Octopus Deploy

Streamlining Deployments:
Lessons from Building a DevOps Paved Path for 150+ Developers

Bob Walker | Field CTO | Octopus Deploy

@bobjwalker

bob.walker@octopus.com

@bobjwalker

bobjwalker.octopus.app

Thank You TechBash 2024 Sponsors!

Visit https://techbash.com/#Sponsors to visit our sponsors' websites and learn more about them!

Bob Walker

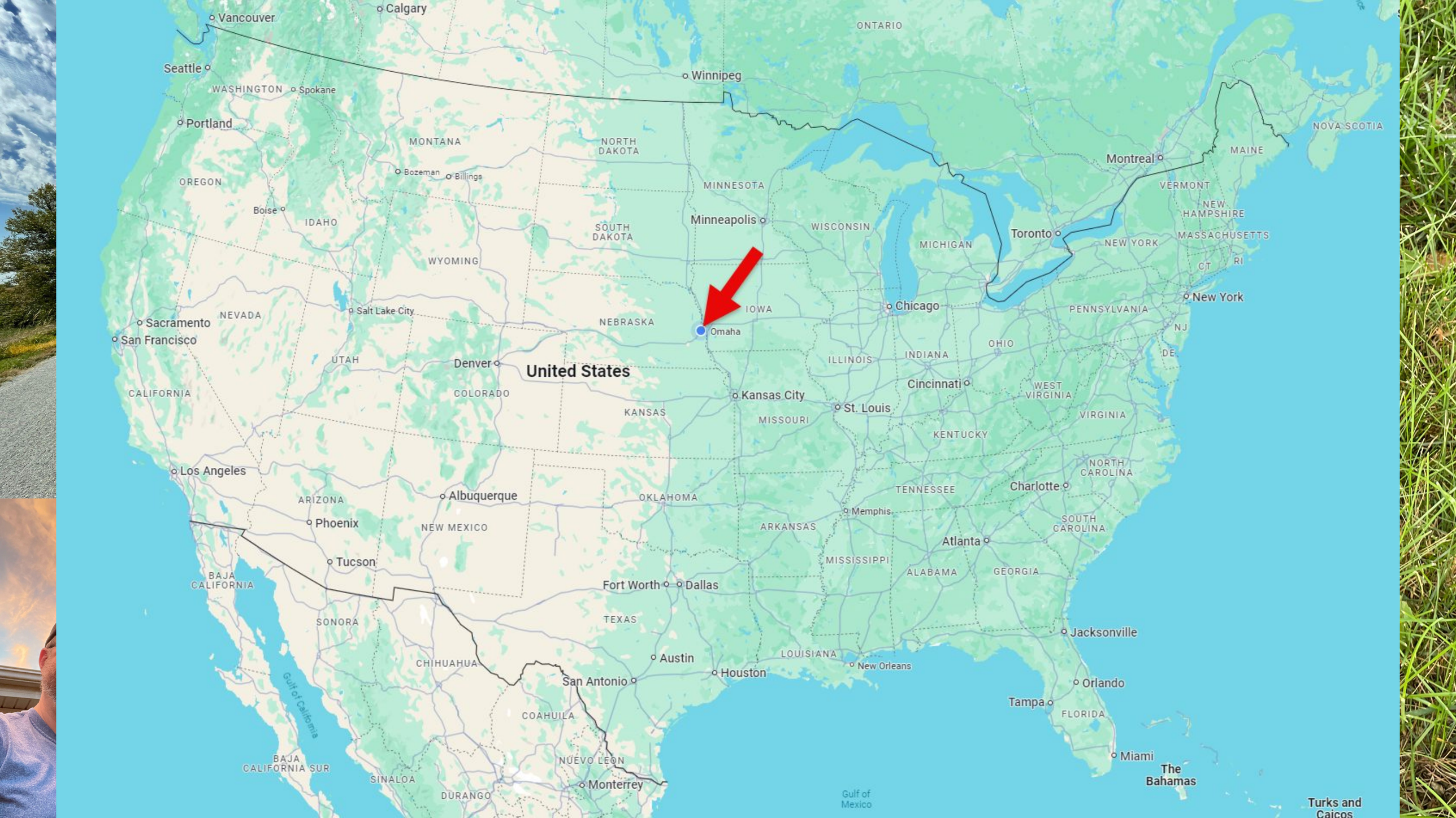
Field CTO
Octonaut since 2018

@bobjwalker

bob.walker@octopus.com

@bobjwalker

bobjwalker.octopus.app


Vancouver
Calgary
ONTARIO
Seattle
Winnipeg
WASHINGTON
Spokane
Portland
MONTANA
NORTH DAKOTA
NOVA SCOTIA
MAINE
Montreal
Bozeman
Billings
OREGON
MINNESOTA
VERMONT
NEW HAMPSHIRE
Boise
IDAHO
Minneapolis
WISCONSIN
SOUTH DAKOTA
Toronto
MASSACHUSETTS
MICHIGAN
NEW YORK
WYOMING
CT
RI
Chicago
New York
Salt Lake City
IOWA
PENNSYLVANIA
NEVADA
Sacramento
NEBRASKA
NJ
San Francisco
Omaha
OHIO
UTAH
Denver
United States
ILLINOIS
INDIANA
DE
CALIFORNIA
COLORADO
Cincinnati
Kansas City
WEST VIRGINIA
St. Louis
KANSAS
MISSOURI
VIRGINIA
KENTUCKY
Los Angeles
NORTH CAROLINA
Charlotte
ARIZONA
Albuquerque
OKLAHOMA
TENNESSEE
Memphis
Phoenix
NEW MEXICO
ARKANSAS
SOUTH CAROLINA
Atlanta
Tucson
MISSISSIPPI
ALABAMA
GEORGIA
BAJA CALIFORNIA
Fort Worth
Dallas
TEXAS
SONORA
Jacksonville
LOUISIANA
Austin
CHIHUAHUA
New Orleans
Houston
San Antonio
Orlando
Tampa
COAHUILA
FLORIDA
Gulf of California
Miami
The Bahamas
BAJA CALIFORNIA SUR
NUEVO LEON
SINALOA
Monterrey
DURANGO
Gulf of Mexico
Turks and Caicos

DTN

Farm Credit Services of America

Sandhills Global

ELECTION Systems & Software

west®

Octopus Deploy

DTN

Farm Credit Services of America

Sandhills Global

ELECTION Systems & Software

west

Octopus Deploy

How I accidentally built a golden path

Agenda

- Difference between a paved and golden path
- Building the paved path at DTN
- Building a golden path at FCSA
- Rolling out the golden path
- Reflection after eight years
- Contributing to a golden path

