

# Dressing to the Nines



**Mark Robinson**  
Infrastructure Engineering, Plaid

WE'RE HIRING TOO

# What would you say you do here?



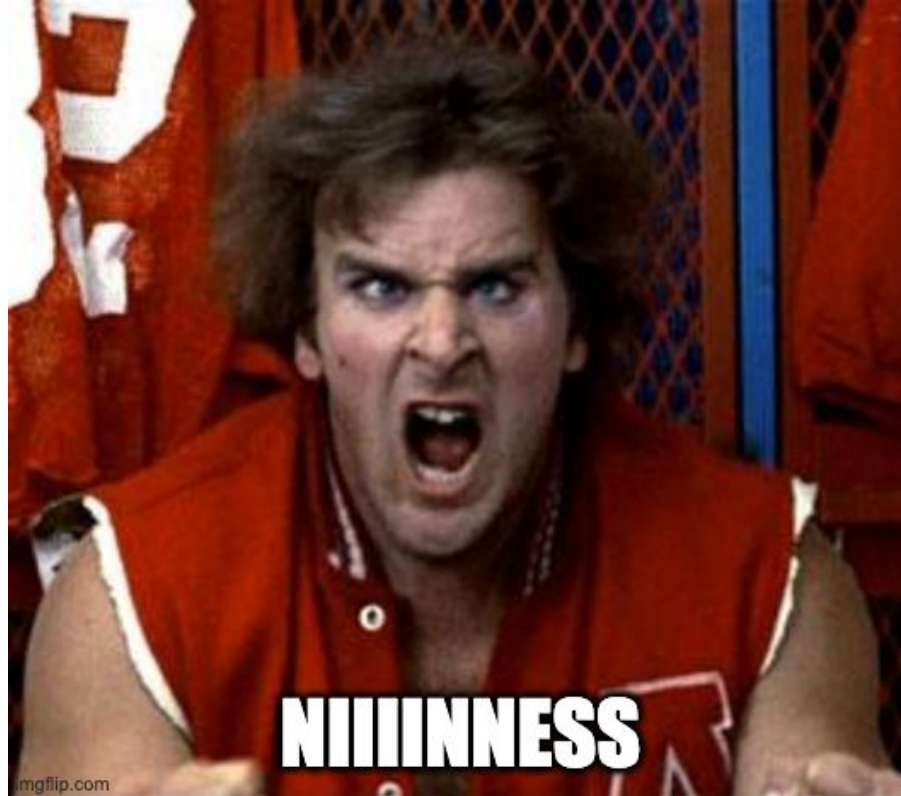


Everybody  
lies (about  
reliability)



WE SHOULD CHANGE THE MARKETING COPY

## How do people talk about reliability?



## The Rule of 9's

- You need one more 9 from your dependencies than your SLA
- This obviously can't scale
  - Doubly so because the power grid is only 99.97%

## Let's talk about bicycles and airplanes

- Bicycles are reliable
- Airplanes are resilient
- Next time, you're on a plane ask for the Tech Log





