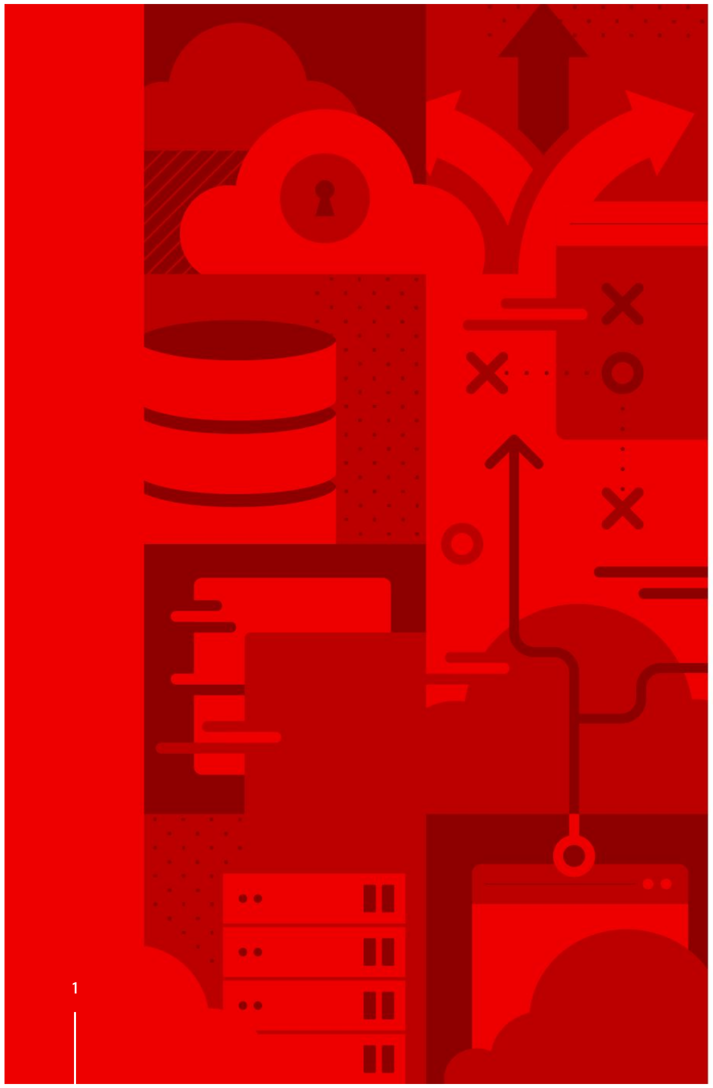


Building a Culture of Continuous Learning

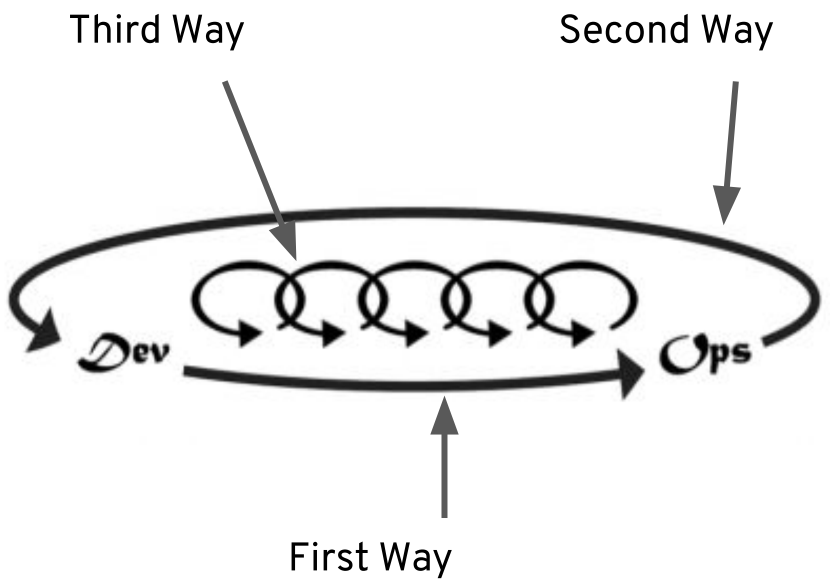
John Osborne
Chief Architect, Public Sector
josborne@redhat.com
@OpenShiftFed