Disclaimer

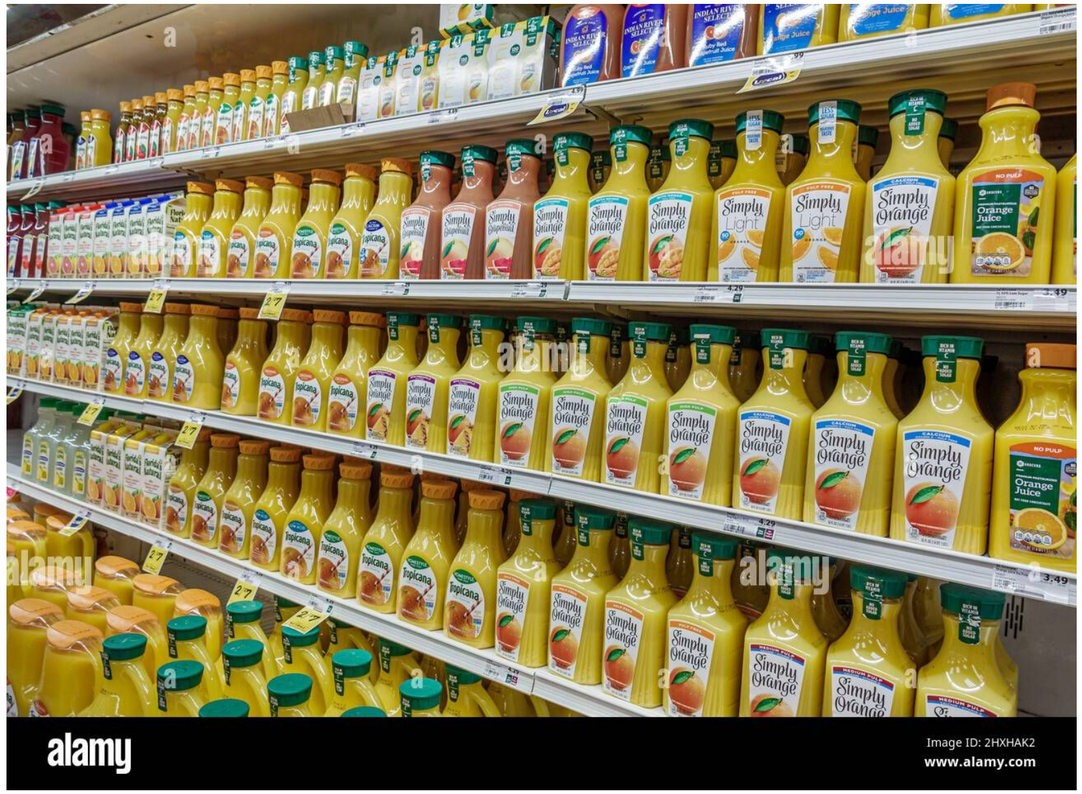
Paved vs. Golden Paths

Paved Path

- Looks to improve an existing process.
- Attempts to leverage automation to remove a specific pain point.
- May or may not solve the underlying problem.
- May or may not be better.

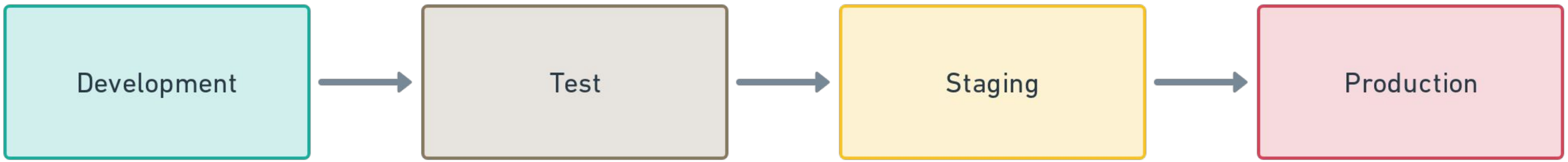
Golden Path

- Reduce burden and remove pain points.
- Solves specific problems developers are encountering.
- May involve automation, or just a new way of doing something.
- Better than what developers currently do, and is easy to adopt.
- Not forced onto a team, it is too enticing to not use it.

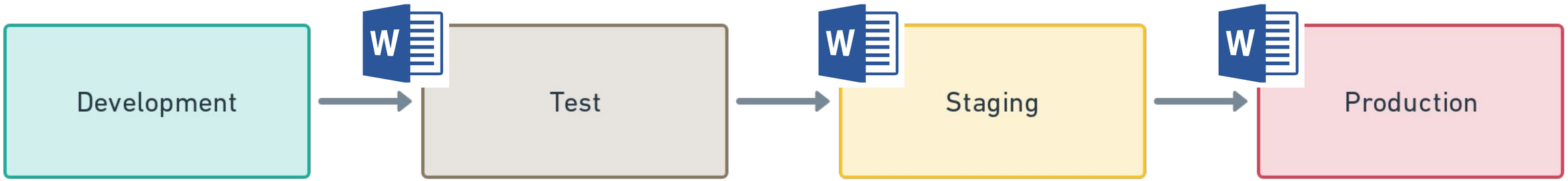


Building a Paved Path at DTN

Paved Path Example



Paved Path Example



Automation will fix this!

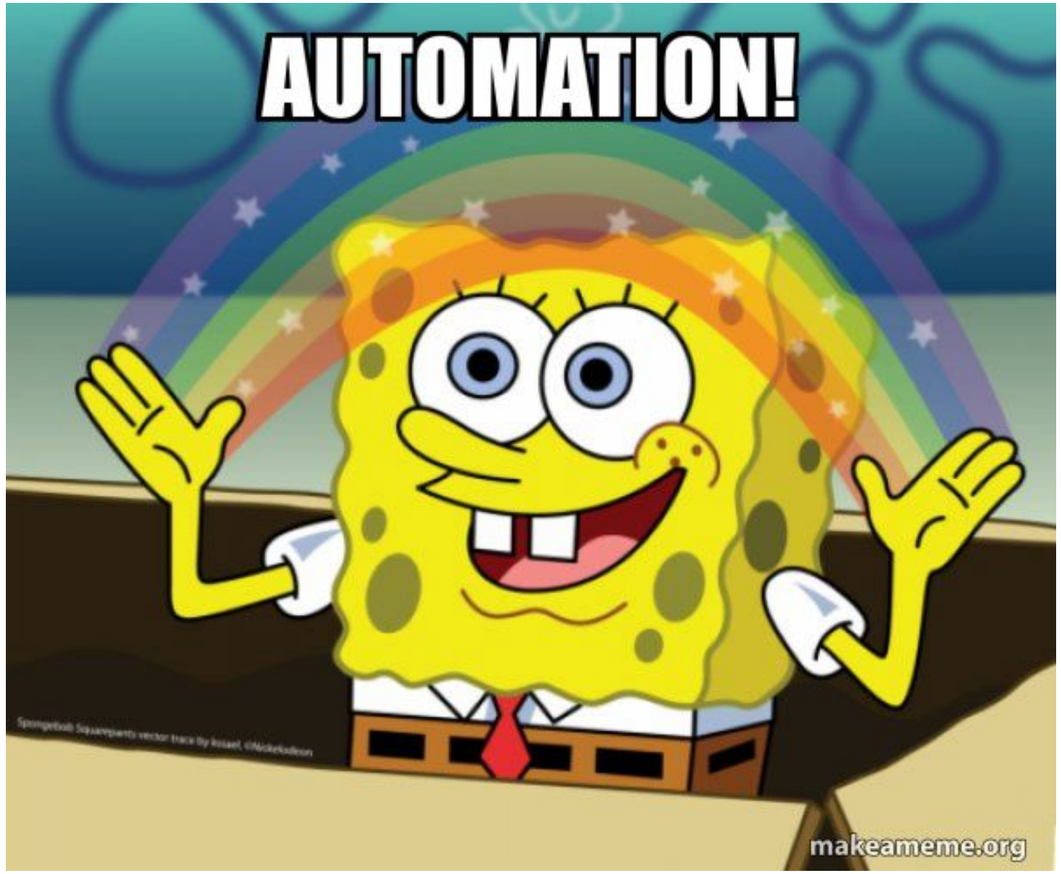
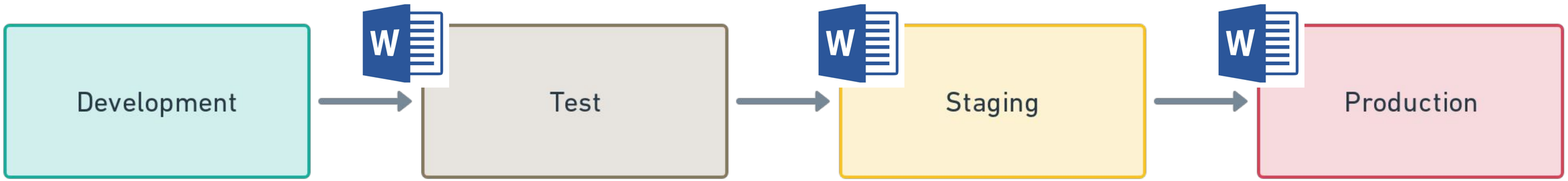