**Think Resiliency  
Not Reliability**

IT'S 2AM LET ME PLAY YOU THE SONG OF MY PEOPLE

## Why is reliability wrong?

- All systems break
- Reliable systems are repaired
- Resilient systems recover



## Focusing on reliability encourages poor decision making

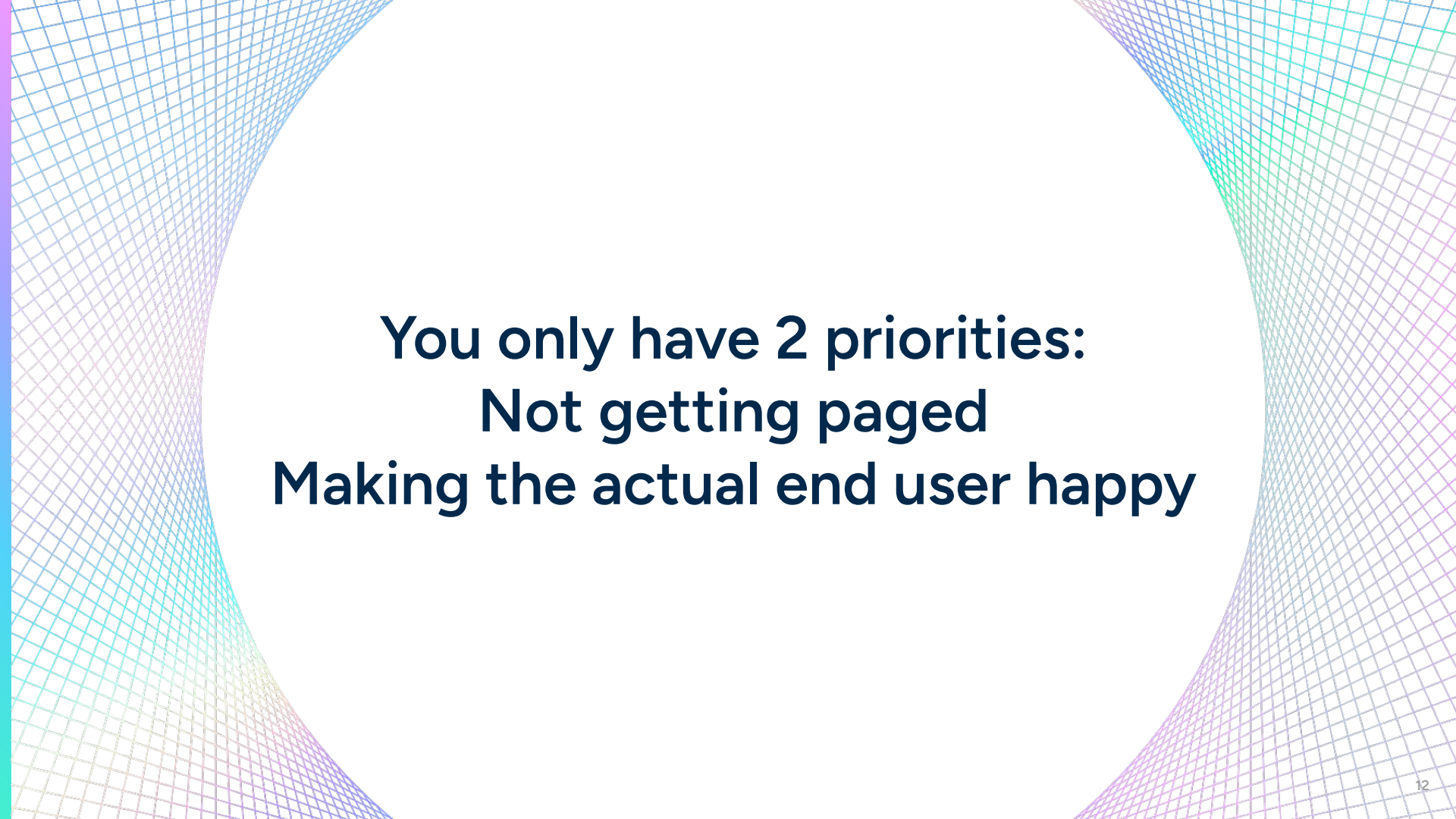
- Skip upgrades to reduce downtime
- Don't restart this machines because it has to run for weeks
- Just don't deploy code

The background features a large white circle centered on a light gray background. On the left and right sides of the circle, there are decorative elements consisting of a grid of thin, curved lines that create a tunnel-like or lens effect. These lines are colored with a gradient, transitioning from purple and blue on the left to green and yellow on the right. The overall aesthetic is modern and geometric.

**How do you actually improve  
resiliency?**

## Be careful around incentives

- Engineers actually like doing the right thing
- Before you roll out any incentives, test them against the smart-alec in your organization
- Lots of small steps are easier than fewer big ones



**You only have 2 priorities:**  
**Not getting paged**  
**Making the actual end user happy**

IN THAT THEY DON'T WISH YOU SPECIFIC HARM

## Making Friends

- You want your teams to come to you for advice
- This means coming to them with solutions rather than demands
- Sometimes you have to let them touch the stove