The Three Ways

1. "Systems thinking"
2. "Amplify feedback loops"
3. "Culture of continual experimentation and learning"

These are deliberately tech agnostic, though often tech-enabled

https://itrevolution.com/the-three-ways-principles-underpinning-devops/



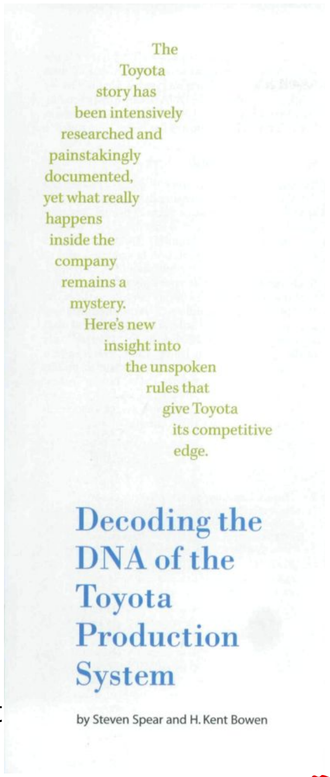
Lean Manufacturing Applied to Software
(Replicating the Output from the Toyota Production System)

- Developer self-service
- Smaller batch sizes
- Fast feedback
  - Logging
  - Monitoring
  - Telemetry
- **Shifting left on security**
  - Automated testing / scans
  - Cross functional teams involved earlier in process

Boeing's Everett factory near Seattle
https://upload.wikimedia.org/wikipedia/commons/c/c8/At_Boeing%27s_Everett_factory_near_Seattle_%2891301605958%29.jpg
Creative Commons

Building a Culture of Continuous Learning
(Replicating the DNA from the Toyota Production System)

- TPS creates a community of scientists
- Whenever Toyota defines a specification, it is establishing sets of hypotheses that can then be tested (scientific method)
- The system actually stimulates workers and managers to engage in the kind of experimentation that is widely recognized as the cornerstone of a learning organization



The Toyota story has been intensively researched and painstakingly documented, yet what really happens inside the company remains a mystery. Here's new insight into the unspoken rules that give Toyota its competitive edge.

Decoding the DNA of the Toyota Production System

by Steven Spear and H. Kent Bowen

"To understand Toyota's success, you have to unravel the paradox - you have to see that the rigid specification is the very thing that makes the flexibility and creativity possible."

Rules are the DNA of the Toyota Production System

How People Work

- All work shall be highly specified as to content, sequence, timing, and outcome
- Even complex and infrequent activities, such as training an inexperienced workforce at a new plant, launching a new model, etc are designed according to this rule.

How People Connect

- Every customer supplier connection must be direct
- Must be an unambiguous yes-or-no way to send requests and receive responses
- As a result, there are no gray zones in deciding who provides what to whom and when

How the Production Line is Assembled

- Pathway for every product and service must be simple and direct
- No forks or loops
- Runs contrary to conventional wisdom about production lines and pooling resources

How to Improve

- Improvement must be made using the scientific method
- All the rules require that activities, connections, and flow paths have built-in tests to signal problems automatically

Toyota does not consider any of the tools or practices - such as kanbans or andon cords, which so many outsiders have observed and copied-as fundamental to the TPS. Toyota uses them merely as temporary responses to specific problems that will serve until a better approach is found or conditions change. ***They're referred to as "countermeasures," rather than "solutions," because that would imply a permanent resolution to a problem.***

That’s great, but I live in the real world and everything is a dumpster fire

- Take a deep breath
- Changes span tools, process, culture
- Focus on constraints
- Use data available from the State of DevOps report
  - Leverage OSS
  - How you define cloud matters
  - Shift left on security

State of DevOps Report
Key Findings (team level)