Image Source: https://makeameme.org/meme/automation-fdf77f0941

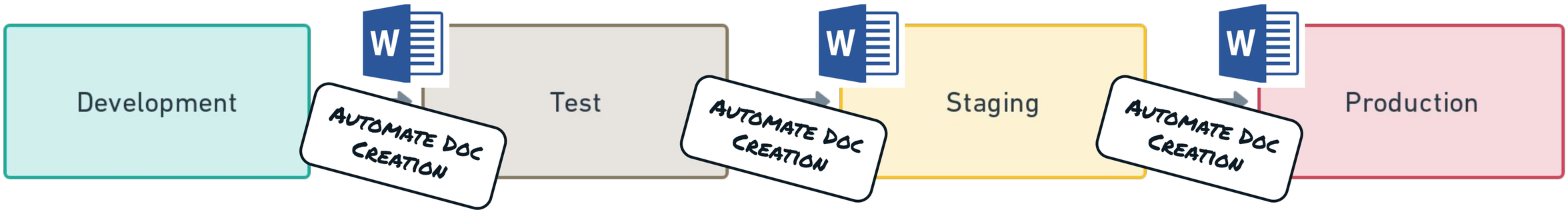
Paved Path Example

🤔 WORD FILES ARE JUST FANCY XML FILES



Paved Path Example

🤔 WORD FILES ARE JUST FANCY XML FILES



Automating a terrible process doesn't make it less terrible

Image Source: https://tenor.com/view/pringles-potato-chips-automation-inventionv-cool-gif-15081336

Paved paths automate for the sake of automation

- Why are we using word documents for deployment instructions?
- Why aren’t we automating the deployment process?
- Why do we need to build a new document / process for each deployment?
- Why are we rebuilding for each environment?
- How we need a two hour manual deployment for QA?
- Why are we doing rolling deployments but not making our database changes backward compatible?

Right now: DevEx problem

Unable to tie it to a key business initiative

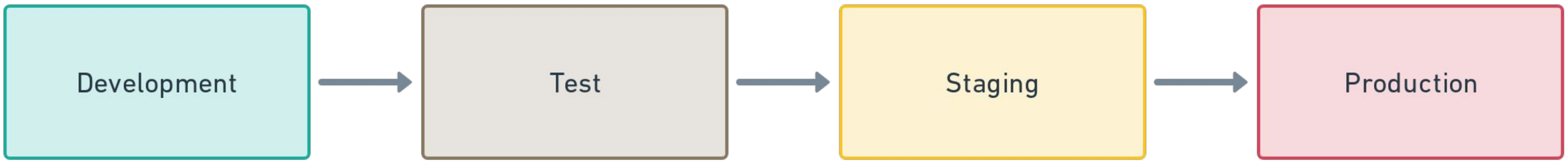
Building the Golden Path at FCSA

The problem: two key business goals

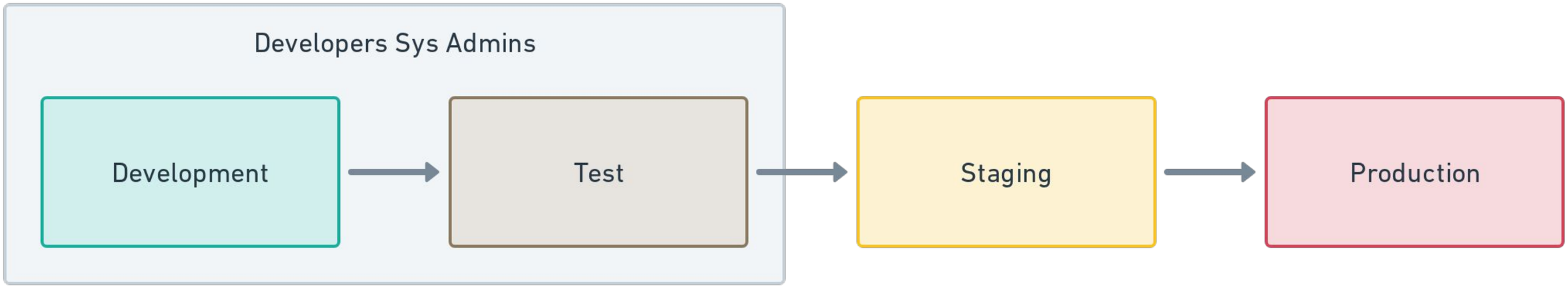
- February 2015
  - Full integrated with an acquired bank
  - Need to keep their loans separate due to regulations
  - But needed to process all loans via the same tooling
- November 2016
  - Strategic partnership with another bank
  - We'd provide the loan processing
  - They'd provide the customer information
- All apps needed to be updated
  - Hundreds if not thousands of deployments to accomplish those goals

