Your Teams Aren’t the Problem: Your System Is

Why most software leaders optimise the wrong things

Simon M White

Chair, TOCICO
Senior Program Manager, WiseTech Global

Done Statement:

Helping others think more clearly about systems

&

Helping others live better lives

The delivery problem

Every team is improving. Delivery is still too slow.

How can both be true?

The system we are trying to improve



The system we are trying to improve



Goal: useful, reliable changes available to customers

Useful + useable = used. Delivered features are not the same as used features.

A simple delivery model

Stage	Capacity per week
Development	12 comparable changes
Verification	**6 comparable changes**
Release	10 comparable changes

Maximum completed delivery: 6 changes per week

More development capacity

Per week	Before	After
Development output	12	20
Verification capacity	6	6
Maximum completed delivery	**6**	**6**
Queue growth before verification	**6**	**14**

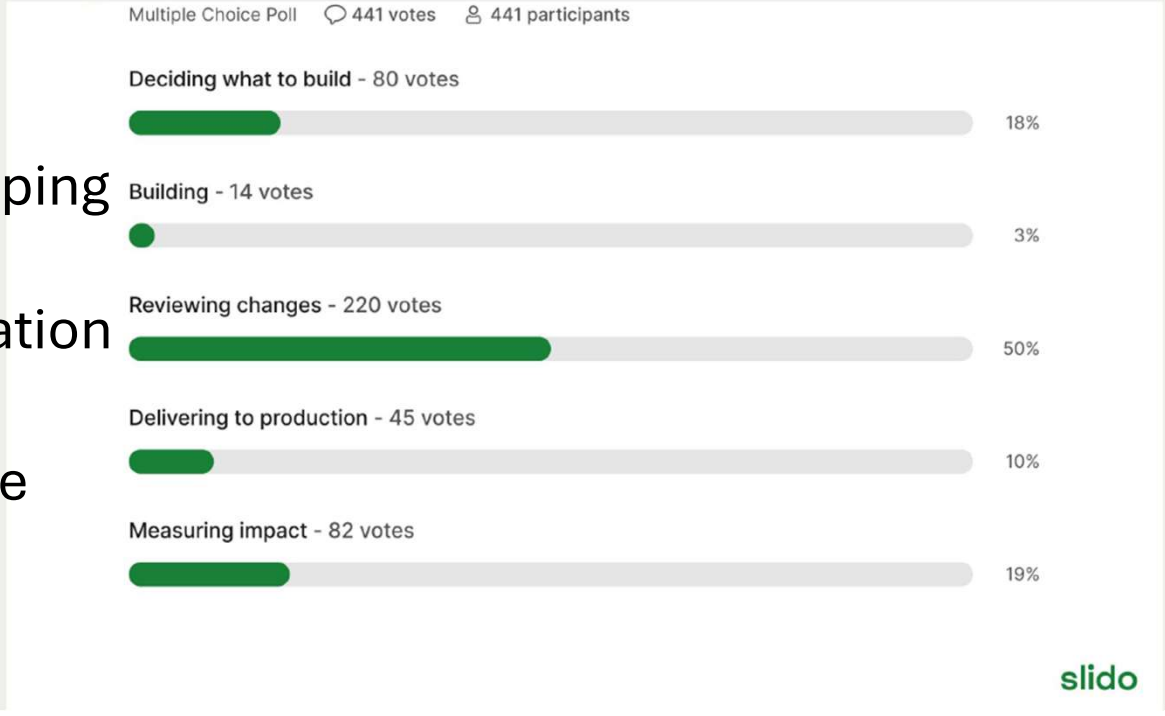
Same model. Development runs at capacity. Release has enough capacity.

Welcome,
builders



Which of these is the biggest ~~friction point~~ bottleneck in your SDLC today?

