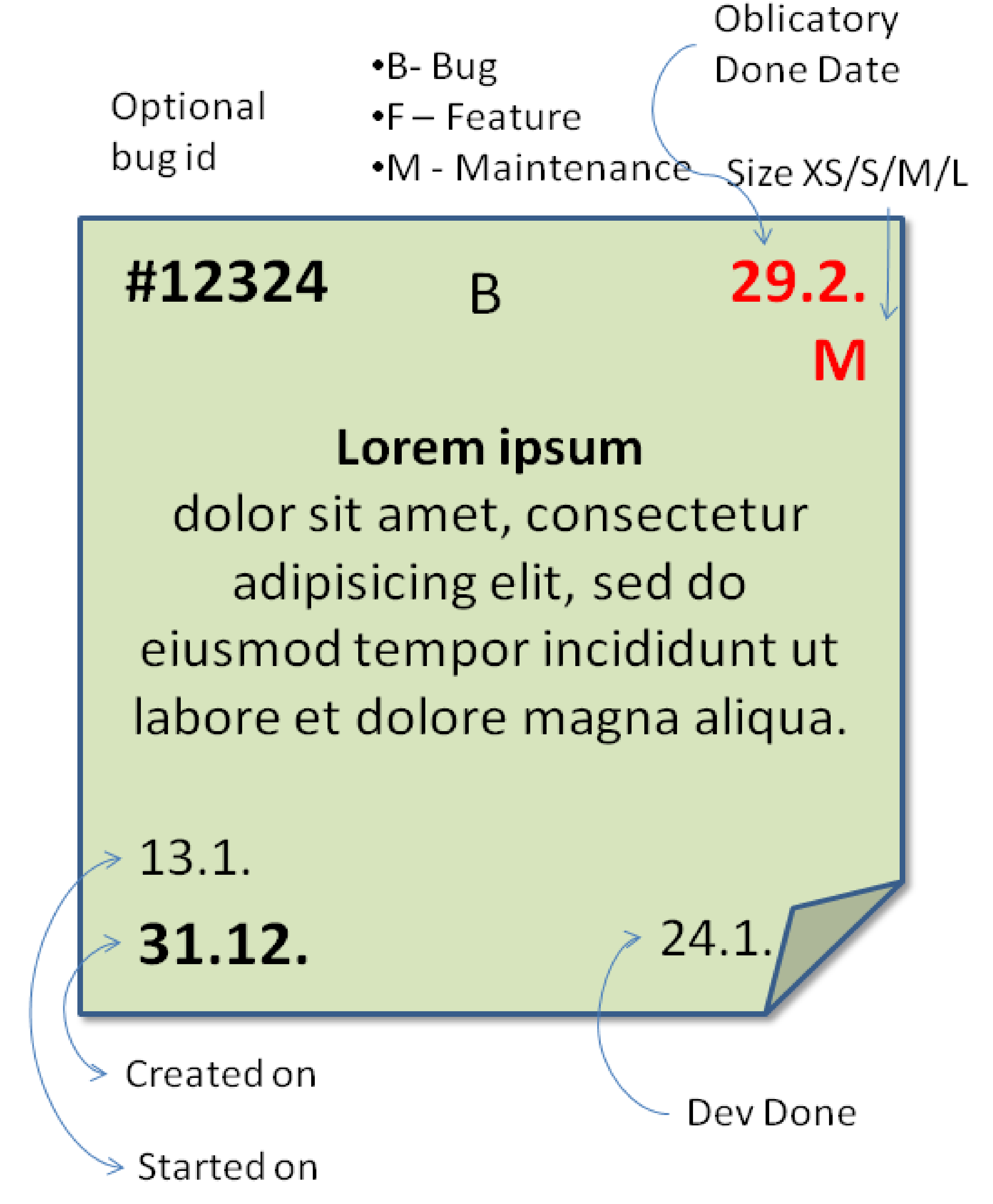
Alex

Name of who's working on the card

Red sticker indicates a blocker

1. Mapping the Value Stream

Work Item Card Structure



Optional bug id

- •B- Bug
- •F – Feature
- •M - Maintenance

Oblicatory Done Date

Size XS/S/M/L

#12324 B **29.2.** **M**

Lorem ipsum
dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

13.1.

31.12.

24.1.