Two Hour Deployments

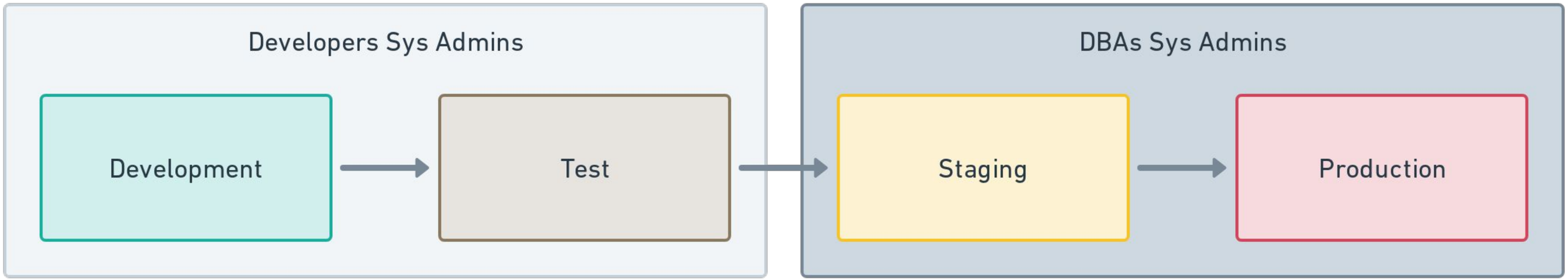
Desired Delivery Pipeline



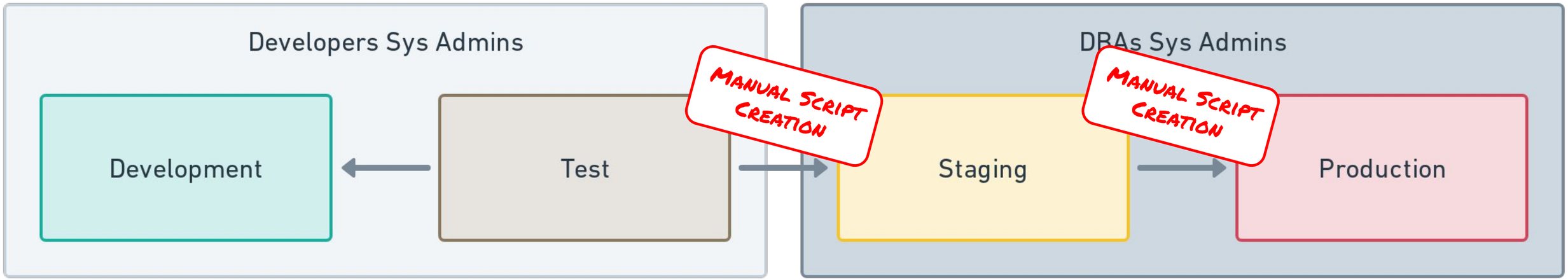
Actual Delivery Pipeline



Actual Delivery Pipeline



Actual Delivery Pipeline



60%

All Releases required an emergency fix the next day

Identifying critical problem to solve

John Morehouse https://sqlrus.com

How we did it

- Two day meeting
  - DBA
  - Database Architect
  - Database Developer
  - Lead Developer
  - Manager
- Identified the key business rules
- Built a process
- PoC the tooling
- Demonstrated the process end of two days

Building a golden path

1. Agree there is a problem to solve
2. Identify key roles and representatives
3. Assign RACI
4. Key representatives agree on business rules
5. Make process changes
6. Pick the tooling
7. Demonstrate it
8. Roll it out

How I built the paved path

1. Agree there is a problem to solve
2. Identify key roles and representatives
3. Assign RACI
4. Key representatives agree on business rules
5. Pick the tooling
6. Make process changes
7. Demonstrate it
8. Roll it out

First Step: Agree on the problem

- Can we tie this to a critical business initiative?
- How often do issues happen?
- Is it a result of manual work or is it automated and needs better documentation?

Second Step: Identify Key Roles and Representatives

- Focus on the roles impacted first
- Avoid: Including everyone who has an opinion or interest
  - 30+ people from various roles
  - Impossible to get consensus
  - Time wasted - resulting in managers making decisions
- Leverage a representative style democracy
  - Person is empowered to speak for their role
  - They can make the dozens of micro-decisions
  - They bring concerns back to the wider groups

Second Step: Identify Key Roles and Representatives

- DBA
- Database Architect
- Database Developer
- Lead Developer
- Manager

Third Step: Assign RACI

- **R - Responsible**: The doers - at least one responsible party required.
- **A - Accountable:** The reviewer - only one person allowed.
- **C - Consulted**: The domain experts - provides input but doesn’t do work.
- **I - Informed**: The impacted - kept informed as impacts their day to day

Third Step: Assign RACI

- DBA
- Database Architect
- Database Developer
- Lead Developer
- Manager

Third Step: Assign RACI

- **DBA** - Responsible
- **Database Architect** - Accountable
- **Database Developer** - Consulted
- **Lead Developer** - Responsible
- **Manager -** Informed

Fourth Step: Key representatives agree on business rules

- Whiteboard the current process
- Ask why a specific step must occur
- Keep track of all the decisions in the process
- Keep asking why until you get a business reason

Fourth Step: Key representatives agree on business rules

- **Auditing** - Business request reason for a change
- **Regulation** - Separation of duties
- **Data integrity -** is more important than speed with million dollar loans
- **Reliability** - The change will work when it is deployed

Fifth Step: Process changes to enforce rules

- Whiteboard the desired process
- Keep the old process in mind, but start from scratch
- Refer back to the business rules to ensure the process follows them

Fifth Step: Process changes to enforce rules

- Truth center for changes
  - Auditable
  - Link to business change request
- Changes must be reviewed both automatically and manually
  - Syntactically correct
  - Company standards
  - Best practices
- Review is done by someone who didn’t make the change
- Consistent process across all environments

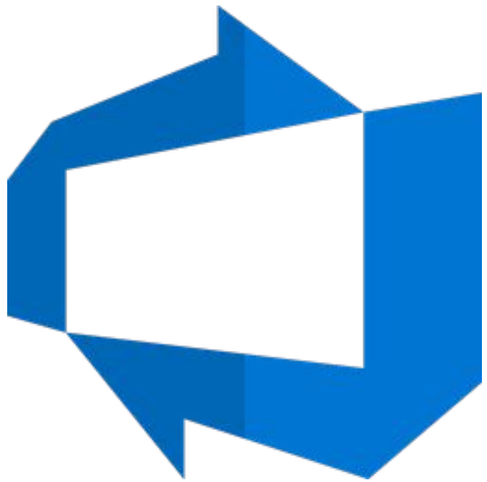
Fifth Step: Make process changes

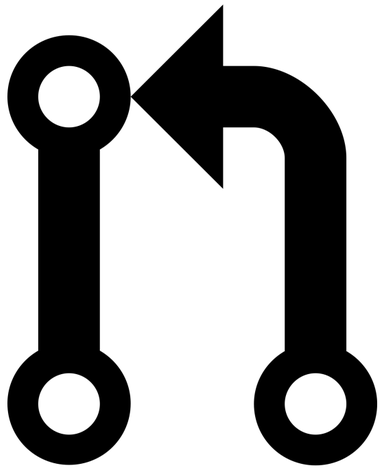
- Put the database schema into source control
- Require local database development
- All changes made on a branch
  - Trigger build on each branch check-in
  - Always verify schema during the build process
  - Pull Requests to approve changes
- Deployment tool to be used across all environments

Sixth Step: Pick the tooling

- Evaluate free and paid tooling.
- PoV with vendors based on your requirements.
  - Think at scale will this work for 100s or 1000s?
  - Be wary of “snake oil” promises “you just need to…”
  - Do they want a partnership or are they looking to simply make a sale?
- Be wary of shadow IT - when tools are misused.
- Personal fan of best in breed vs. all-in-one.