DORA



Developing

Verification

Release

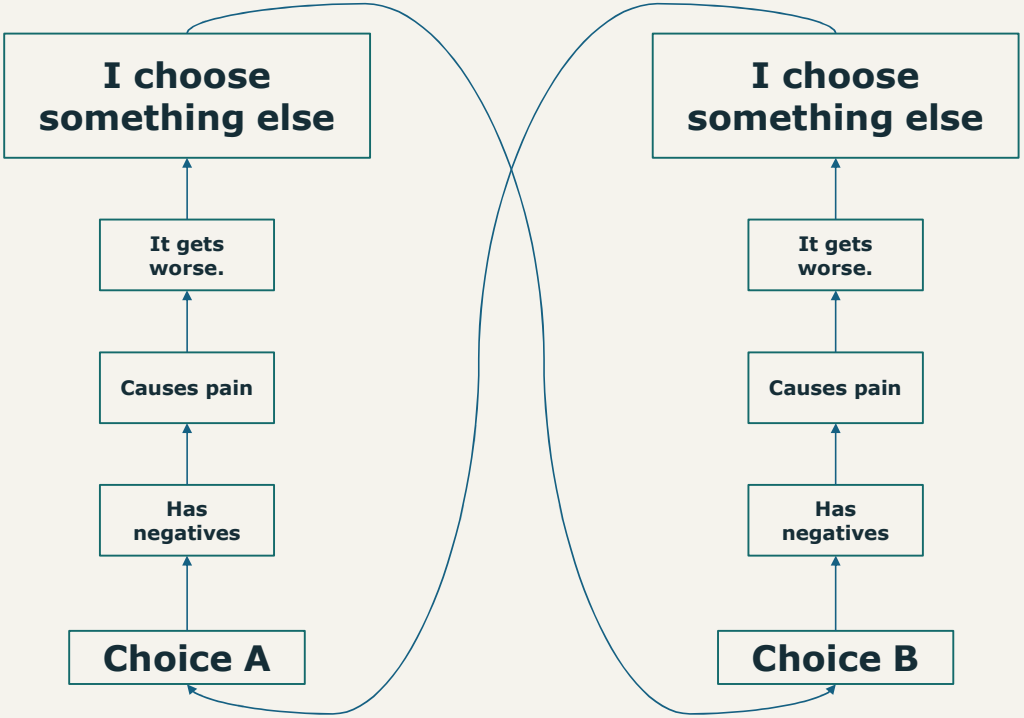
Thinking clearly *about* the problem you are solving

What *kind* of problem am I trying to solve?

A recurring choice problem?

A significant obstacle?