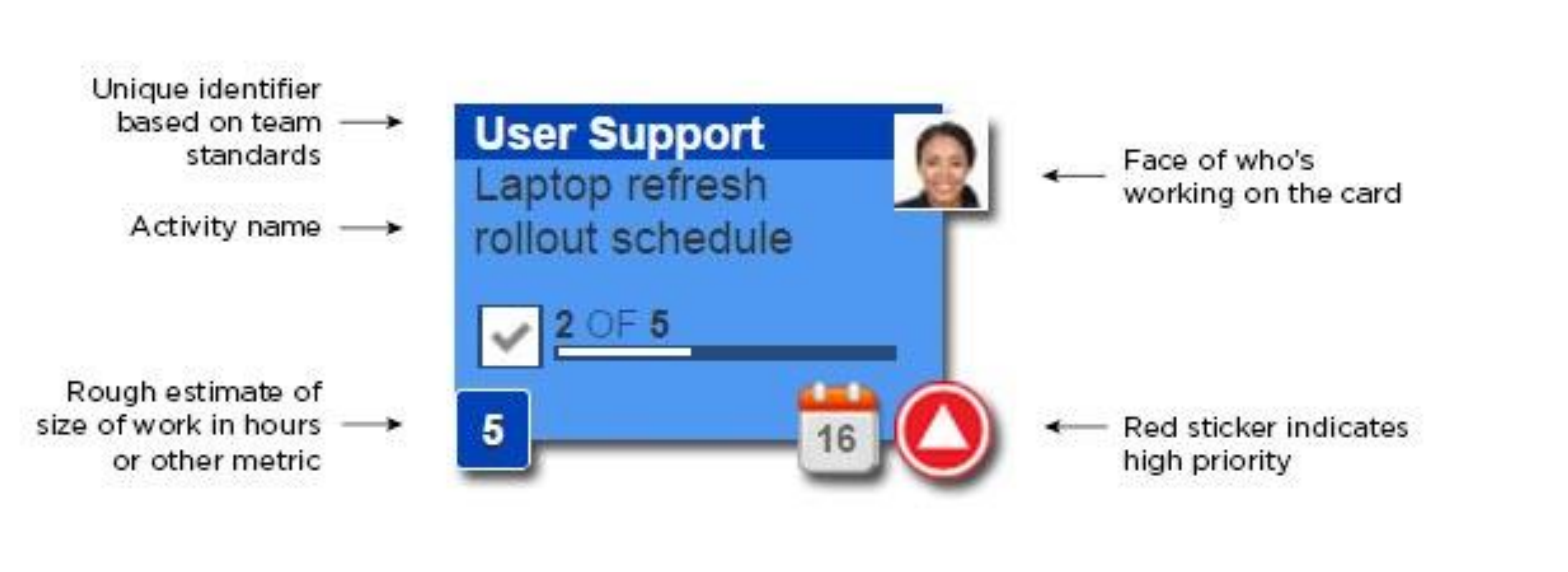
Created on

Started on

Dev Done

1. Mapping the Value Stream

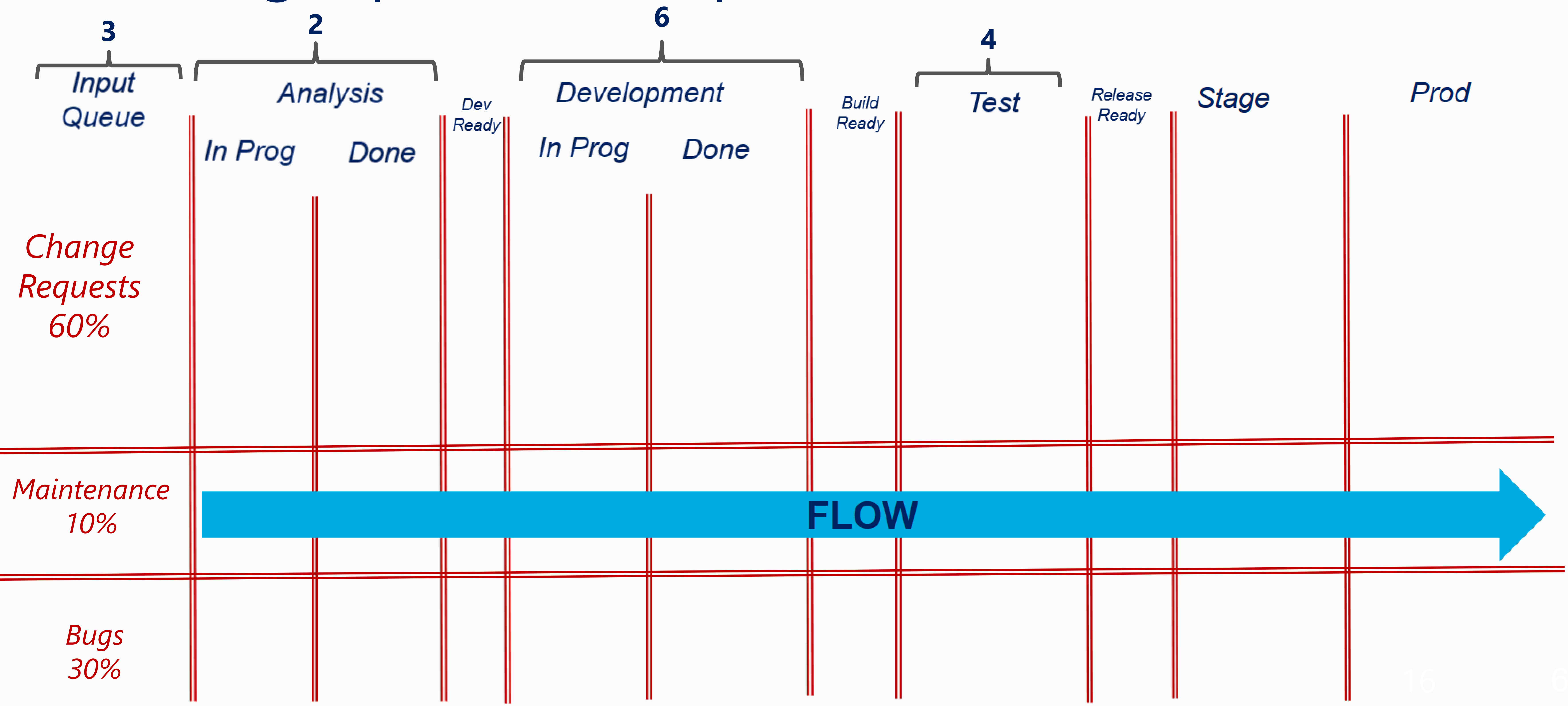
Work Item Card Structure


Unique identifier based on team standards
Activity name
Rough estimate of size of work in hours or other metric
User Support
Laptop refresh rollout schedule
2 OF 5
5
16
Face of who's working on the card
Red sticker indicates high priority

The design of the card should have sufficient information to facilitate project-management decisions, such as which item to pull next, without the intervention or direction of a manager => empower team members with transparency of process, project goals and objectives, and risk information.