Sixth Step: Pick the tooling





Seventh Step: Demoing the golden path

- Discuss what the problem to solve is
- Articulate why that problem important to solve
- Demonstrate how the process will help solve the problem
- Address questions and concerns

Seventh Step: Demoing the golden path

- Created an invite for all staff
- Demoed the new process, showing it deploying up to test
- Answered questions and concerns
- Discussed timelines

Eighth Step: Rolling out the golden path

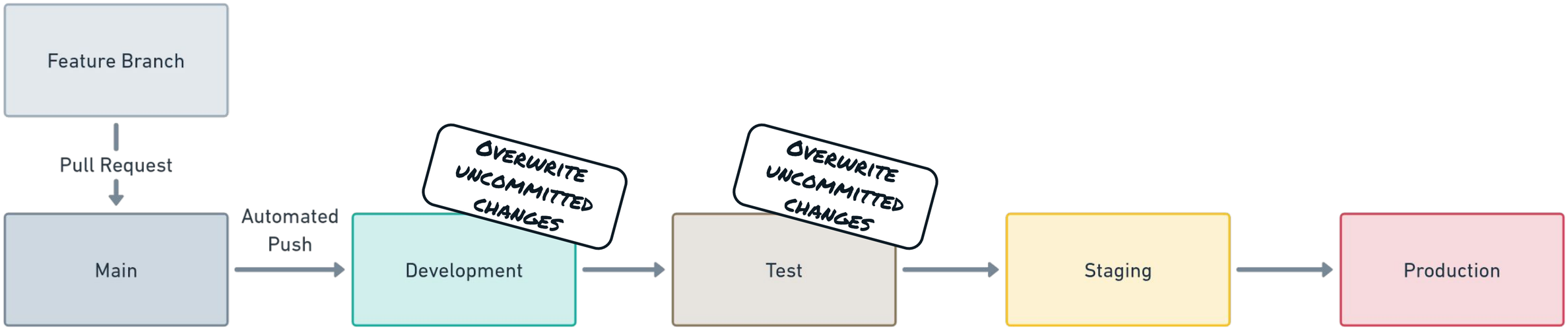
1. Pilot - Use the process to deploy to Production for one or two applications
2. Early Adopters - Repeat for a couple more applications
    a. Find flaws in the process
    b. Identify shortcuts and assumptions
3. General Adoption - Rollout to everyone
    a. Training
    b. Documentation
    c. Templates

Rolling out the Golden Path

#1 Challenge - Enforcement

- Cannot remove sysadmin rights on dev and test right away
- Haven’t seen the value
- Go back to path of least resistance to get work done
- Doomed attempts at other automation in the past

v1 Delivery Pipeline



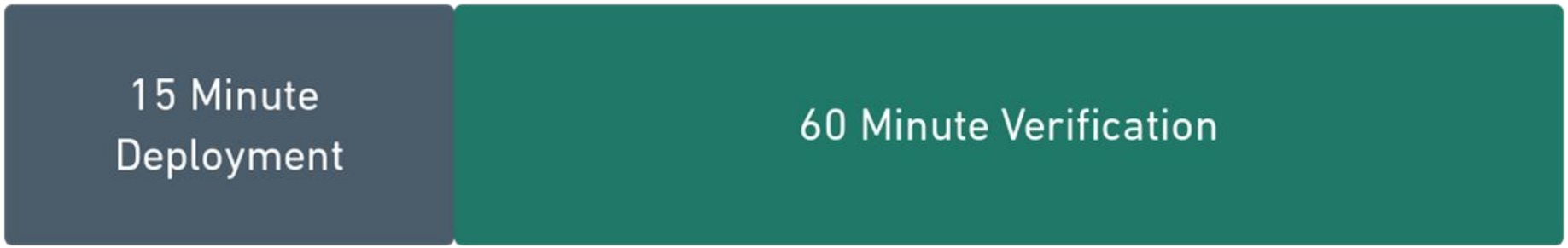
0%

All Releases required an emergency fix the next day

120 Minute Deployment

100 Minute Deployment

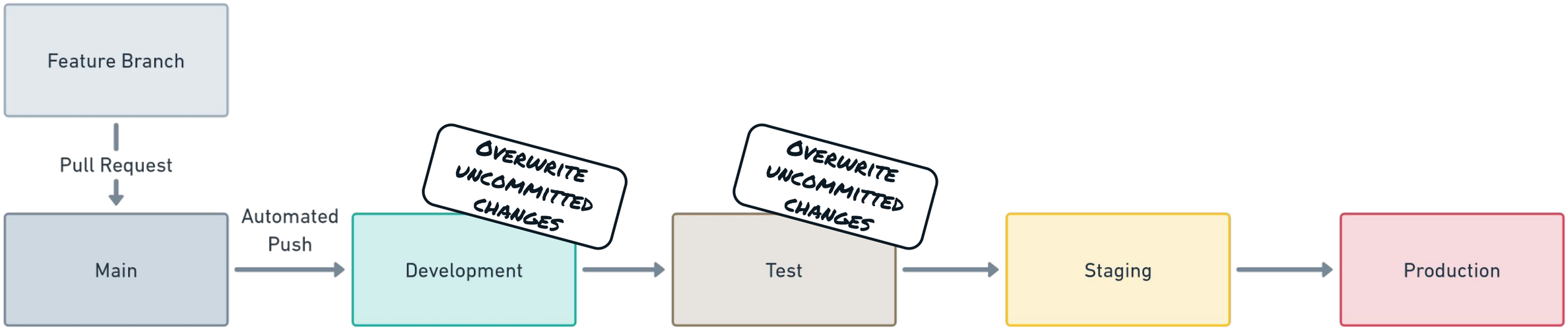
75 Minute Deployment



15 Minute Deployment

Key Mistake: Believing that process was perfect

v1 Delivery Pipeline



Say no to changes that compromise business rules

Didn’t do a good job answering: Why should we change?

