The Shadow Culture

Why engineering principles fail under load

Manogna Machiraju

New role. New company.

You want to build something. Of course both. It's always both.

WHAT I HEARD	WHAT I INFERRED
The code is archaic. Pipelines broken. Security is hardly a thought	*Engineering was always catching up and busy executing*
Delivery pressure, Product Fn doesn't care about Platform.	*No shared principles. Everyone optimising for different aspects.*

People had intentions to improve!

The Why, The What & The How

The Why	The What	The How
Engineering as equal partner agreed not just stated	**The Principles built with people in the trenches**	**Tech Strategy. Get the right skills & autonomy.**

Fix all three, and the culture follows.

Everyone agreed. Nothing changed.

Hoped to See	WHAT WAS ACTUALLY HAPPENING
Engineering referencing to Principles in Decisions	*Understood but not using them*
Behaviours changing, Incidents Reducing and Platform Quality Improving, TTM shrinks by 10x :)	*Same Incidents, Same Shortcuts & Same Tech Debt, Same Concerns from Product*

But Why?

Maybe the teams don't care.

Maybe it's not translating.

Maybe I am just an ineffective leader.

Quest begins. With ONE Focus. ONE Intent

They were quietly building a Culture

Shadow Culture naturally emerges in the environment we operate in.

A Failure of System Design.

Not a lack of Belief. Not a lack of Competence.

Right principles is only half the job

> A principle told what to value.
>
> But they didn’t answer 'by doing what?'

Specific Behaviours

Stress test them

Culture is a Product

Stress test behaviours like features

Value	**Viability**	**Usability**	**Feasibility**
Does it reduce risk & improve outcomes?	Does our business situation allow this?	Is it easier than the shortcut?	Do we have what it takes to do this?

The lens failed tells you what to fix

Value

Behaviour happens. Outcomes don't

→ **Review if the outcome is important**

Viability

Business situation doesn't allow it

→ **Make bold subtractions.**

Usability

The path exists; just difficult.

→**Redesign the Path; Remove Friction**

Feasibility

we literally don't have the capability

→ **Invest in the Tooling, Frameworks.**

We are here to Build

"Block your calendar DeepWork. Decline all meetings."

Value:
✓ Deep work produces what meetings never can.

Viability
✗ Death by a thousand reasonable exceptions.

Usability
✓ Calendar block is simple.

Feasibility
✓ Anyone can block a calendar day.

Meeting Observability - Async before Sync

We know it before the customer does.

Instrument so we recover from the customer issues in 10 minutes.

Value
✓ Quicker TTR - Lesser frustration

Viability
✗ Can’t survive slow time to market

Usability
✗ Takes iterations to get it right

Feasibility
✗ No tooling. No skills & No time

- Made Observability THE priority for platform for a quarter
- Accepted that we only can focus on Critical Journeys

Same principles. Different weight.

	Built to be maintained	Done beats perfect
Core team Existing proposition. Real customers.	**Stable product as** one bad deploy is revenue loss.	**Move & learn fast** but in a confined blast radius behind flags
Frontier No customers yet. Might not survive the quarter	Any % hardening tax that may not exist in 3 months is a **waste**	**Moving fast is the point** Cost of breaking is near zero.

Two principles. Three Teams. Four Lenses

01 Define

What does living the principles actually look like for your team right now?

02 Stress Test

Run the answers through the 4 lenses

03 Action Plan

Systemic & Team level Commitments with clear Owners

Be Ready for the hard answers & Measure

Culture evolves. Design for it.

Principle past its expiry is its own kind of shadow culture.

Sometimes change the principles.	What mattered at 10 engineers breaks at 100.
Sometimes change the behaviours.	The behaviour needs to catch up.

Evolve the systems to support

So, Why Principles fail under load?

> Good Intentions alone isn’t enough
>
> We need to build a SYSTEM

Building that SYSTEM is OUR JOB.