1. Mapping the Value Stream

Setting Input and Output Boundaries



3 Input Queue

2 Analysis: In Prog, Done

Dev Ready

6 Development: In Prog, Done

Build Ready

4 Test

Release Ready

Stage

Prod

Change Requests 60%

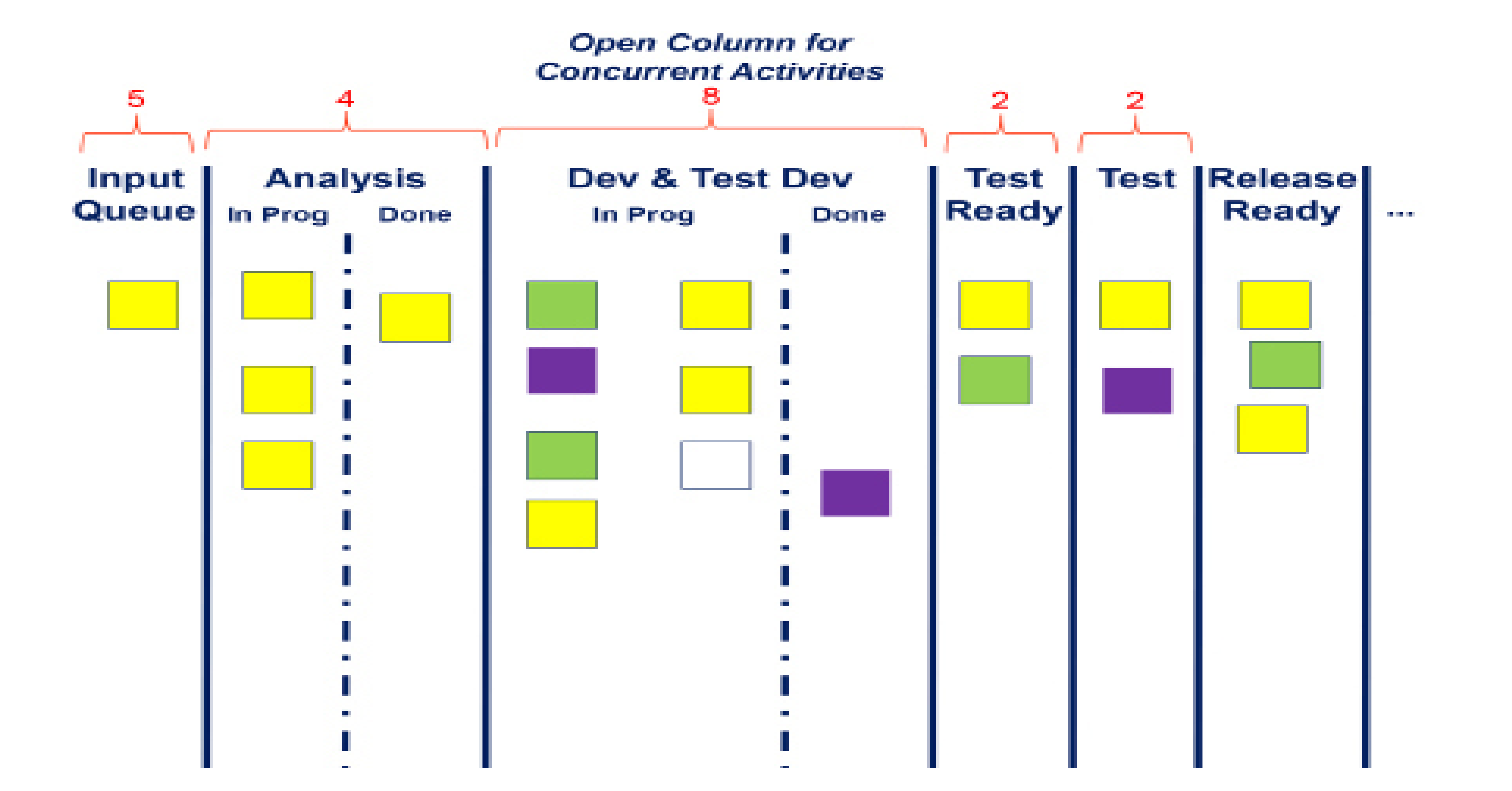
Maintenance 10%

Bugs 30%

FLOW

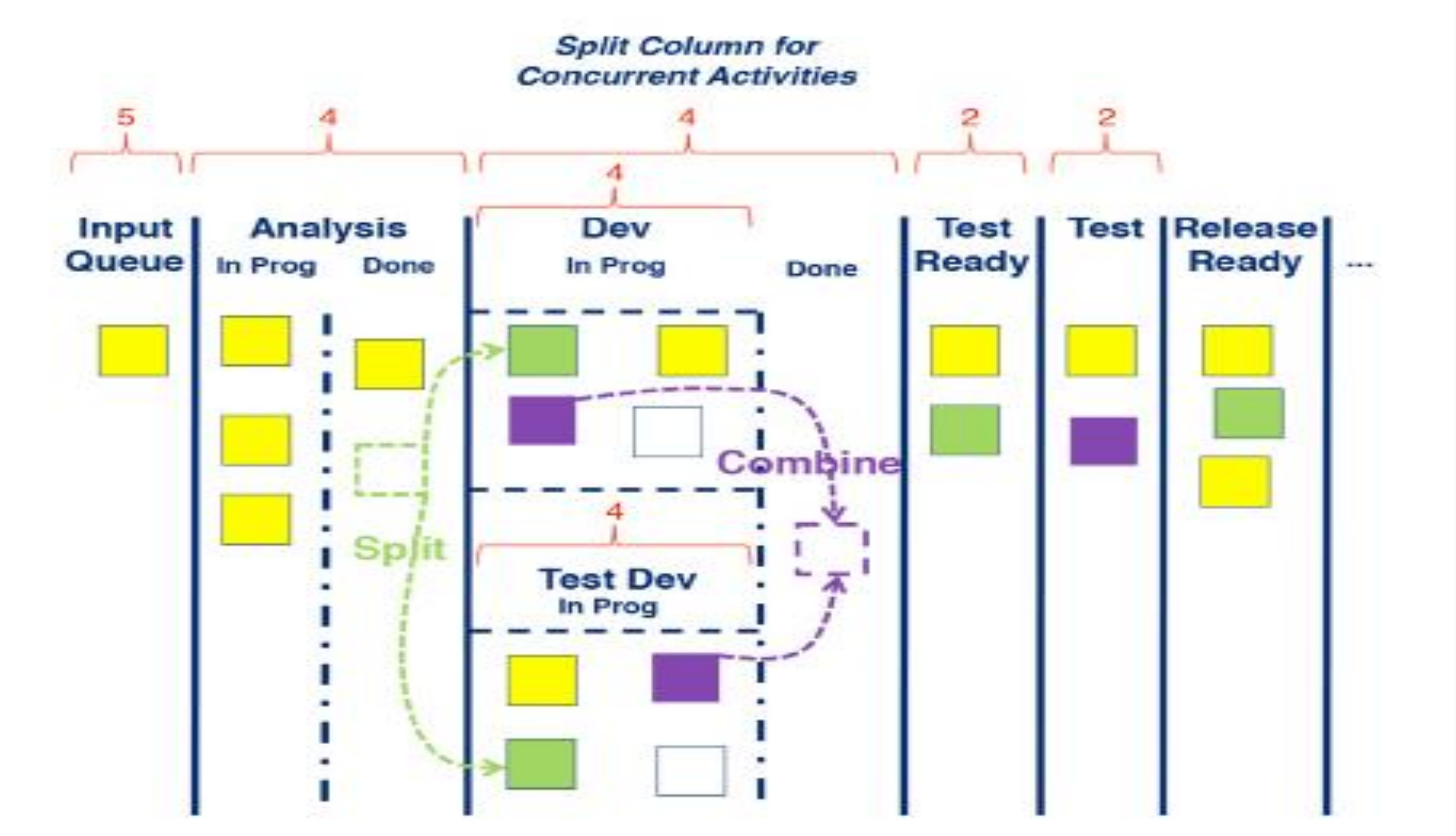
1. Mapping the Value Stream

Coping With Concurrency



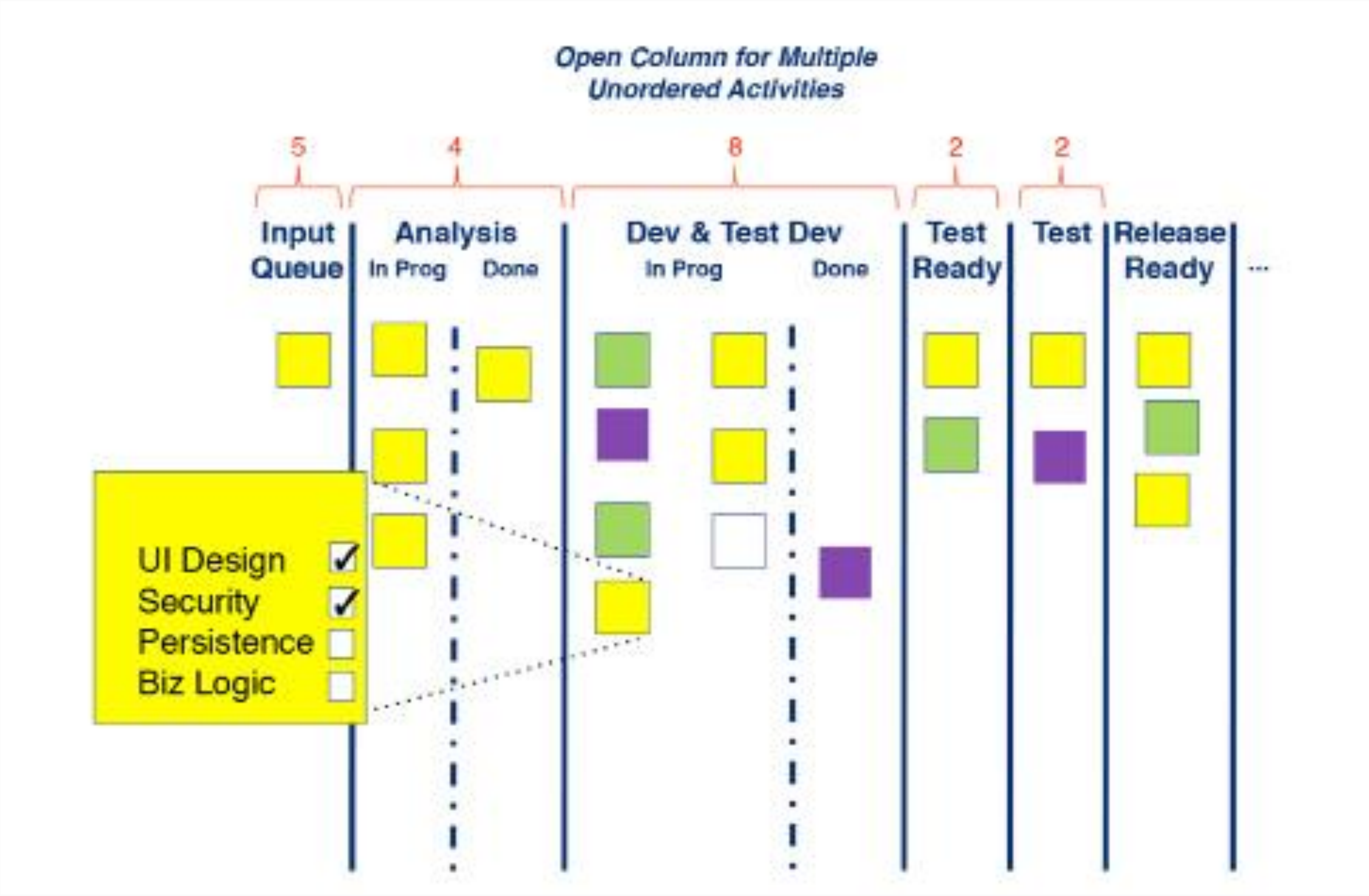
1. Mapping the Value Stream

Coping With Concurrency



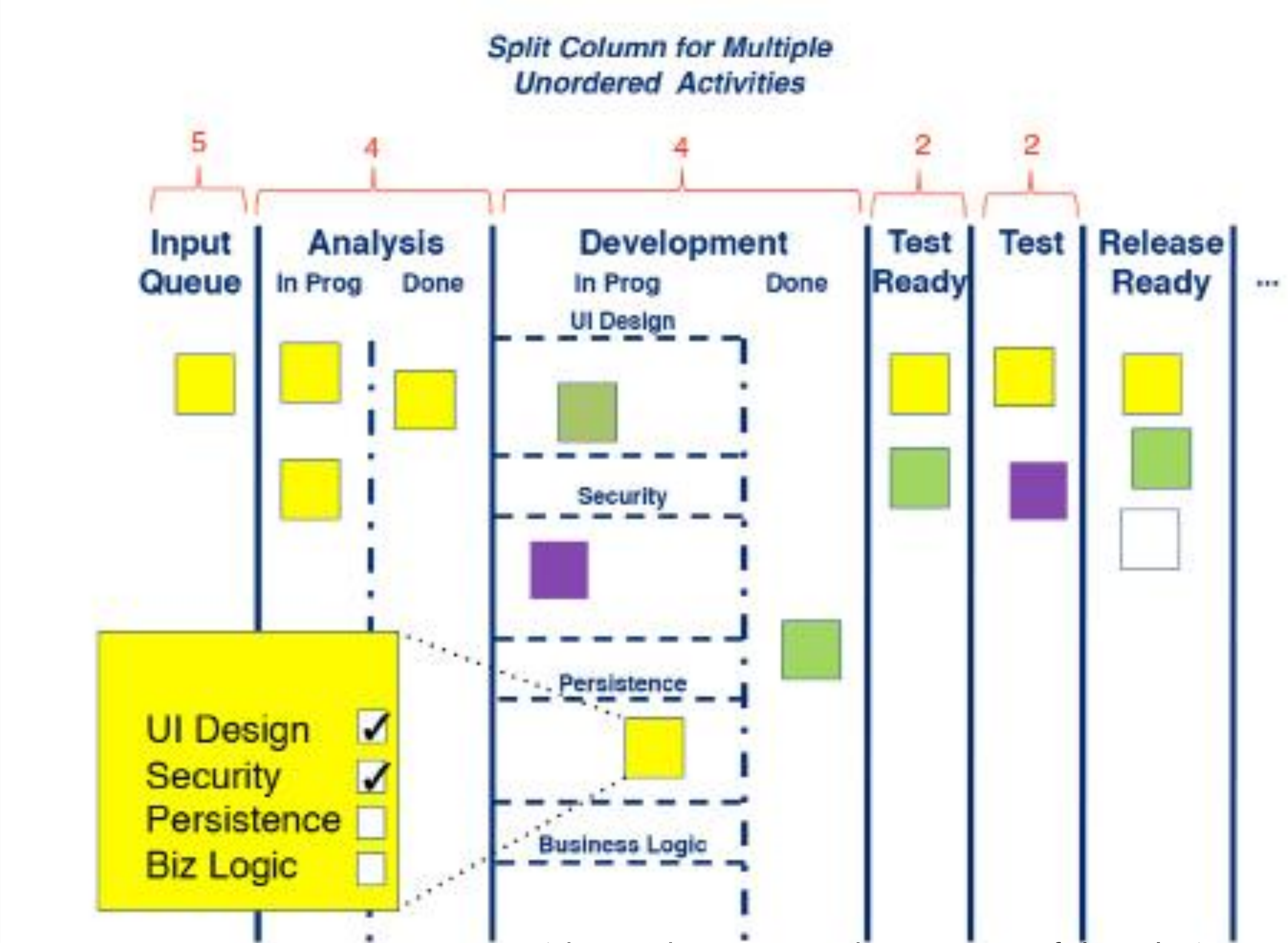
1. Mapping the Value Stream

Coping With Unordered Activities



1. Mapping the Value Stream

Coping With Unordered Activities



