The Mechanics of Scaling

Maryia Tarpachova

Senior Software Engineer, Octopus Electroverse

The Mechanics of Scaling

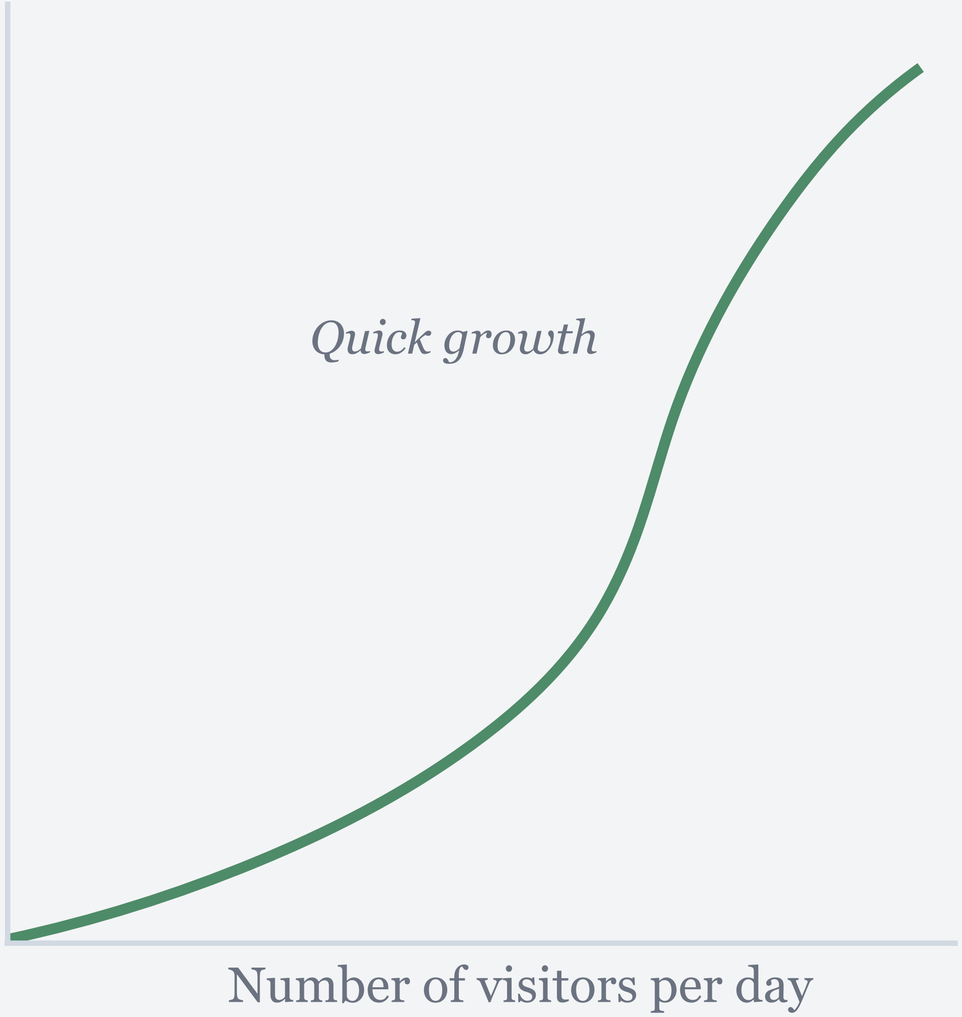
Maryia Tarpachova

Senior Software Engineer, Octopus Electroverse

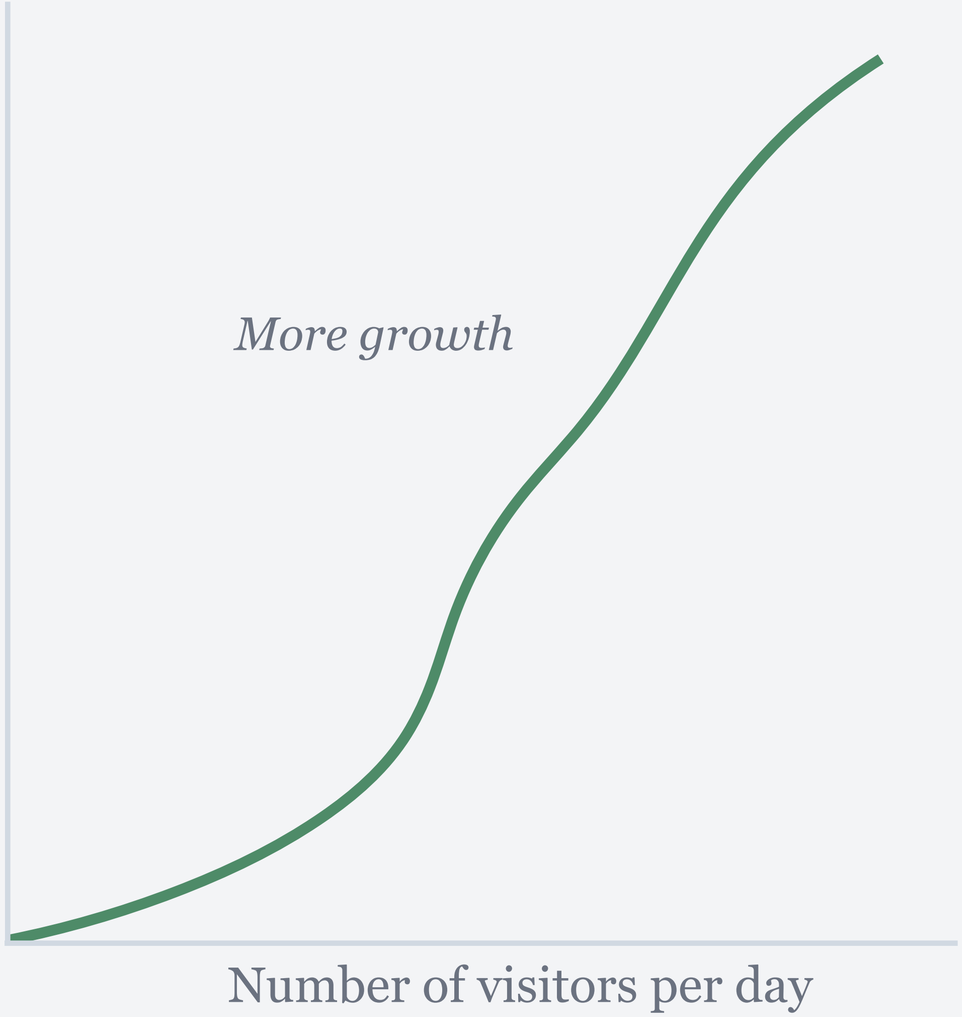
Systems Thinking

Systems with similar structures
produce similar behaviours

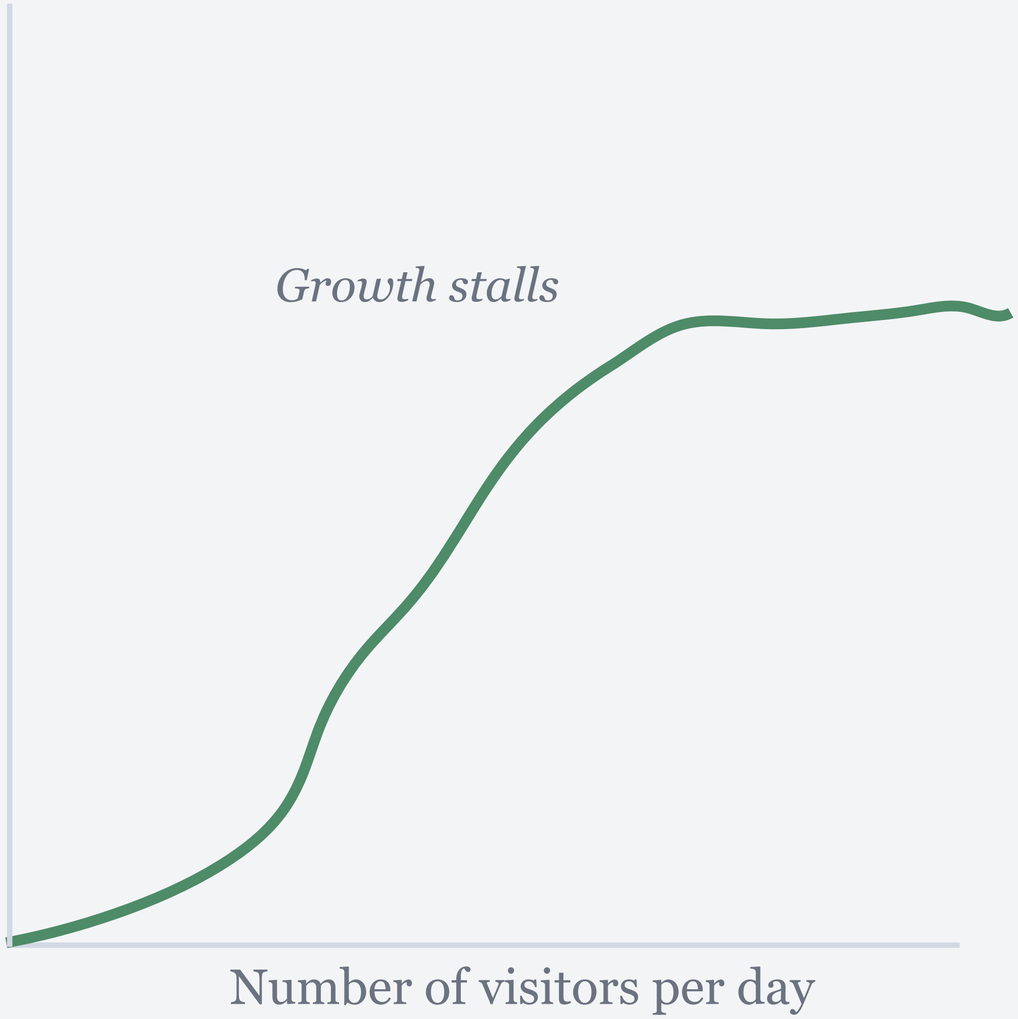
A park opens...