Recurring choice problem

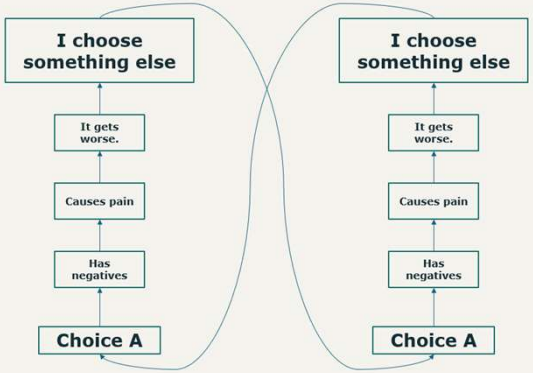


The core conflict



Clarity in problem solving
=
knowing the *kind* of problem you are dealing with

Recurring choice problem



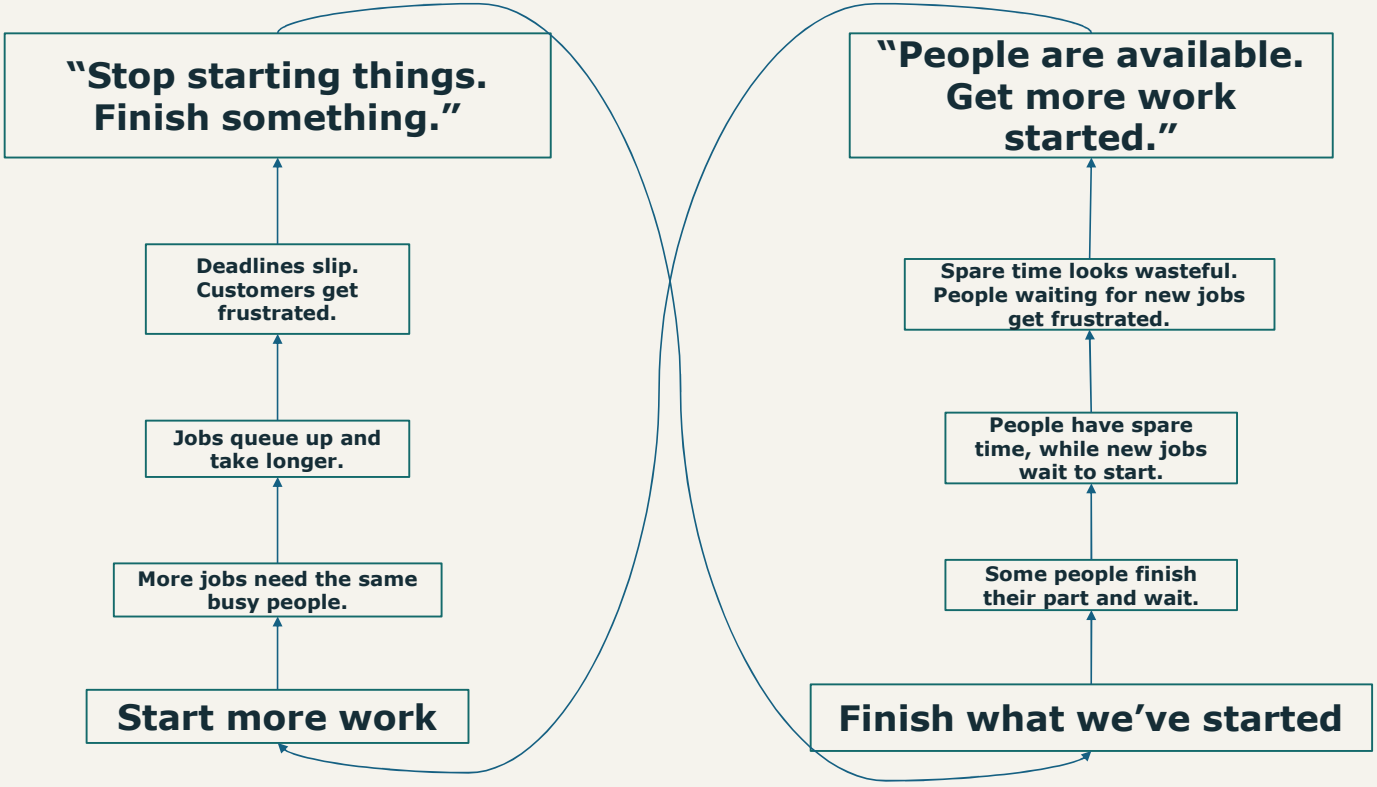
The core conflict



How do I know when to use either?

The recurring choice shows a pattern	The core conflict explains what drives it.
We keep **switching**	We feel forced to **choose**
Repeated swings between opposite **fixes**	Two valid needs seem **incompatible**
Each fix creates the opposite **problem**	Either choice **sacrifices** something important
Growing pain triggers another **reversal**	People **defend** opposing necessary actions