Source: David J. Anderson - Kanban: Successful Evolutionary Change for Your Technology Business

1. Mapping the Value Stream

Electronic Tracking System

- Lean Kit Kanban,
- Agile Zen,
- Target Process,
- Silver Catalyst,
- RadTrack,
- Kanbanery,
- VersionOne,
- Greenhopper for Jira,
- Flow.io

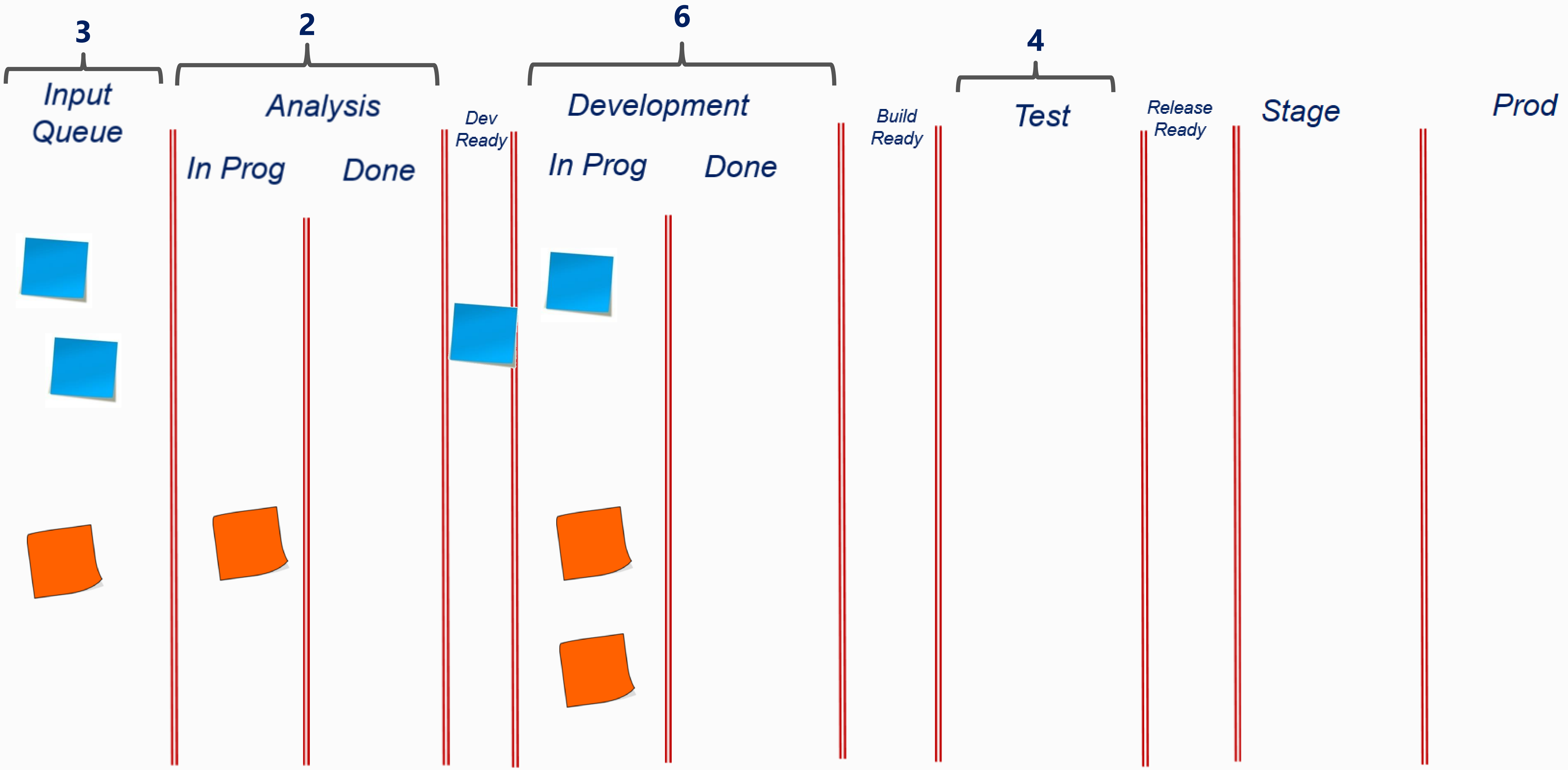
Step 1
Mapping the Value Stream

- Identify work item types
- Draw a card wall
  - Define swim lanes
  - Add buffers/queues
  - Allocate capacity according to demand
- Define Work item card structure
- Set input/output boundaries
- Cope with concurrency & unordered activities
- Chose an electronic tracking system