- **4 key metrics differentiate performers**
- **Open source software improves performance**
- Outsourcing by function hurts performance
- **How you implement cloud infrastructure matters**
- Key technical practices drive high performance - *including “continuous testing… **integrating security earlier**”*
- Tightly coupled architectures hurt performance
- Concurrent efforts (processes & tech) drive success

Accelerate State of DevOps 2019
https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Cloud Infrastructure

- It’s not where you run, but how you run
- Cloud can be run on a mainframe, the tactical edge, or a special access environment
- It fundamentally matters how you define cloud for your mission
- Elite performers are 24x more likely to hit these characteristics

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-145

- Self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Red Hat

Back to the dumpster fire

- Focus on your constraints
  - Continuous learning requires a stable system
- Learn by doing industry
  - No one true light
- Prepare for common pitfalls
- Leverage open source and cloud infrastructure
- Everyone can do *something*, that *something often depends on your role*

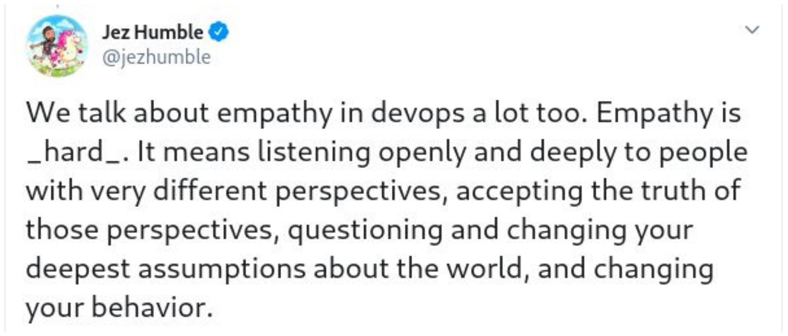
Tips On Getting Started

- Choose your initial applications wisely
  - Low resistance, high influence
  - Start small and build momentum
  - Be ready for the frozen middle
- Create a cross functional team for each application
- Make your work visible
- Stay laser focused on removing constraints

https://www.govtech.com/opinion/How-Government-Can-Accelerate-the-Adoption-of-DevOps-Contributed.html

Build Bridges Not Walls

- Create Cross Functional Teams
  - Assign IT liaison to each teams
  - Create Shared Services
  - Use chat not tickets
- Make work visible
  - Shared tools
  - Single repository of truth
- Blameless Post-Mortems

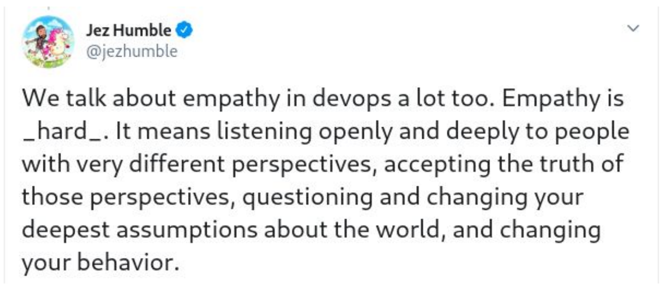

Jez Humble
@jezhumble

We talk about empathy in devops a lot too. Empathy is _hard_. It means listening openly and deeply to people with very different perspectives, accepting the truth of those perspectives, questioning and changing your deepest assumptions about the world, and changing your behavior.

https://www.govtech.com/opinion/How-Government-Can-Accelerate-the-Adoption-of-DevOps-Contributed.html