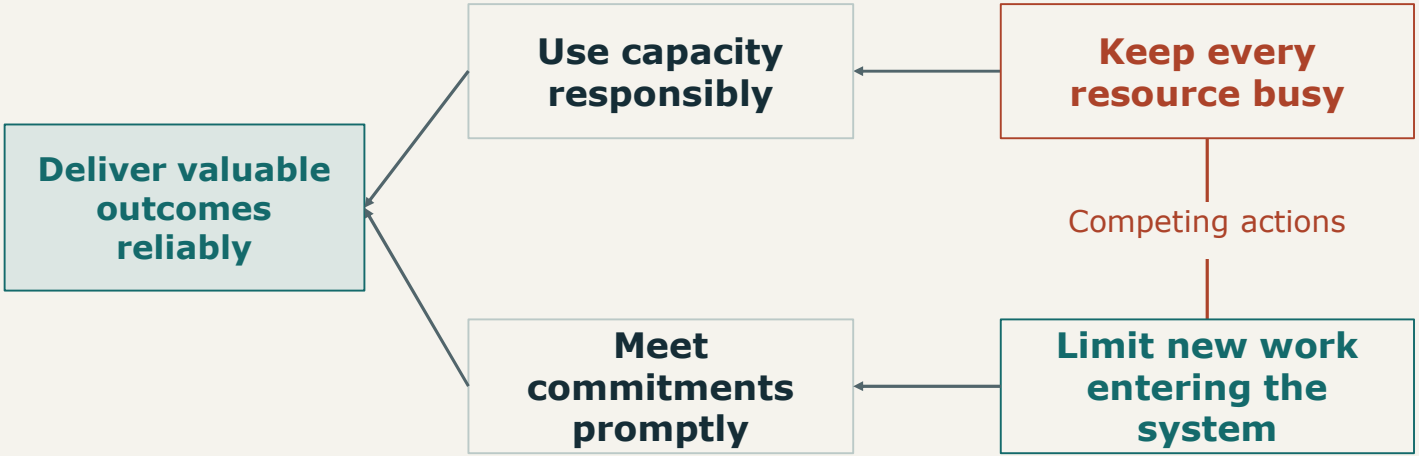
Back to our recurring choice problem



Urgency can make the loop worse

Urgent arrival	A new request displaces committed work
Hidden trade-off	The displaced commitment keeps its original promise
Next escalation	The older commitment becomes urgent too

The core conflict driving *our* recurring problem



The assumption behind the conflict

“If a team has spare capacity, starting more work must improve performance.”

Only true if… that work helps the whole system deliver more.

A different upstream operating *rule*

Start work at a pace the system (ONLY at its' bottleneck) can tolerate.

How can spare development capacity support verification completion, absorb variation and reduce risk?

A queue is evidence, not a verdict

Observation	What to investigate
Work ages before verification	Rework, missing evidence, capacity
Approval waits for a meeting	Decision policy and available authority
Several teams wait for one expert	Shared dependency and competing demand

What observations can you make about the biggest queue in your own system?

The limit may sit outside the team

Capacity	A scarce skill, environment or verification operation
Policy	A decision or approval rule that delays ready work
Dependency	A shared component or specialist many teams need

The Five Focusing Steps

Step	Leadership question
1 Identify	What currently limits completed delivery?
2 Exploit	How can we use that capacity better?
3 Subordinate	What must the rest of the system change?
4 Elevate	What additional capacity or change is needed?
5 Repeat	What now limits delivery? Which rules must change?

1 Identify the current limit (bottleneck)

What change would allow more valuable work to finish?

Trace recent work. Locate lost capacity due to waiting and rework.

Check whether the next stages could absorb more.

2 Use verification capacity well

Ready inputs	Complete evidence before scarce review time begins
Useful work	Move avoidable work away from the scarce skill
Protected time	Make urgent exceptions and their trade-offs explicit

3 Align the rest of the system



Enough "ready" work. Explicit priorities. Support for the constrained operation.

Local measures must support these decisions

4 Increase the limiting capacity

Capacity per week	Before	After
Development	20	20
Verification	6	12
Release	10	10
Potential completed delivery	**6**	**10**

5 Reassess the system

The next limit is release.

The operating rules must follow the current constraint.

Three familiar interventions

Intervention	The question before approval
More people	Which limiting capacity will they increase?
A reorganisation	Which dependency or decision delay will change?
Tools or AI	Will the extra output finish, or join a queue?

