

# Story Time!

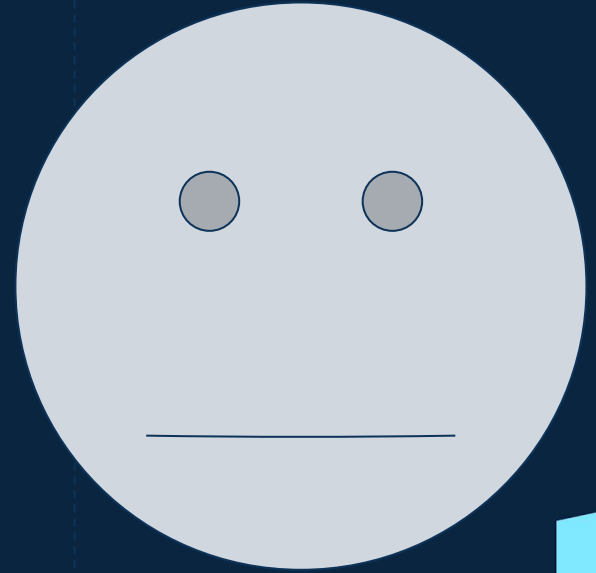


**You started life moving fast but development has slowed down.**











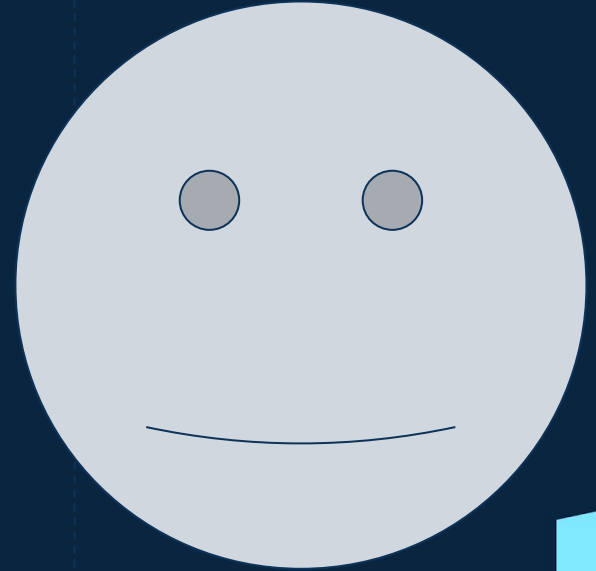


# Six Months Pass

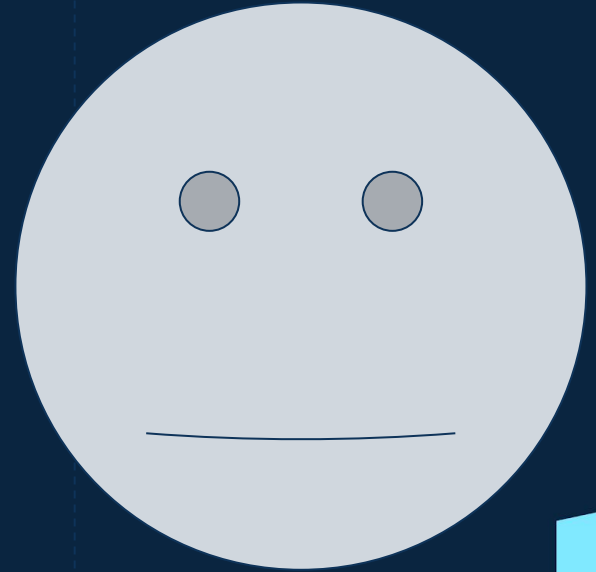


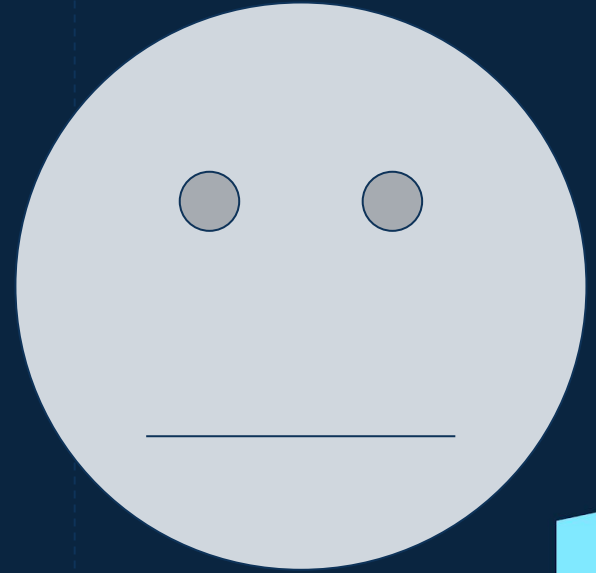
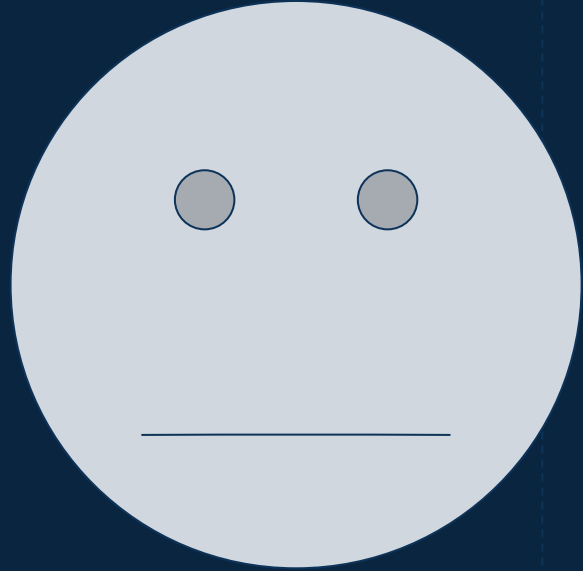
# Six More Months





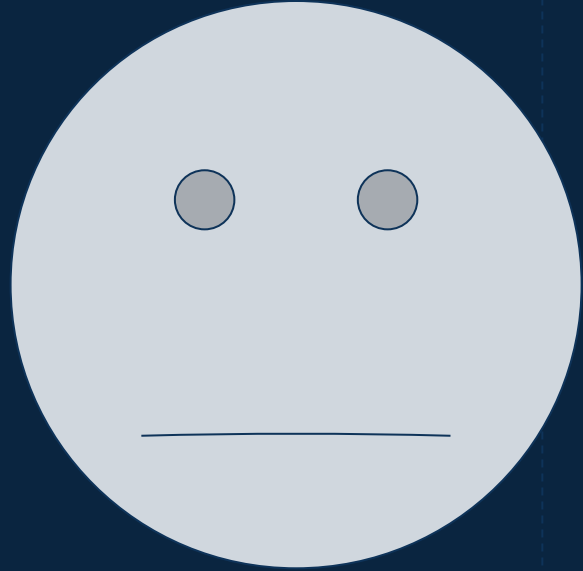
**Six More Months Pass**







**Six More Months Pass**





**We didn't fully achieve our  
goals, why?**

# Confused Technical Vision With Technical Strategy!

stripe

# Technical Vision vs. Technical Strategy

The Difference and Why it Matters

Jonathan Maltz (@maltzj)



**Spoiler Alert:** *It matters because whether you like it or not, you've got a vision and a strategy. The only things that matter are whether they are agreed upon and fit the problem at hand.*

Section 1

# Vision vs. Strategy



# Technical Vision

**Technical Vision:** *An aspirational view of where your system should end up. Likely anchored to a north-star of “here’s what our system will look like in X years”*

# Technical Vision

- 1 Helps tie break decisions

# Technical Vision

- 1 Helps tie break decisions
- 2 Helps get people excited about where you're going

# Technical Vision

- 1 Helps tie break decisions
- 2 Helps get people excited about where you're going
- 3 Helps attract collaborators to your cause

# Technical Vision - Pitfalls

- 1 Not sufficiently compelling

# Technical Vision - Pitfalls

- 1 Not sufficiently compelling
- 2 Doesn't shoot for the right target. It's either too far behind or too far in the future

# Technical Strategy



**Technical Strategy:** *A clear problem statement about what's stopping you from achieving that vision, a guiding policy to fixing that problem, and concrete steps to put that policy into action.*

'Has every chance of becoming a business classic'  
*Management Today*

RICHARD RUMELT

GOOD  
STRATEGY  
BAD  
STRATEGY

The Difference and Why It Matters

# Technical Strategy

- 1 Zeros in on a part of the vision to achieve

# Technical Strategy

- 1 Zeros in on a part of the vision to achieve
- 2 Crafts a way to solve

# Technical Strategy

- 1 Zeros in on a part of the vision to achieve
- 2 Crafts a way to solve
- 3 Implicitly chooses a set of initial non-goals in achieving the vision

# Technical Strategy - Pitfalls

- 1 Failure to face the challenge

# Technical Strategy - Pitfalls

- 1 Failure to face the challenge
- 2 Bad Strategic objectives

# Technical Strategy - Pitfalls

- 1 Failure to face the challenge
- 2 Bad Strategic objectives
- 3 Mistaking strategy for goals



*Technical Vision is about saying “yes” to things.  
Technical Strategy is about saying “no” to things*

# Applying it to our example

Vision: *Serverless compute controlled by terraform*

Vision: *Serverless compute controlled by terraform*



Strategy: 🤔

# Technical Strategy

- 1 Problem Statement: We don't have the whole platform up and running

# Technical Strategy

- 1 Problem Statement: We don't have the whole platform up and running
- 2 Guiding policy: Get an MVP of the platform to show people the future, then onboard them

# Technical Strategy

- 1 Problem Statement: We don't have the whole platform up and running
- 2 Guiding policy: Get an MVP of the platform to show people the future, then onboard them
- 3 Concrete actions: Spend 6 months building an MVP of the platform



# Differentiating the two is hard

# 🤔 How to manage

# Differentiating Vision vs. Strategy

- 1 Create language for the two vision vs. strategy (emojis help!)

# Differentiating Vision vs. Strategy

- 1 Create language for the two vision vs. strategy (emojis help!)
- 2 Create different artifacts when setting vision vs. strategy

# Differentiating Vision vs. Strategy

- 1 Create language for the two vision vs. strategy (emojis help!)
- 2 Create different artifacts when setting vision vs. strategy
- 3 Active listening when you get feedback

*“I hear you saying X. To clarify, are you disagreeing with the vision, or are you concerned we don’t have the right strategy?”*

Section 2

# Some Common Patterns

# Pattern 1: The Grand Migration



# Pattern 1: The Grand Migration

# Pattern 1: The Grand Migration

- 1 Articulate the Vision

# Pattern 1: The Grand Migration

- 1 Articulate the Vision
- 2 Build PoC to prove that a solution works

# Pattern 1: The Grand Migration

- 1 Articulate the Vision
- 2 Build PoC to prove that a solution works
- 3 Migrate everything onto the solution

# Pattern 1: The Grand Migration

This pattern gets poo-pooed a lot

# Pattern 1: The Grand Migration

This pattern gets poo-pooed a lot **but it can work**, within certain constraints!

# Pattern 1: The Grand Migration Constraints

# Pattern 1: The Grand Migration Constraints

- 1 Strong organizational conviction to the solution



# Pattern 1: The Grand Migration Constraints

- 1 Strong organizational conviction to the solution
- 2 Very obvious, clear, and well-aligned value

# Pattern 1: The Grand Migration Constraints

- 1 Strong organizational conviction to the solution
- 2 Very obvious, clear, and well-aligned value
- 3 (Helpful but not strictly necessary) Ability to walk away from tricky bits of legacy implementations.

# Pattern 1: The Grand Migration Failure Modes

# Pattern 1: The Grand Migration Failure Modes

- 1 Most other cases

# Pattern 1: The Grand Migration Examples

# Pattern 1: The Grand Migration Examples

- 1 Khan Academy's migration from Python 2 -> go lang

# Pattern 1: The Grand Migration Examples

- 1 Khan Academy's migration from Python 2 -> go lang
- 2 Stripe's India Data Locality work

# Pattern 1: The Grand Migration Examples

- 1 Khan Academy's migration from Python 2 -> go lang
- 2 Stripe's India Data Locality work
- 3 Many companies' work to get GDPR compliant



# Pattern 1: The Grand Migration Summary

Depends on a very simple strategy to turn a vision into reality

# Pattern 1: The Grand Migration Summary

Seductive for its simplicity. Be very careful in its deployment

# Pattern 2: The No Vision Strategy

# Pattern 2: The No Vision Strategy

# Pattern 2: The No Vision Strategy

- 1 Make a list of all the problems facing the team

# Pattern 2: The No Vision Strategy

- 1 Make a list of all the problems facing the team
- 2 Stack-rank them according to what's the most important