Build Bridges Not Walls

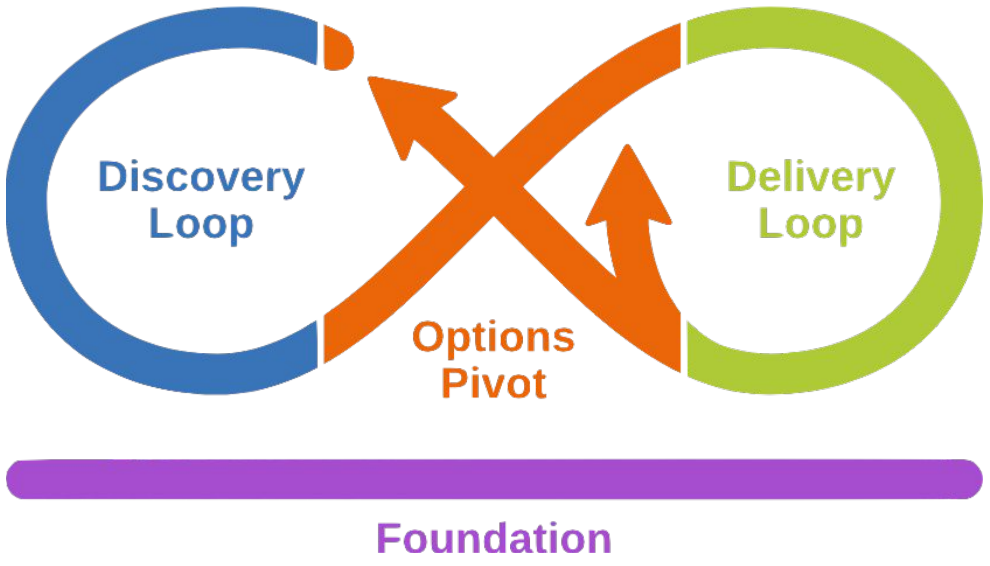
- Focus on the flow
  - Automate manual processes
  - Change boards not correlated to better software
  - Limit Work In Progress
- Practice not an end-state
  - Culture of high trust and learning


Jez Humble
@jezhumble

We talk about empathy in devops a lot too. Empathy is _hard_. It means listening openly and deeply to people with very different perspectives, accepting the truth of those perspectives, questioning and changing your deepest assumptions about the world, and changing your behavior.

https://www.govtech.com/opinion/How-Government-Can-Accelerate-the-Adoption-of-DevOps-Contributed.html

OPEN PRACTICE LIBRARY



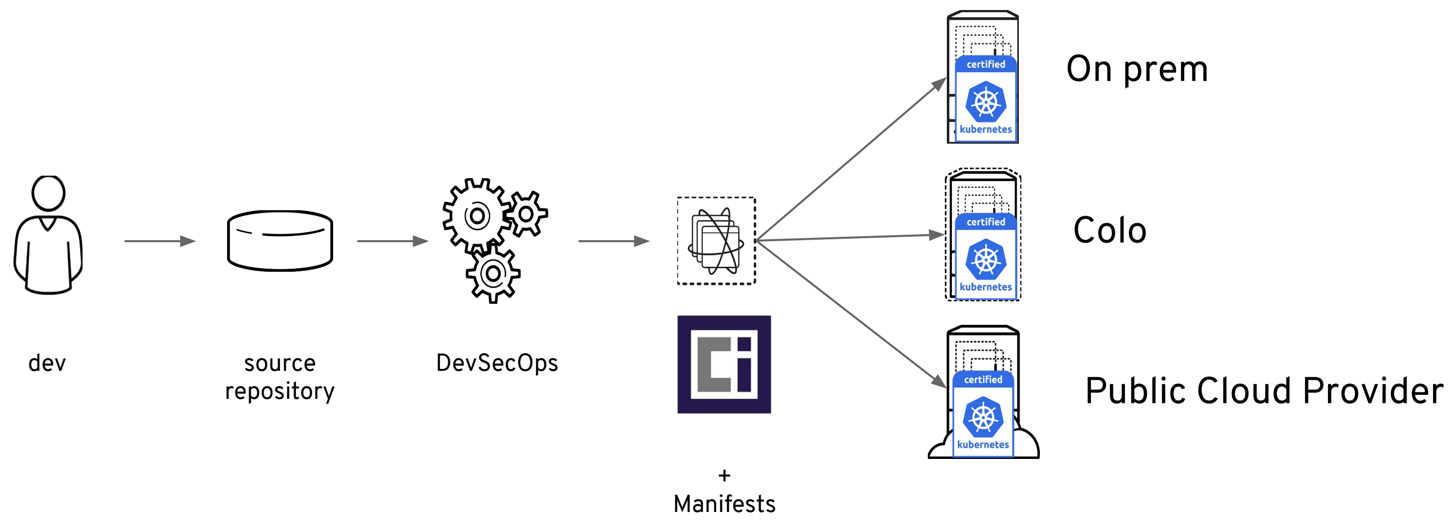
Value Stream & Process Mapping | Impact Mapping | Event Storming | Backlog Refinement | Celebrating Failure | Burndown | Social Contract | Stop the World Event | Team Spaces | Team Sentiment and more...