More rides added...



Arcade added...



Better ads?

More promotions?

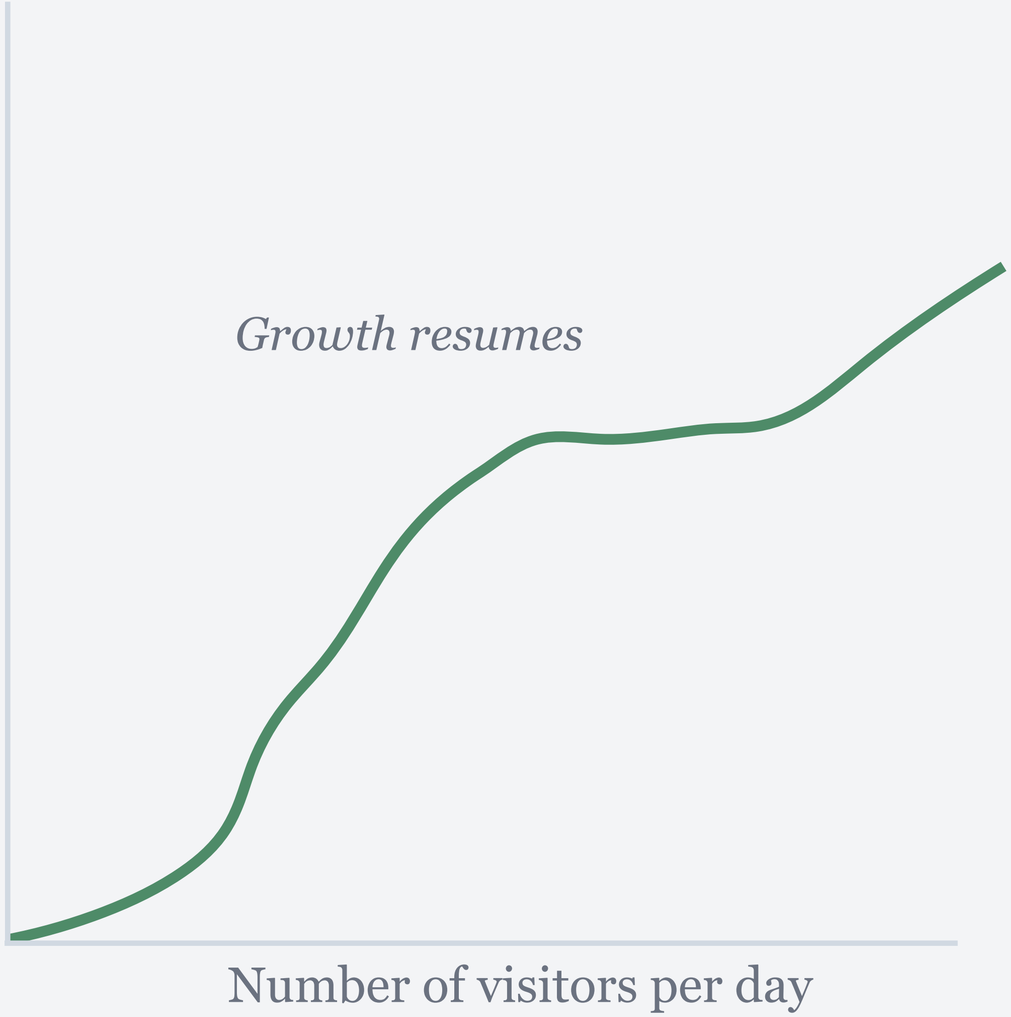
Even more attractions?