# Pattern 2: The No Vision Strategy

- 1 Make a list of all the problems facing the team
- 2 Stack-rank them according to what's the most important
- 3 Go and execute on the most important ones in a locally optimal way

# Pattern 2: The No Vision Strategy Success Modes



# Pattern 2: The No Vision Strategy Success Modes

- 1 Things are on fire and you need to start putting them out ASAP

# Pattern 2: The No Vision Strategy Success Modes

- 1 Things are on fire and you need to start putting them out ASAP
- 2 You are asked to “create a plan” but don’t actually know enough to create a vision

# Pattern 2: The No Vision Strategy Success Modes

- 1 Things are on fire and you need to start putting them out ASAP
- 2 You are asked to “create a plan” but don’t actually know enough to create a vision
- 3 Situations where your operating horizon is 3 -> 6 months

# Pattern 2: The No Vision Strategy Failure Modes

# Pattern 2: The No Vision Strategy Failure Modes

- 1 The organization wants a Big Vision to solve a problem

# Pattern 2: The No Vision Strategy Failure Modes

- 1 The organization wants a Big Vision to solve a problem
- 2 You need to coordinate a lot of people over a long time.

# Pattern 2: The No Vision Strategy Examples

# Pattern 2: The No Vision Strategy Examples

- 1 Team has too much operational work and needs to dig out



# Pattern 2: The No Vision Strategy Examples

- 1 Team has too much operational work and needs to dig out
- 2 Launching a new thing and figuring out why it's not working

# Pattern 2: The No Vision Strategy Examples

- 1 Team has too much operational work and needs to dig out
- 2 Launching a new thing and figuring out why it's not working
- 3 Starting work in a very uncertain space where you don't have your feet underneath you.

## Pattern 2: The No Vision Strategy Summary

Invests very little time to set vision and instead focuses on crafting + executing on a strategy to solve today's problems

## Pattern 2: The No Vision Strategy Summary

Great for dealing with short-term problems, less sustainable over the long-term.

# Pattern 3: Big Vision with small experiments along the way

# Pattern 3: Big vision with small experiments

# Pattern 3: Big vision with small experiments

- 1 Identify problems facing the team

# Pattern 3: Big vision with small experiments

- 1 Identify problems facing the team
- 2 Make a vision about how to solve the problem. Get rough buy-in that it's directionally right.



# Pattern 3: Big vision with small experiments

- 1 Identify problems facing the team
- 2 Make a vision about how to solve the problem. Get rough buy-in that it's directionally right.
- 3 Determine a vertical slice of value to deliver within that vision. Execute on it for 3 -> 6 months.

# Pattern 3: Big vision with small experiments

- 1 Identify problems facing the team
- 2 Make a vision about how to solve the problem. Get rough buy-in that it's directionally right.
- 3 Determine a vertical slice of value to deliver within that vision. Execute on it for 3 -> 6 months.
- 4 Repeat.

# Pattern 3: Big vision with small experiments success modes

# Pattern 3: Big vision with small experiments success modes

- 1 Big ambiguous problem where you've got a rough idea of how to move forward

# Pattern 3: Big vision with small experiments success modes

- 1 Big ambiguous problem where you've got a rough idea of how to move forward
- 2 Expectations are that your work needs to span  $> 1$  team and  $> 6$  month time-horizons

# Pattern 3: Big vision with small experiments failure modes

# Pattern 3: Big vision with small experiments failure modes

- 1 Team is really prioritizing efficiency of solution.

# Pattern 3: Big vision with small experiments failure modes

- 1 Team is really prioritizing efficiency of solution.
- 2 Can be shortcut if there's an existing vision



## Pattern 3: Big vision with small experiments summary

Spends more time establishing a fuzzy directional vision, but embraces a strategy of building confidence in that over time.

## Pattern 3: Big vision with small experiments summary

A sensible default pattern, but you don't always need the whole thing.

# Summary!

**1. Vision and strategy are  
two things. Create  
language for separating  
them**

**2. Vision is about saying  
yes to things, strategy is  
about saying no to things**

**3. Default to “big vision  
with incremental steps”  
but other options can  
work!**



# Further Reading

- [The Staff Engineer's Path](#)
- [Good Strategy Bad Strategy: The Difference and Why it Matters](#)
- <https://lethain.com/strategies-visions/>
- <https://lethain.com/grand-migration/>