https://openpracticelibrary.com

Red Hat

Focus, Flow, and Joy

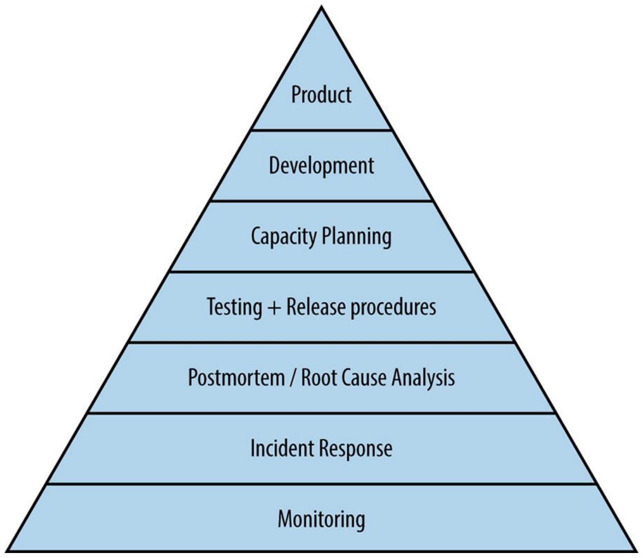
Why every DevOps effort involves a platform



Kubernetes makes for a good DevOps backbone because everyone will speak the same language

Be prepared to adopt something in the SRE space

SRE Hierarchy of Needs



Google SRE Book is a strong starting point for things like incident response, metrics (4 Golden Signals)

DevOps Failure Modes

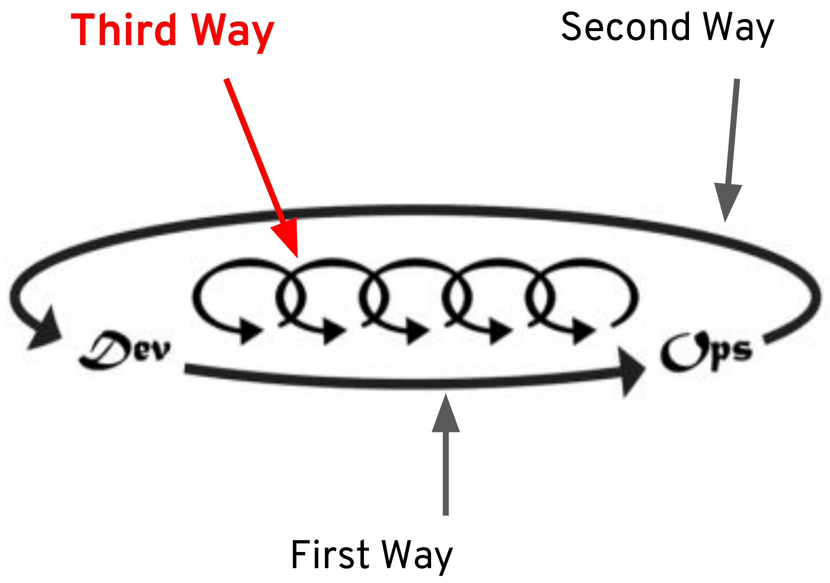
Delays => Larger Blast Radius => Instability

Expectation setting	Build vs Buy
Signal to Noise of Logs and Traces	Vanity metrics
Alert Fatigue	Building a distributed monolith
Cross functional team frustrations	Platform Field of Dreams
Just writing glue code	Copying Google SRE (or other high flyer)
Grassroots > Big Bang (CoE / Dojo)	Not measuring success by team

Red Hat

Where you want to get to

- Improvement of Daily Work
- Increased Automation
- Winning over the frozen middle
- Building a culture of continual experimentation and learning



Thank you

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat