## Exercise – Sources of Delay (Part 1 of 2)

10 min

- Gather 5-10 sticky notes: Where in your organization does work wait (queue), and for how long?
- Cluster these sticky notes based on the source / cause for the delay! Do dependencies cause delay? Which type? Internal to your service, external to your service but internal to your business, external to your business?
- Is the length of the delay predictable or not? What is a typical lead time distribution?

Kanban University

**Delay**

## Exercise – Sources of Delay (Part 2 of 2)

10 min

- On more sticky notes, write down any actions available to avoid or influence these different delays.
- Does avoiding delay change the risk in delivering the work?
- Can you imagine reducing the size of queues and length of queuing time? If so, what are the implications?
- Can you imagine circumstances where avoiding dependencies that cause delay would be acceptable?

Kanban University

Lower WIP
Less context Shifting
Less Stress
More predictability
Happier Team Members

Projects → Waiting for Budget Approval (several weeks) | Waiting for Architecture/ Cybesecurity approval (4 weeks) [P] | Waiting for Vendor SoW (several weeks) | Devs waiting while changes are going through NPE (5 days) [P] | Business Signoff of changes (2-3 weeks) | Project closeout activities (1-2 weeks)

Change how we approve budgets, so that we include level funding
WIP Limits
Class of Service discussions

Get Architecture and Cyber closer to the work - implicit approvals
WIP Limits

Nature of Skills can be a bottleneck - Build teams rather than fixed scope of work
Use data to generate SLAs

Reduce frequency of releases, more automation in release processes

Product development processes to demo features and not seek approvals

Deliver work as enduring product teams rather than individual projects

- Issue in dev environment
- 1 day to months
- Unpredictable

Management escalation | External

- Info required from production environment
- 1 - 2 day
- Predictable

No change required (optimal) | External

- Change required in production environment
- 1 day - 2 weeks
- Predictable

Limited by the existing change process | External

- Information required from other dev teams
- 1 day - weeks
- Unpredictable

Agressive chasing, management escalation | External

- Task switching to accomodate urgent task
- 1 day - months
- Unpredictable

Separate resource for unplanned work. Monitor volume to ensure it doesn't become a lot of unplanned work | Internal

Sources of Delay

## Exercise – Dependencies (1 of 2)

10 min

1. On what service(s) do you depend, i.e. which ones do you call?
2. How do you identify in good time which work items will have dependencies on external services?
3. Can you predict when you will demand capacity on dependent services?
4. Where are you dependent on your customer(s)?

Kanban University

**Dependencies**

1) Travel Help Desk
2) Response is before I need to book
3) Most of the time
4)

- Peeks and troughs
- blocked demand is escalated
- waiting demand is based on
- Peers
- Don't think so
- Probably based on Hierarchy, but ideally not

Application Feature Implementation Dependencies
1. API implementation to be done by another scrum team
2. During design phase.
3. Yes as soon as possible i.e. dependencies must be communicated straight away and sooner the work is done better it is for us.
4. Upstream when enhancement request is lodged. Downstream after production release for feedback and bugs reports.

1. Blocking/waiting demand -> expedite, prioritize
2. Multiple sources of demand: Other scrum teams, tech support, management, internal
3. Classes of service: story, defect, general work item, development assistance request

2. Architecture
3. Security
4. Vendors
5. PMO
6. Resource Managers

- Early high level planning
- Domain knowledge
- Historical projects

- Business case endorsement
- Solution acceptance
- On-going project engagement (updates, testing)

## Exercise – Dependencies (2 of 2)

10 min

**Walk in the dependent service(s)' shoes!**

- How does demand look from that perspective?
  - How do they manage demand that is blocking or waiting?
  - Do they have multiple sources of demand?
  - Do they offer different classes of service?