- **Deployment Time:** 2+ Hour to 15 minute Deployments
- **Deployment Frequency:** Quarterly releases to every ten days
- **Failure Rate:** Emergency Fixes the next day from 60% to 0%

Make it enticing

- Track key metrics
  - Deployment frequency
  - Deployment time
  - Post deploy emergency fixes
  - Recovery time
- Sell the process
  - Case studies
  - Internal blog of success and failures
  - Demonstrations

Reflection after eight years

I moved onto another company

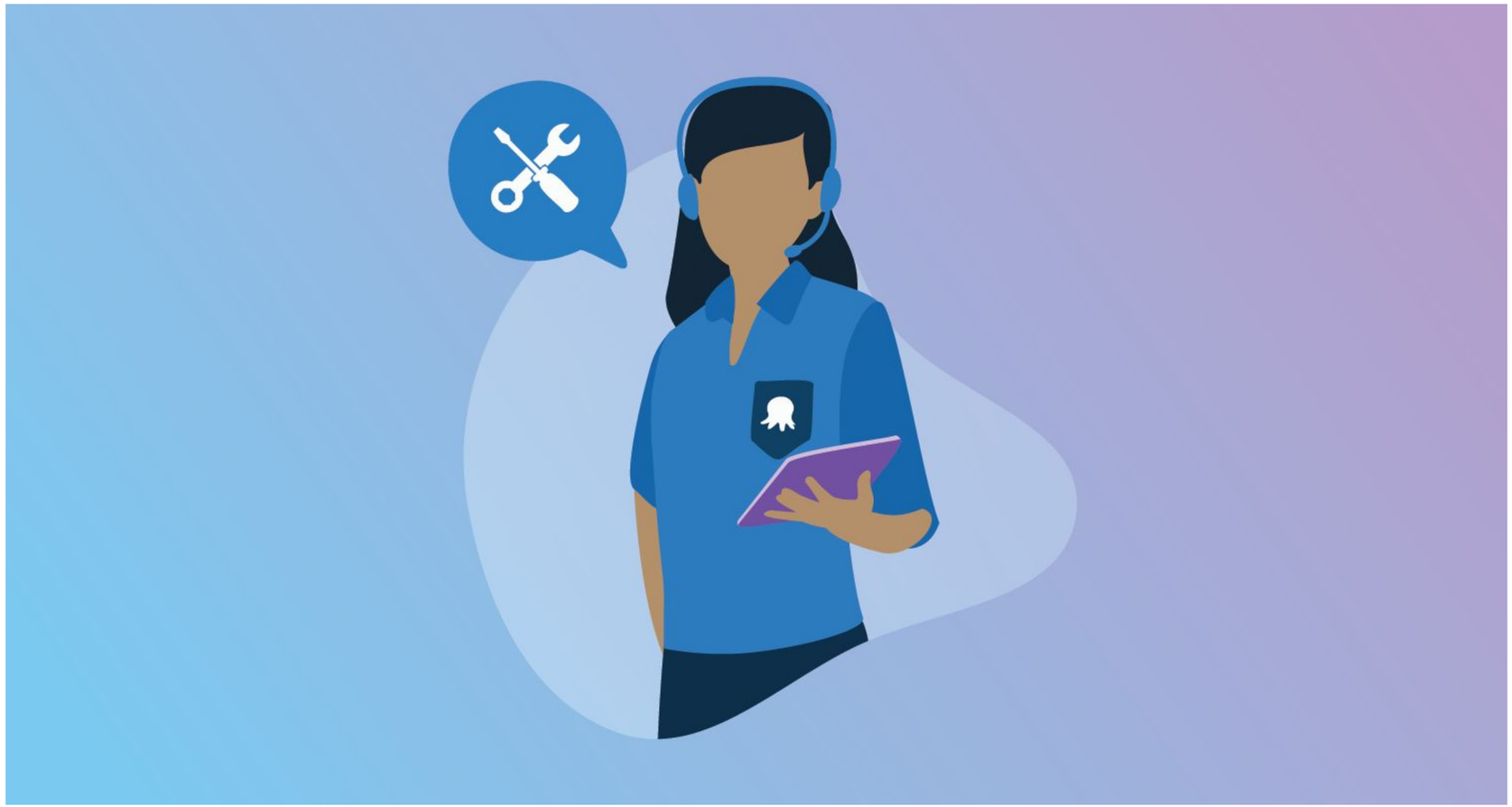
Problems I discovered later

Image Source: https://getyarn.io/yarn-clip/d4948831-0519-4532-9872-9ae747471d67/gif

Change is inevitable

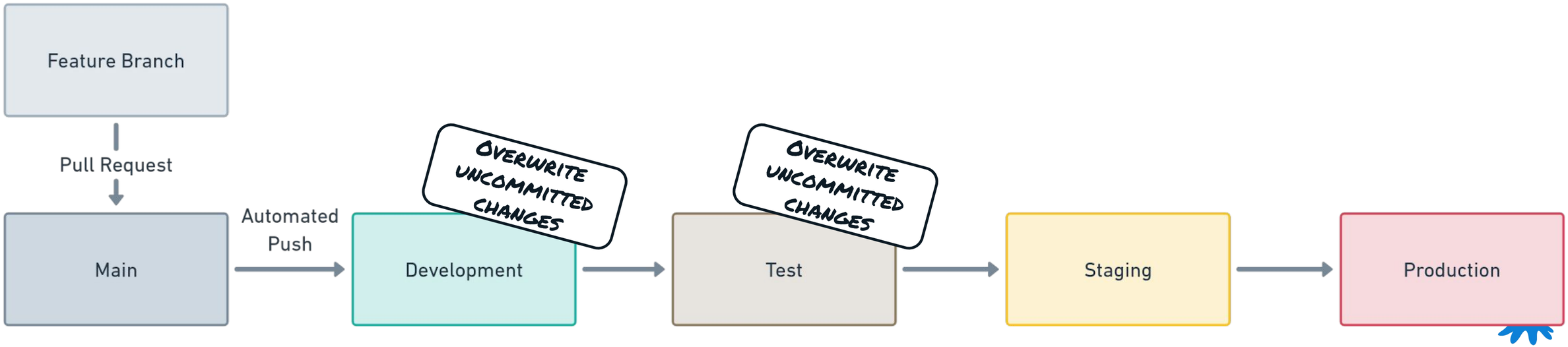
- Understood the tooling better
- Steps we thought were necessary were superfluous
- Unexpected use cases
- Don’t try to design the perfect process give yourself room to iterate

Ask why when saying no



Feedback Loop

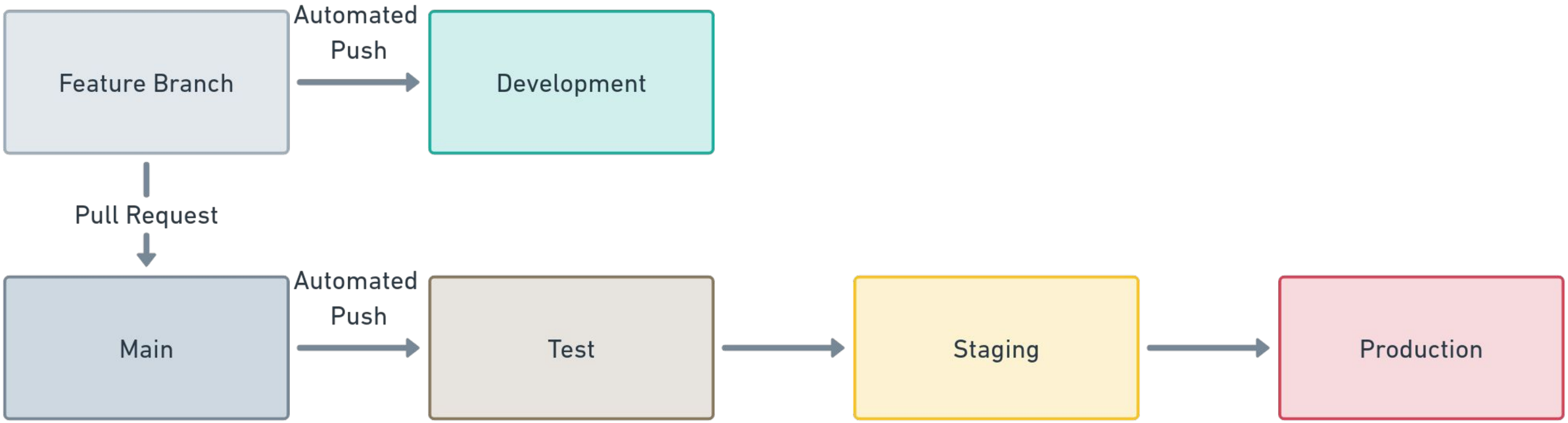
- To get feedback we had to push to the shared server
- Unfinished changes were getting merged into main
- Needed to adjust our usage of the tooling as we learned more