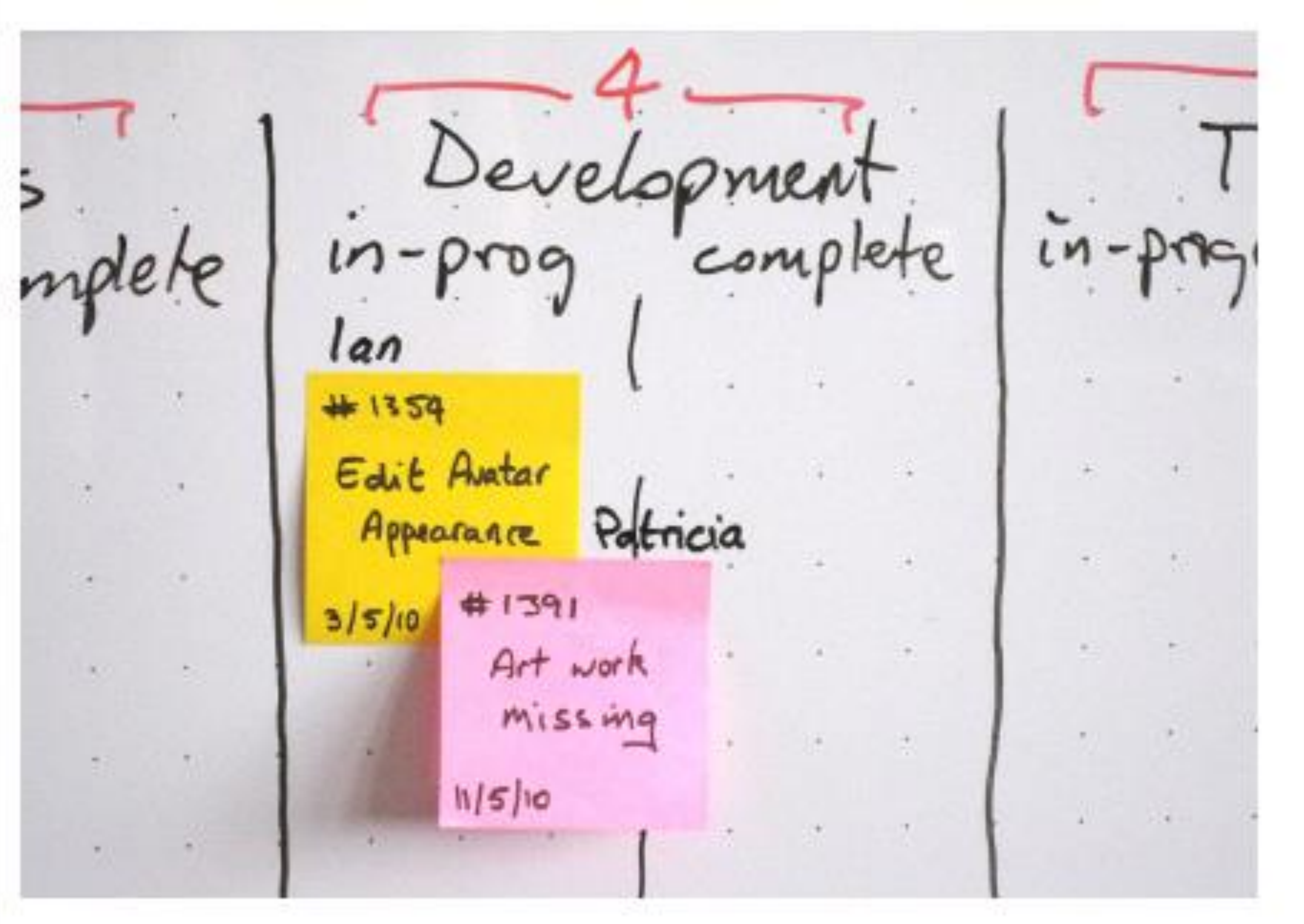
Do not force visualization, transparency, and WIP limits
on any department that does not volunteer to collaborate.

Step 2
Coordination

2. Coordination



2. Coordination



The goal is to visually communicate enough information to make the system self-organizing and self-expediting at the team level.

2. Coordination

Daily Standup Meetings

- Similar with Scrum meetings
- The cards answer the 3 standup questions

  => the focus is on flow of work

  - The manager/facilitator will “walk the board”
  - From right to left (in the direction of pull)
  - More emphasis on blocked/delayed items (mark a dot behind the ticket for each day it stays in one single location) – *mature teams will talk only about these items*
- Discussion on any other blocking issue that is not on the board
- Followed by The After Meeting (detailed discussions on issues)

2. Coordination

Queue Replenishment Meeting

- Frequency: weekly (2 hours)
- Main purpose: prioritization
- Postponed until the last reasonable moment
- Together with business representatives (*from potentially competing groups within the company*)
- The cadence of these meetings affect the queue sizing => affect overall lead time

2. Coordination

Release Planning Meeting

- Regularly or not (depends on releases occurrence)
- All interested parties are invited (including SO & network specialists, developers, testers, business analysts etc).
- Risk analysis
- The outcome is a release plan

2. Coordination

Triage

- Medical term to classify emergency patients:
    - Beyond help and likely to die soon
    - Likely to live only if given immediate treatment
    - Likely to survive without immediate treatment
- In software the term is used to classify defects (sometimes it is called “backlog grooming”)
- Infrequent intervals
- Keep or delete choice
- Main purpose: reduce backlog size (easier to prioritize from 200 items than from 2000)

2. Coordination

Synchronizing across geographic locations

- use electronic tracking
- stand-up meeting over the telephone / video-conference system
- “sticky buddies” ☺

Step 2
Coordination

- Use both physical card wall & electronic tracking system
- Regular meetings reduces coordination cost
- Prioritization & release planning should be done independently and should have independent cadence
- Daily standups used for discussing issues and flow
- Daily standups are followed by informal process-improvement discussions
- Escalation paths & policies should be clearly defined

Step 3
Delivery Cadence

3. Delivery Cadence

= An agreed-upon, regular interval between deliveries of working software

- Kanban decouples delivery cadence from both development lead time and prioritization cadence
- Short, time-boxed iterations have led to dysfunction with some teams attempting to adopt Agile development methods

3. Delivery Cadence

- Kanban decouples delivery cadence from both development lead time and prioritization cadence

- Short, time-boxed iterations have led to dysfunction with some teams attempting to adopt Agile development methods

3. Delivery Cadence

Efficiency of Delivery

Delivery Efficiency% =

100% x (Total Cost – (Coordination Cost + Transaction Cost)) /
Total Cost of Software Release

To be more efficient we have to
(a) increase the time between deliveries, or
(b) reduce the coordination and transaction costs.

Regular delivery builds trust.

3. Delivery Cadence

Efficiency of Delivery

Ad hoc / on-demand delivery can make sense for:

- high-maturity organizations with established,
- high levels of trust
- low transaction- and coordination costs of delivery.

Step 4
Setting Work-in-Progress Limits

4. Setting Work-in-Progress Limits

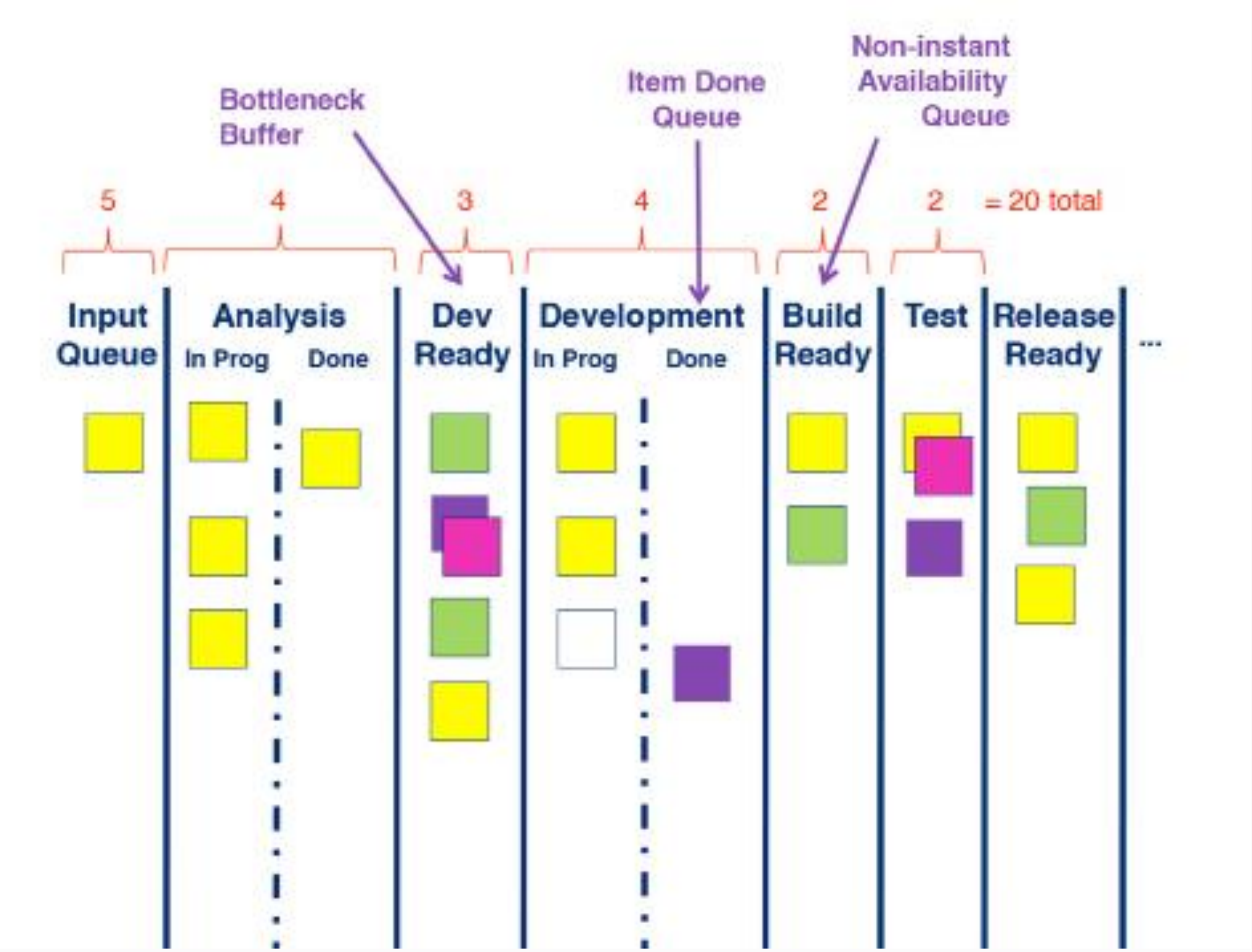
- WIP limits should be agreed with *up*- and *downstream stakeholders* & *senior function management*.
- Unilateral declaration of WIP limits is possible but may prove difficult to defend later, when the system is placed under stress.
- WIP limits for work tasks should be set as an average number of items per person, developer pair, or small, collaborative team.
    - Typically, the limit should be in the range of one to three items per person, pair, or team.