600 m

The challenge is not creating demand.

It's handling it.

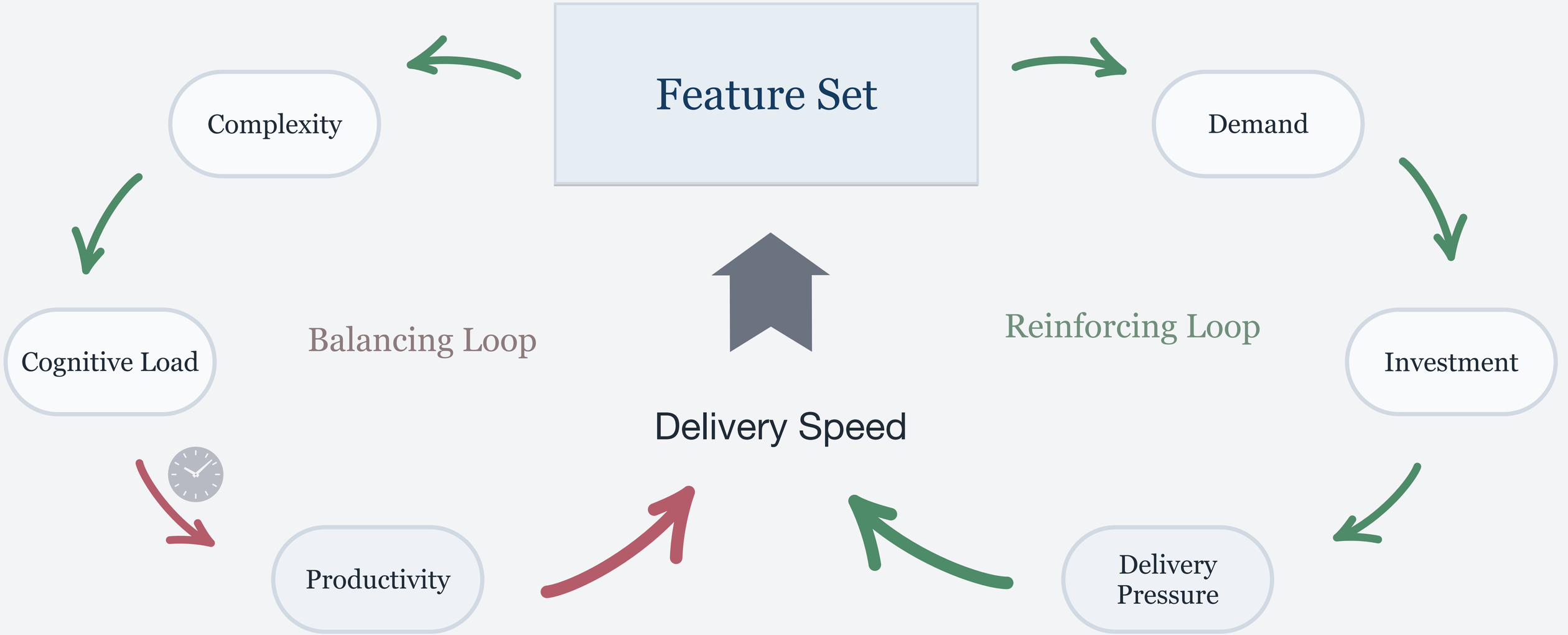
Bus service added...