Kanban University

- Expedite/prioritise high priority items
- multiple sources - projects, BAU teams, unforeseen issues or product releases
- CoS - not consciously but they act on priority when needed or via escalations

Dependencies

## Bottlenecks in day-to-day life

10 min

- As a group, take 2 minutes to collect as many "bottleneck situations" you know from your day-to-day life as you can. Put them as stickies on the virtual whiteboard.
- Select some of the examples, and add information about:
  - The cause of the bottleneck
  - The impact on you
  - How it has been dealt with / How it could have been dealt with.

Kanban University

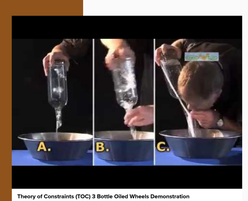
**Bottlenecks**

Traffic Jam | Slow queue at stores | Full public transport | Leaving the office | Slow elevator | Red light | Slow internet connection

Getting an Electrician in to fix something | The double doors entrance to Wimbledon | Applying for a new Passport

Coffee ordering at 10am in our coffee shop → Cause: everyone having 10am as their coffee break/ coffee meeting → Impact: Delays, makes me go to another coffee shop → Dealing: more baristas, allow pre-ordering

Motorway on-ramp → Cause: rush hour on ramp lights → Impact: queueing off the motorway → Dealing: incentives to use motorway at off-peak hours




**Theory of Constraints (TOC) 3 Bottle Oiled Wheels Demonstration**

Practical demonstration of how the Theory of Constraints (TOC) can help you to improve your business. Three identical bottles of water flowing at different r...

Youtube

Bottlenecks

## The Five Focusing Steps

10 min

- In your groups, select a bottleneck situation from private or work lives.
- What is the typical response to this situation?
- Now, apply the 5 focusing steps to it.
- What insights did you gain? Did you discover other options you did not think about at first?

Kanban University

- The Five Focusing Steps:
  - Identify the system's constraint(s).
  - Decide how to exploit the system's constraint(s).
  - Subordinate everything else to the above decision(s).
  - Elevate the system's constraint(s).
  - Start all over!

Identify = find | Exploit = Optimize | Subordinate = coordinate, collaborate, cultivate | Elevate = Upgrade

Five Focusing Steps

What improvements can you think of that can shape the demand or improve delivery capability?

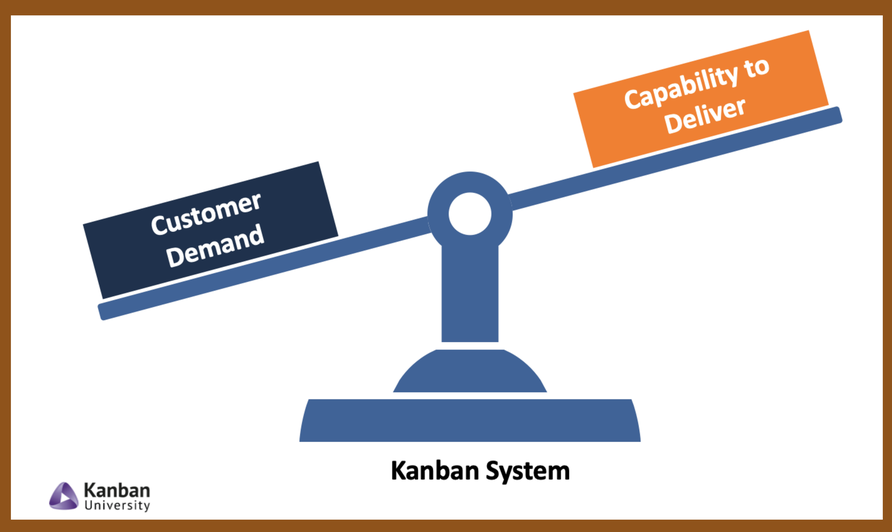
Reduce total number of incoming requests

Categories | Allocation | Work Types

Reduce scope of customers "e.g. only consider high profit customres

Order, sequence, select | Class of Service to prioritise | Triage

introduce WIP Limits | increase delivery capability




Demand and Capability