When introducing Kanban:

- Business owners should be educated on:
  - the workflow
  - the input queue
  - the pull mechanism
- If there are 2 slots free in the queue:
  - “Which two new items would you like next?”
  - “Which two items would you most like delivered 30 days from now?” (based on the avg. lead-time)

4. Setting Work-in-Progress Limits

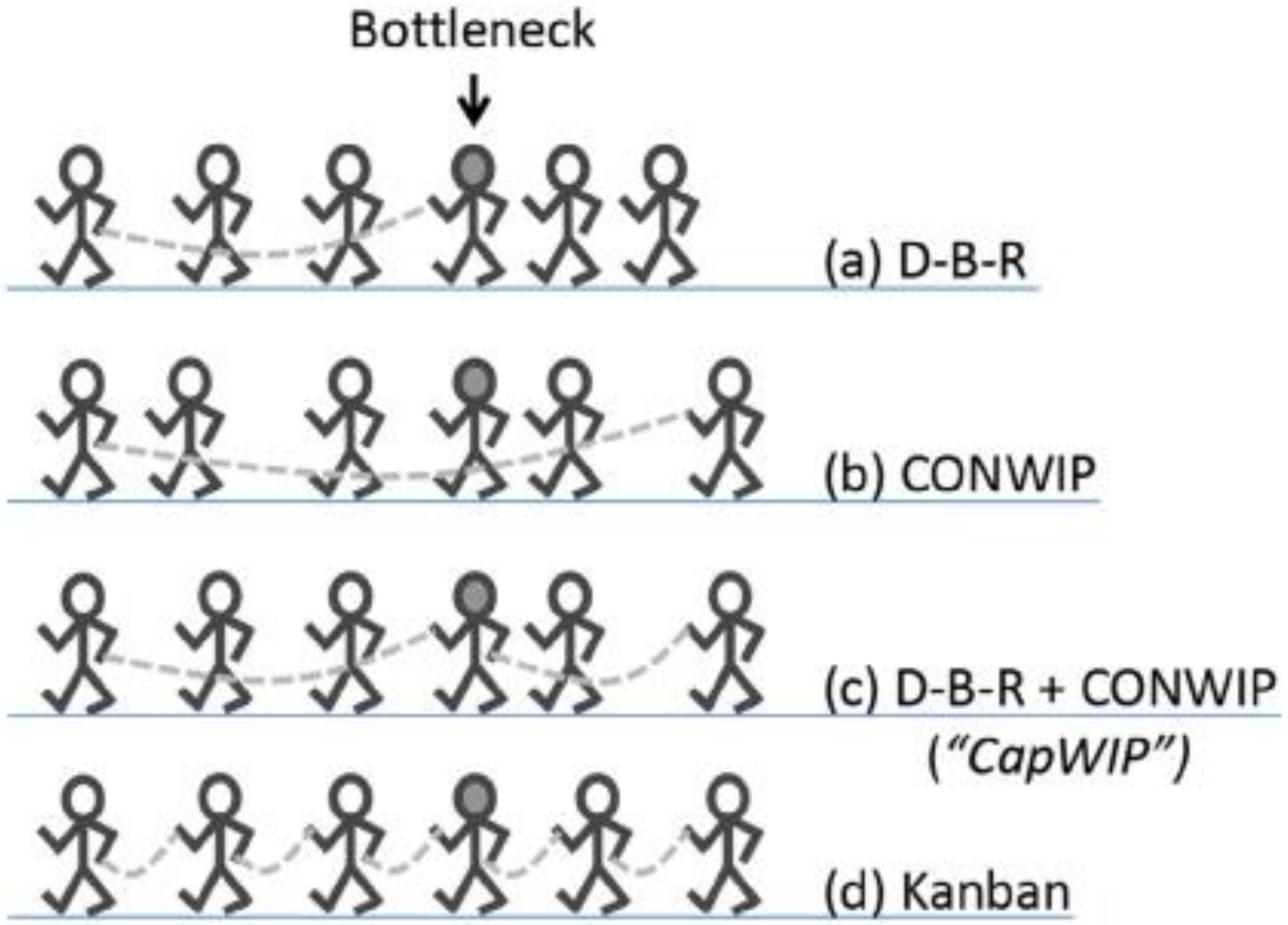
- Kanban is an empirical process
  - All WIP limits can be adjusted empirically
- Bottlenecks should be buffered
- Buffer sizes should be as small as possible, but large enough to ensure optimal performance in the bottleneck



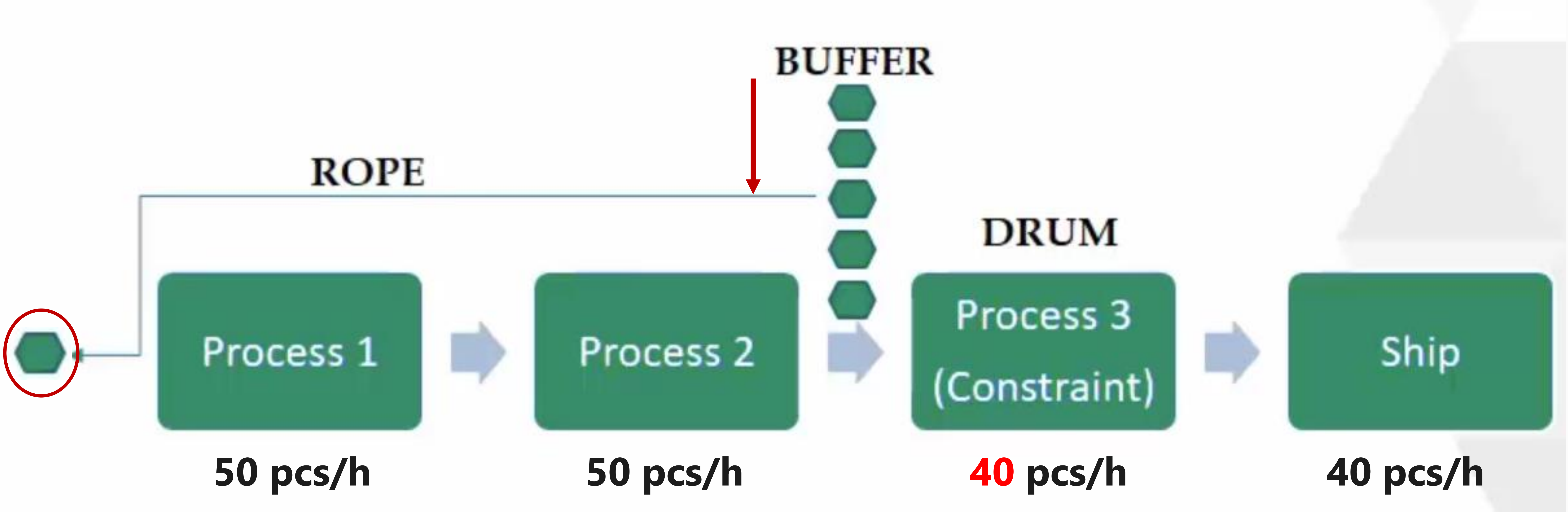
4. Setting Work-in-Progress Limits

(a) Drum – Buffer – Rope

- Drum – the constraint / sets the pace
- Buffer – level of inventory needed to maintain consistent production
- Rope – signal generated from the buffer to release material (indicates an amount of inventory was consumed)