Rules for your processes

- Deploy all tightly coupled components together
- Follow the same process for all scenarios - including emergencies
- Main or primary branch must always be ready to deploy
- Don’t build unless something has changed
- It’s okay to redeploy the same version - add logic gates to skip where appropriate

V2 Delivery Pipeline



Rolling out changes

- Pilot / Early Adopters - manual is okay
- General adoption - pick a model
- Make changes “for people” not “to people”
  - Automate and create PRs to push changes
  - Educate why changes are needed

Handling Drift

Template

Environments | Hosts

Runbooks | Variables

Deploy Process

ServiceNow Approval

Prepare Infrastructure

Deploy via Helm

Log Deployment in DataDog

Update Feature Flag

CHANGE COMPLIANCE TOOLING

UPDATE DEPLOYMENT STANDARDS

DRIFT

Application #1

Environments | Hosts

Runbooks | Variables

Deploy Process

ServiceNow Approval

Prepare Infrastructure

Deploy via Kustomize

Log Deployment in Splunk

Swap Blue with Green

USING NON-STANDARD TOOLS

USING NON-STANDARD DEPLOYMENT STRATEGIES

Application #2

Environments | Hosts

Runbooks | Variables

Deploy Process

JSD Approval

Prepare Infrastructure

Deploy via Helm

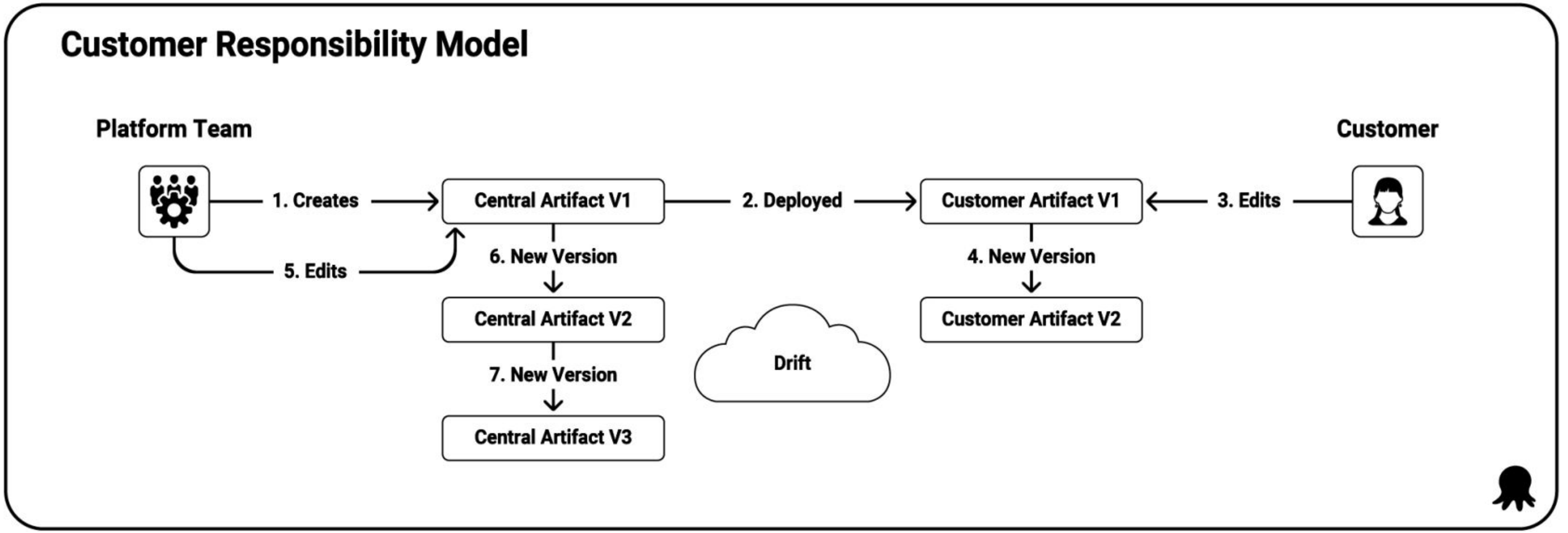
Log Deployment in Splunk

CHANGES COMPLIANCE TOOLS

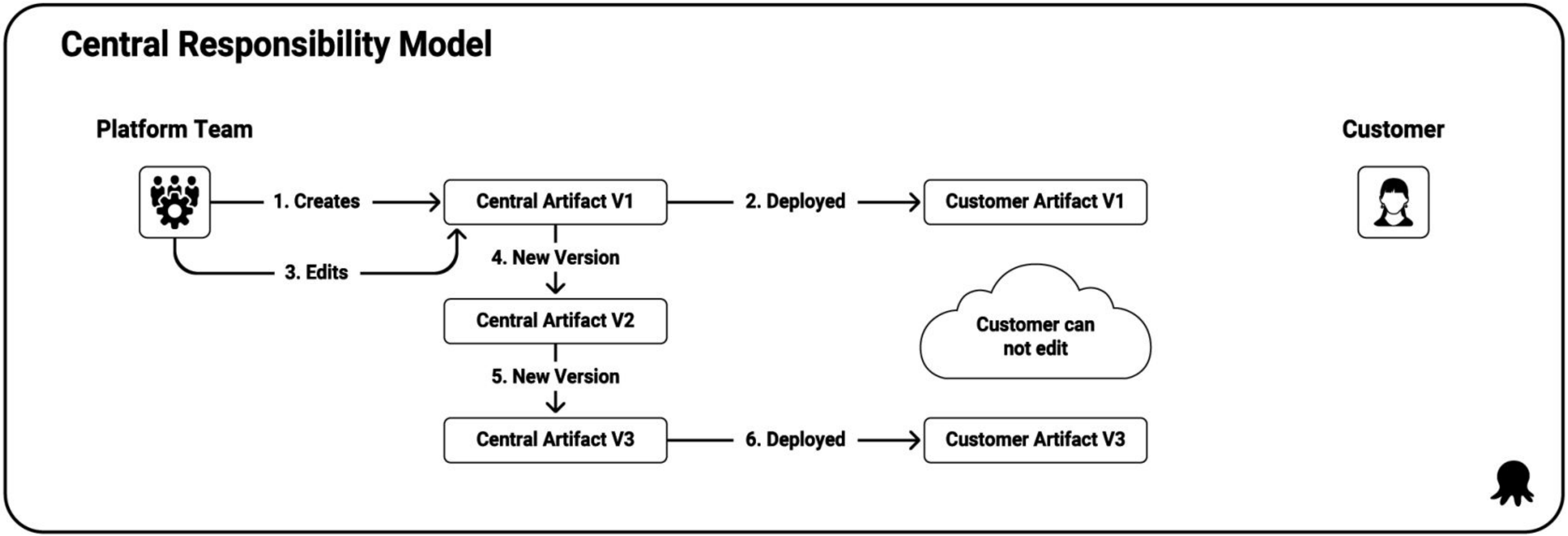
PLATFORM TEAM

DEVELOPMENT TEAMS

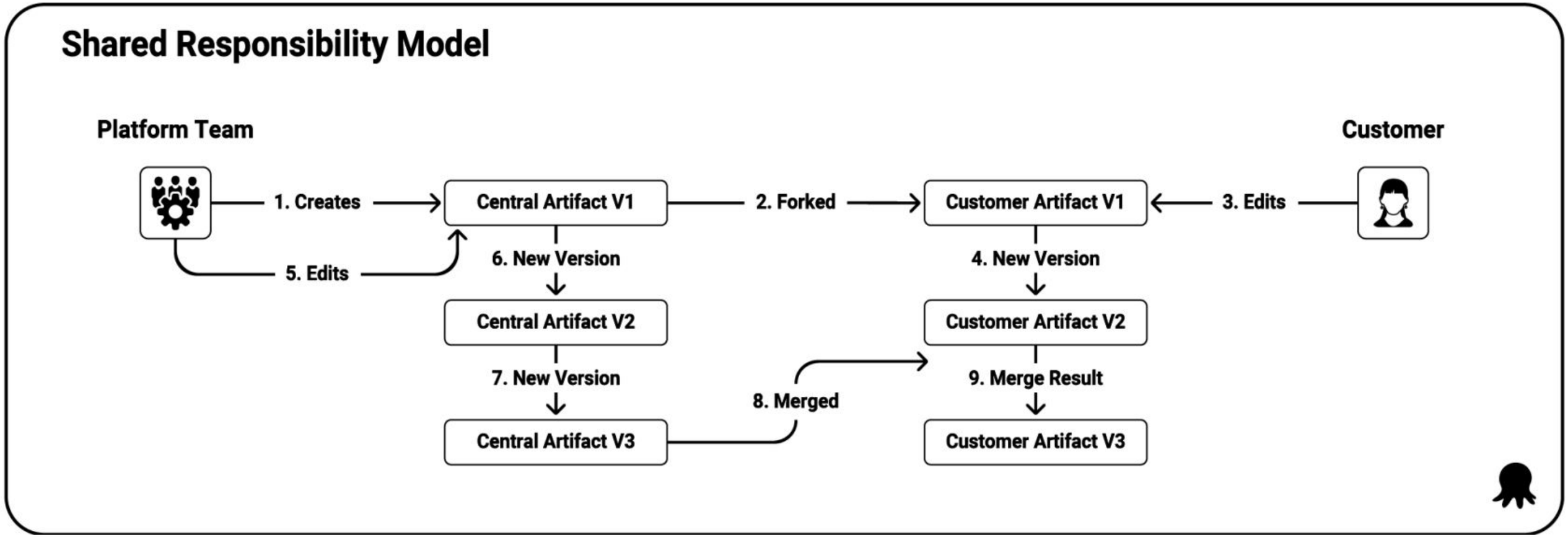