A credible improvement proposal explains the effect on the whole system

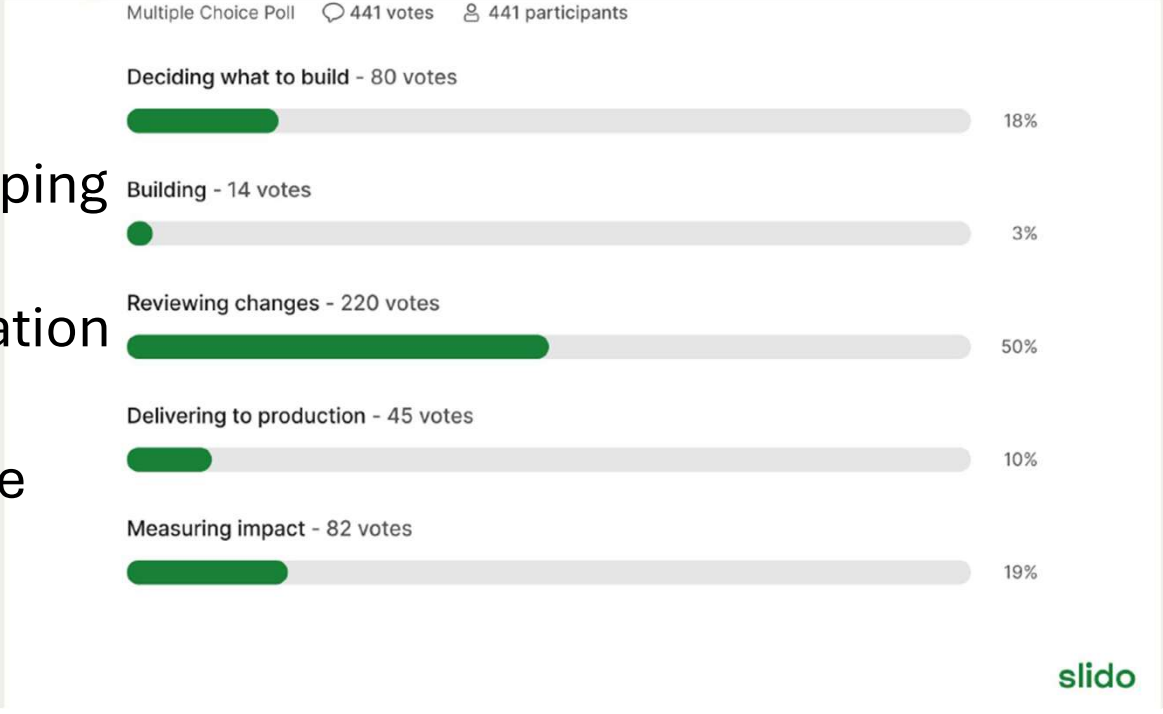
Which of these is the biggest friction point in your SDLC today?

DORA

Developing

Verification

Release



A minute with your own system

Choose one delayed commitment.

What useful outcome was it meant to produce?

Where did it wait, and what was it waiting for?

What evidence points to the current limit?

What change could you test?

The typical Theory of Constraints process

Full diagnostic

Visualise the system and define both The Goal and Throughput

Gather ‘where does it hurt symptoms’

Map out the Current Reality

Break the core conflict by hacking assumptions

Map out Future Reality

Map prerequisites and Necessary Conditions

Address the Yes Buts

Remove the obstacles

Trim Negative Branches

The simplified model, for you to try

Get a simple drawing of the system.

Write one sentence on The Goal of the system.

Look for the biggest queue.

Use only observable evidence to establish the reason for the queue.

Understand what kind of problem you are trying to resolve.

Break just one arrow/connection.

Unload the bottleneck, by reducing the incoming work, at zero cost.

Then, and ***only*** then, invest.

Station X43 vs Challenger Tank

One operation.
Consequences across the system.

One went well.
One didn’t.

Done Statement:

I can think more clearly about systems

&

I can help others live better lives

What are we improving, and what will it allow the whole system to finish?

Thank you. Questions?

Simon M White

e. simon@simonmwhite.au

www.linkedin.com/in/simonmwhite-au