4. Setting Work-in-Progress Limits



BUFFER

ROPE

DRUM

Process 1

Process 2

Process 3 (Constraint)

Ship

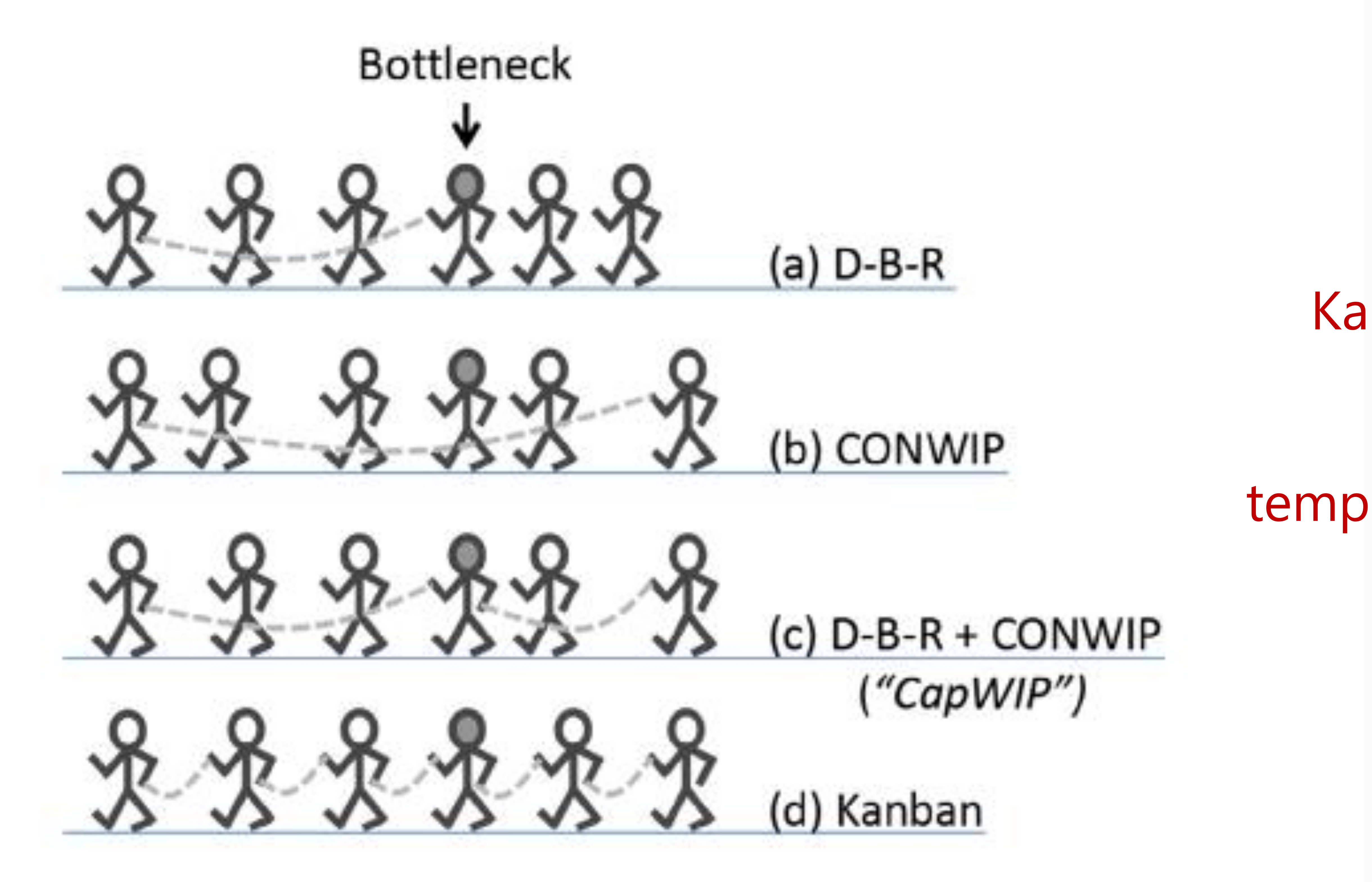
50 pcs/h

50 pcs/h

40 pcs/h

40 pcs/h

4. Setting Work-in-Progress Limits



Kanban deals well

with

temporary bottlenecks!

4. Setting Work-in-Progress Limits

“Excessive time should not be wasted trying to determine the perfect WIP limit; simply pick a number that is close enough, and make progress. Empirically adjust if necessary.”

David J. Anderson

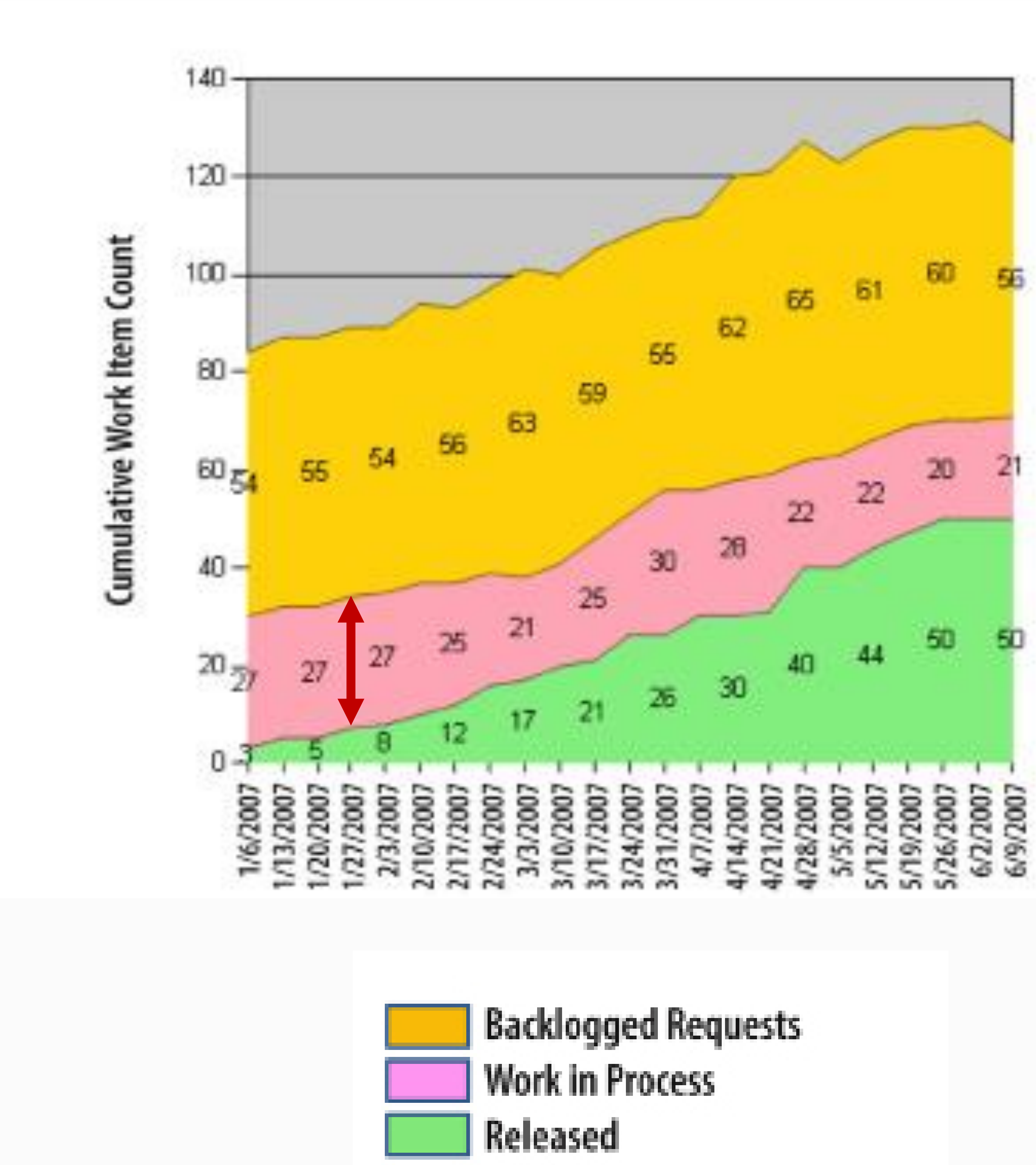
5. Metrics & Management Reporting

- Kanban changes the way the team interacts with its partners

- Reports used to show that
  - Project is on-time and a specific plan is followed – **NO!**
  - System is predictable and is operating as designed – YES!