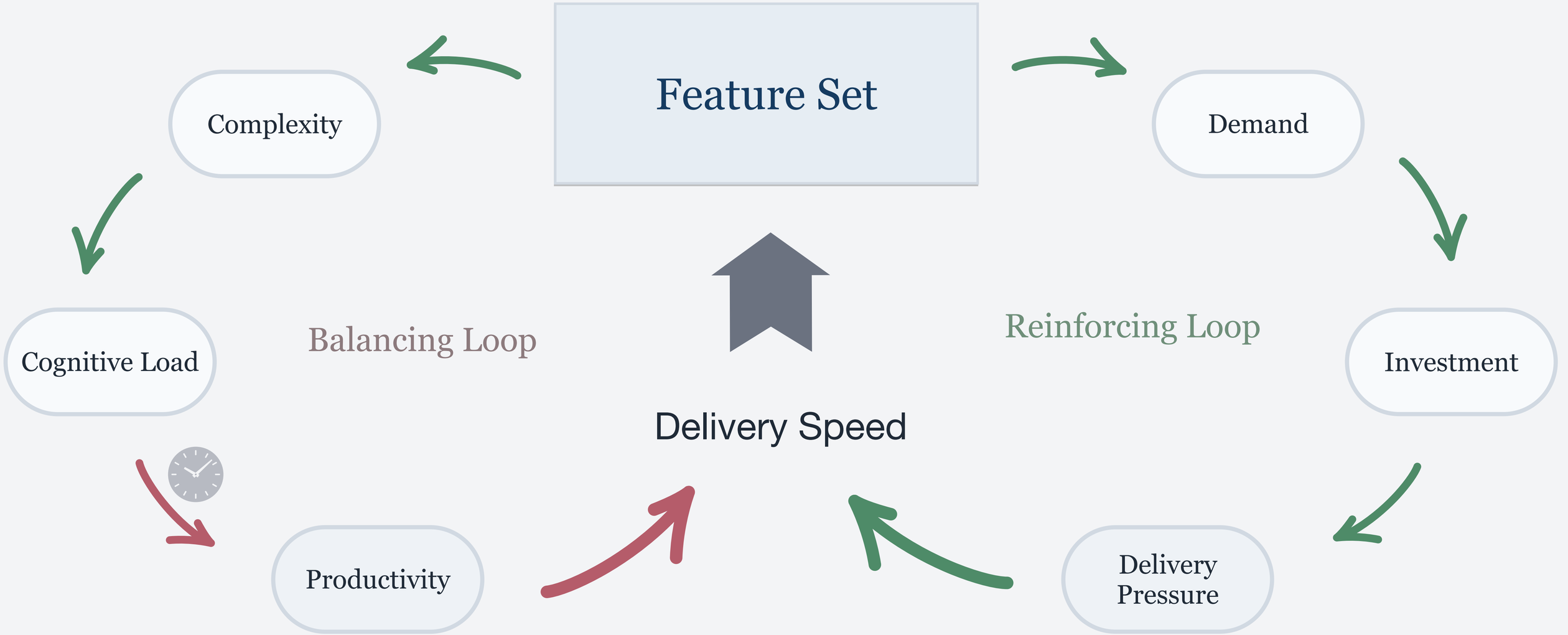
Limits to Growth

Growth creates conditions that limit further growth

Growth is never isolated



Growth is never isolated



The leverage is

in redesigning the system around the constraint

How do you tame complexity?

Make the things that fit in your head the important ones

Clean abstractions

Clear dependencies

Shared language

Strong conventions

Protect your team's ability to reason about the system

Appropriate domain size

Reliable documentation

Stable ownership

Clear boundaries

Don’t normalise accumulated complexity

Acknowledge when it’s hard

Celebrate work that reduces complexity

Create space for system improvements

Continued success
means constantly unlocking
new limits to growth

How do you navigate other limits to growth?

Constraints are scaling alongside your product.

Watch for where the pressure is building.

Team burnout

Dependencies

Knowledge silos

Coordination load

Things will look fine… right until they don’t.

Watch for increasing strain behind stable KPIs.

The best time to respond to a limit is before it becomes a crisis.

Set clear thresholds. Act the moment they’re crossed.

Error budgets

Recovery time > X

Change failure rate > X

If performance drops without a clear reason,

you have likely hit a limit.

Stop accelerating.

Find the constraint.

Redesign the system.