## Other Players



Cost



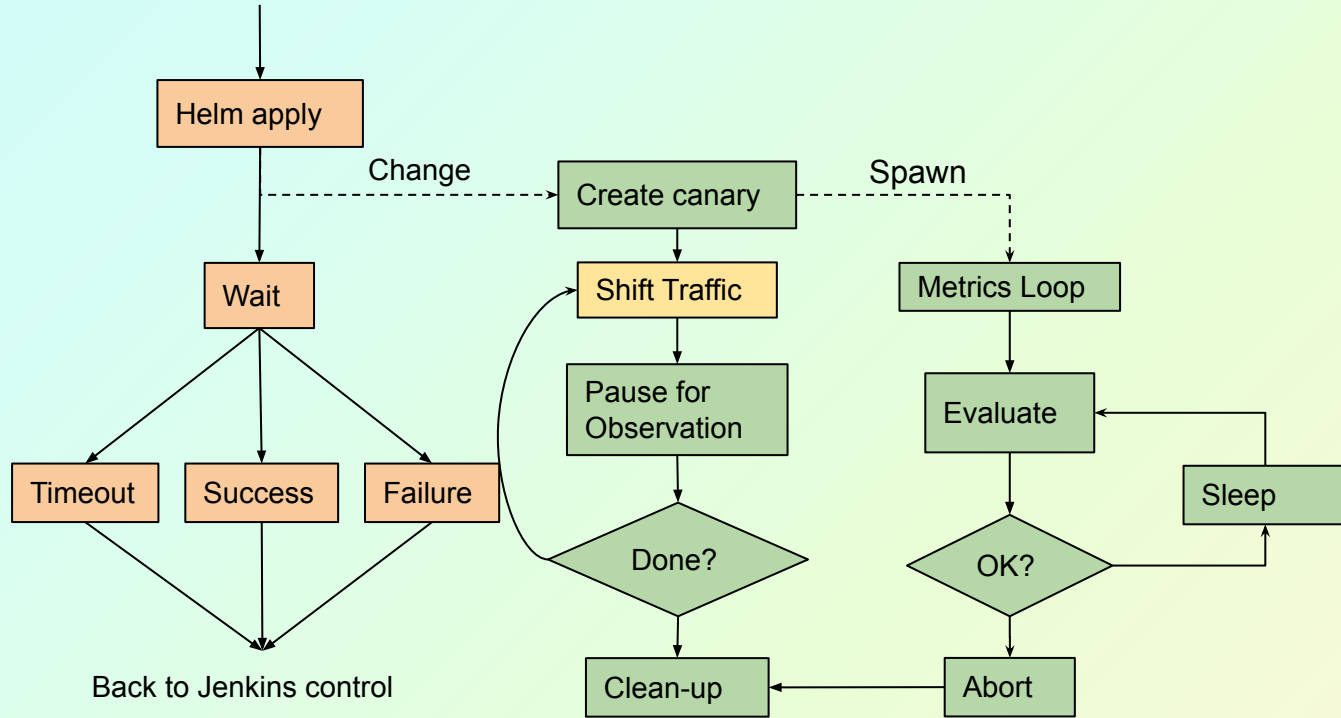
Security



Developer Velocity



# Automation





**4th Law of thermodynamics**  
**People will chose the laziest option**

I'VE MADE A HUGE MISTAKE

## Make the right choice, the easy choice

- Make the default the correct choice
- Make people turn off backups and alerting
  - Don't require they figure out how to turn them on
- Make a golden path, but let people leave it
  - At their own peril

SEE MARGE, BECAUSE OF ME THEY ADDED A LINTER

## Use tools to encourage quality

- Linters are the perfect villain
- Have examples that show ideal behaviour
  - People will copy whatever you demonstrate

## Design for Failure

- Think about how your system will fail beforehand
- Decide how you will handle it ahead of time
  - Crash
  - Degrade
  - Alternative path

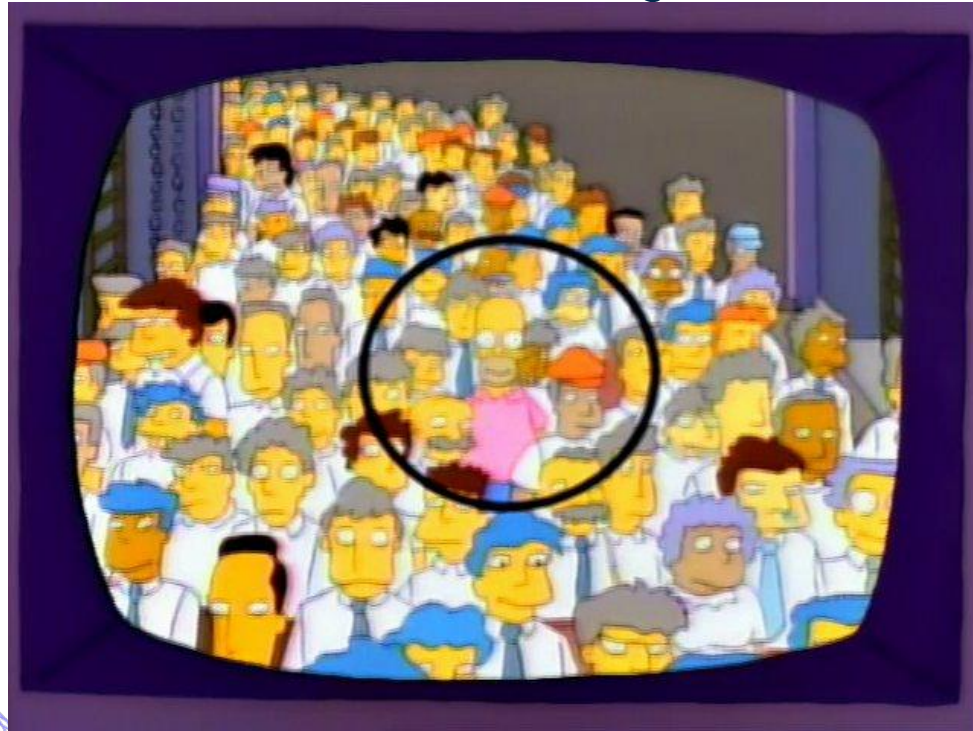
I WANNA PUSH THE BUTTON

## What happens if the self-healing breaks





# Resiliency at scale means uniformity



## When things break

- It's not a person who screwed up, it's a system that allowed them to screw up
- You need to establish learnings
  - That are actually implementable
- And then follow through

## You can't page yourself to high reliability

- The road to resiliency doesn't go there
- There is an absolute maximum of reliability that you can achieve with a human in the loop

## Read-only Fridays

- In life, there are three ways to deal with the regular needs of life
  - Get good (enough) at it
  - Hire someone else to do it
  - Structure your life around avoiding the problem

THEY LIVE IN MEETING, OF COURSE THEY KNOW BEST

## Management sign off

- Asking for permission to deploy generally doesn't work
- Your defects are still there
- But you find them less quickly

OH HALLELUJAH IT'S OVER

## The final word - Flexibility