Customer Responsibility Model



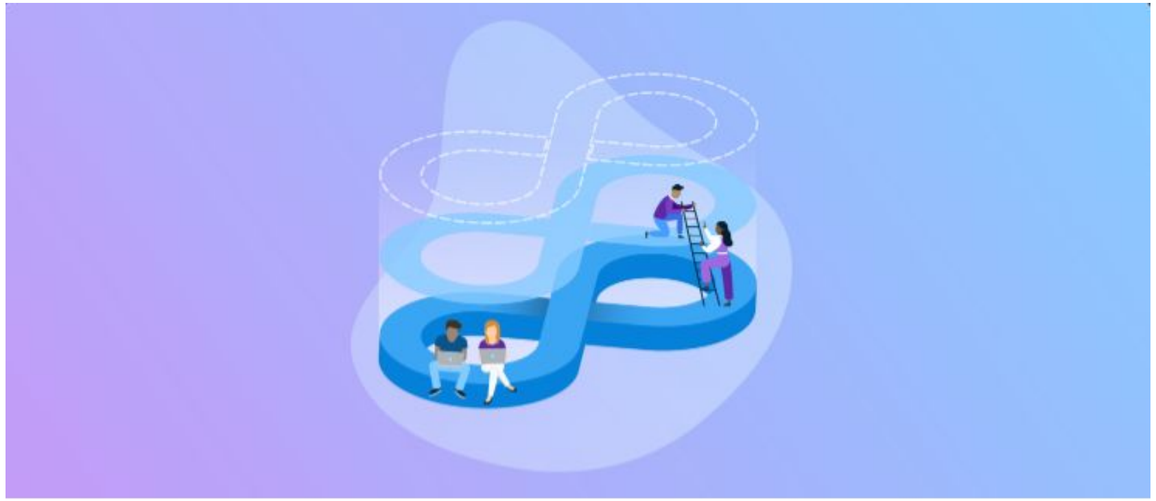
Central Responsibility Model



Shared Responsibility Model



Not an easy problem to solve



Blueprints to empower Platform Engineers

SHIPS MAHINDRA | Posted on May 2

Submit your ideas on how Blueprints could benefit you below!

Over the past few months we have been learning about the needs of Central DevOps or Platform teams that own and maintain Octopus. Based on that, we heard that Platform teams want to:

- Reduce the onboarding or getting started time for a new pipeline
- Have a consistent pipeline that is easier to maintain

We are investigating Blueprints for Process and Runbooks to enable standardising on a grouping of Steps that can be reused across multiple projects. Related ideas that we are exploring are:

- How to validate Processes by defining rules / guardrails
- How to roll out changes to these Blueprints
- How to view usages of Blueprints and impact on team metrics

Contributing to a golden path

It is everybody’s process not just yours

Make it easy to contribute changes

- Benevolent owners - today we call them platform engineers
- Processes should be open source
- Make it easy for anyone to contribute and improve
- Contribution guidelines
  - How to contribute
  - How to test
  - PR requirements

Octopus Deploy

Thank you!

Bob Walker - Field CTO at Octopus Deploy

@bobjwalker

bob.walker@octopus.com

@bobjwalker

bobjwalker.octopus.app