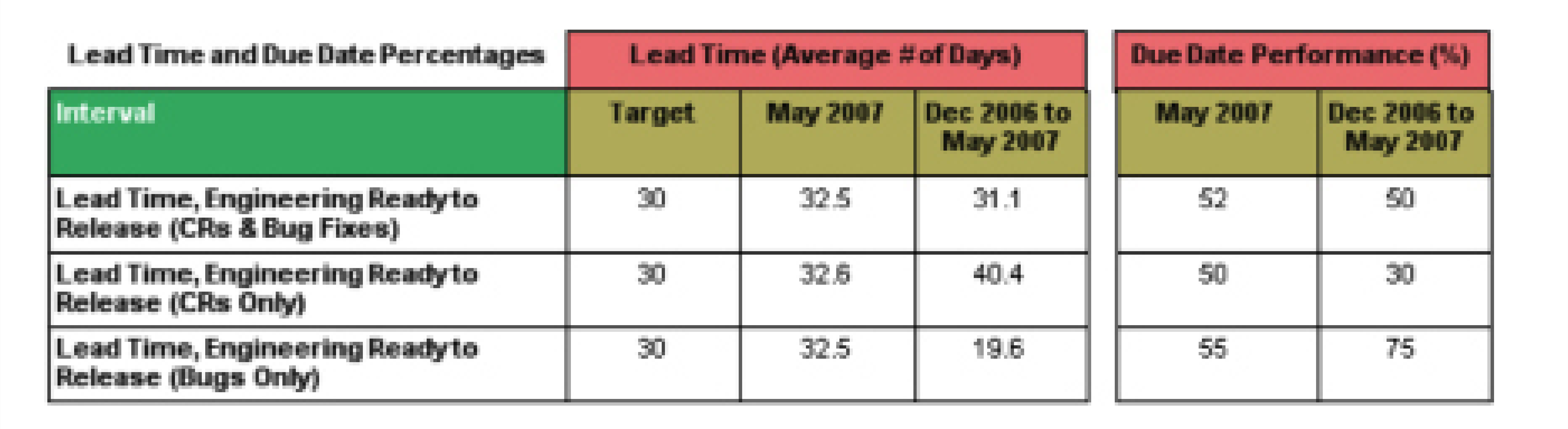
Work In Progress

If the Kanban system is flowing correctly, the bands on the chart should be smooth and their height should be stable.



Source: David J. Anderson - Kanban: Successful Evolutionary Change for Your Technology Business

Lead Time

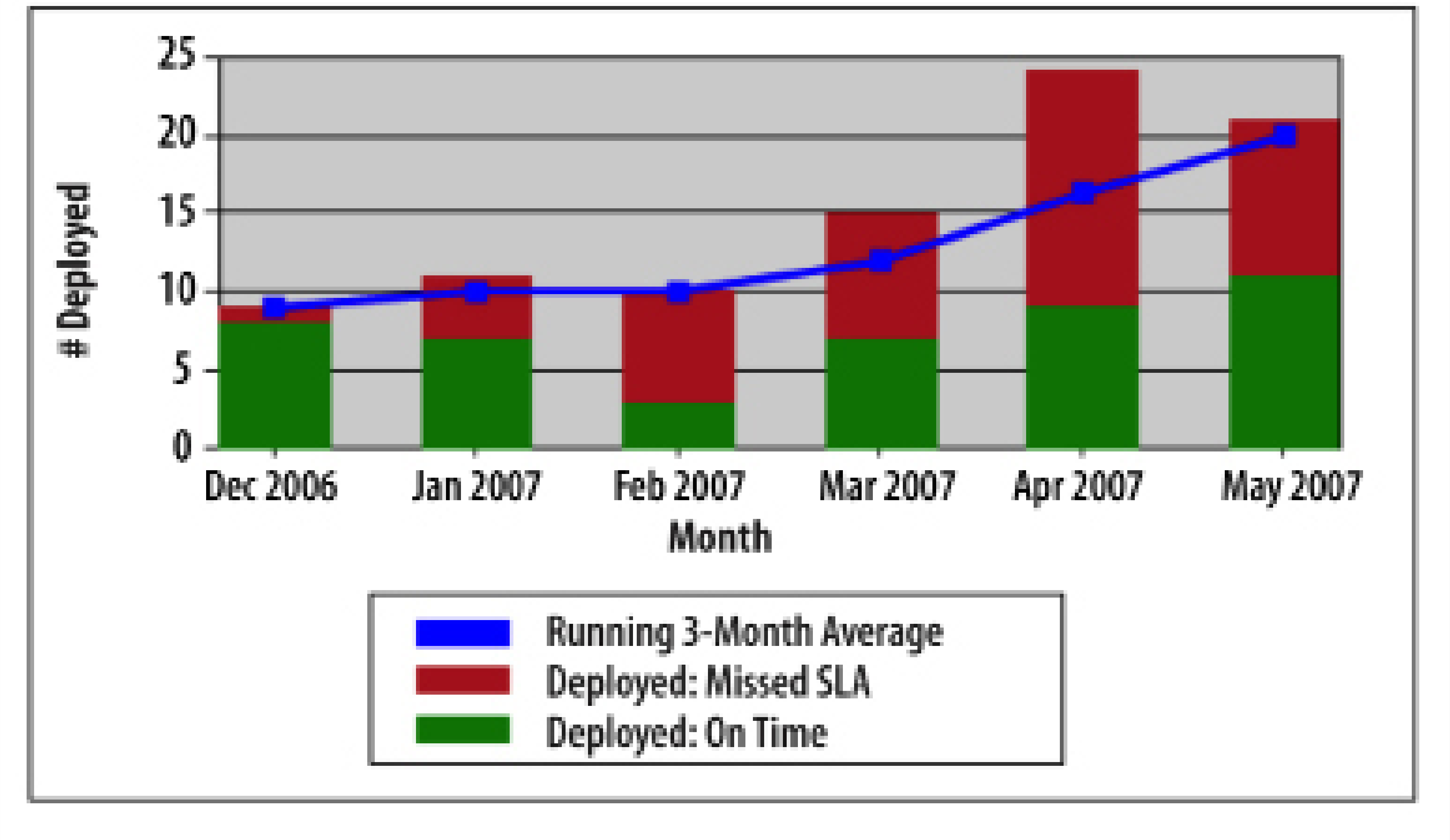


Lead Time and Due Date Percentages	Lead Time (Average # of Days)			Due Date Performance (%)	
Interval	Target	May 2007	Dec 2006 to May 2007	May 2007	Dec 2006 to May 2007
Lead Time, Engineering Ready to Release (CRs & Bug Fixes)	30	32.5	31.1	52	50
Lead Time, Engineering Ready to Release (CRs Only)	30	32.6	40.4	50	30
Lead Time, Engineering Ready to Release (Bugs Only)	30	32.5	19.6	55	75

Source: David J. Anderson - Kanban: Successful Evolutionary Change for Your Technology Business

Throughput

Number of items delivered in a given period of time



Source: David J. Anderson - Kanban: Successful Evolutionary Change for Your Technology Business