CHICAGO
MARCH 4-7

INTERNET2
2024
COMMUNITY
exchange

Building a NetDevOps Team

Shannon Byrnes - Sr. NetDevOps Engineer, Infrastructure Systems & Software, Internet2
Dustin Mouton - Assistant Director, Network Services, University of Arizona
AJ Ragusa - Manager - Network Analysis and Control - GlobalNOC - University of Indiana
Mike Simpson - Director, Infrastructure Systems & Software, Internet2

March 7, 2024

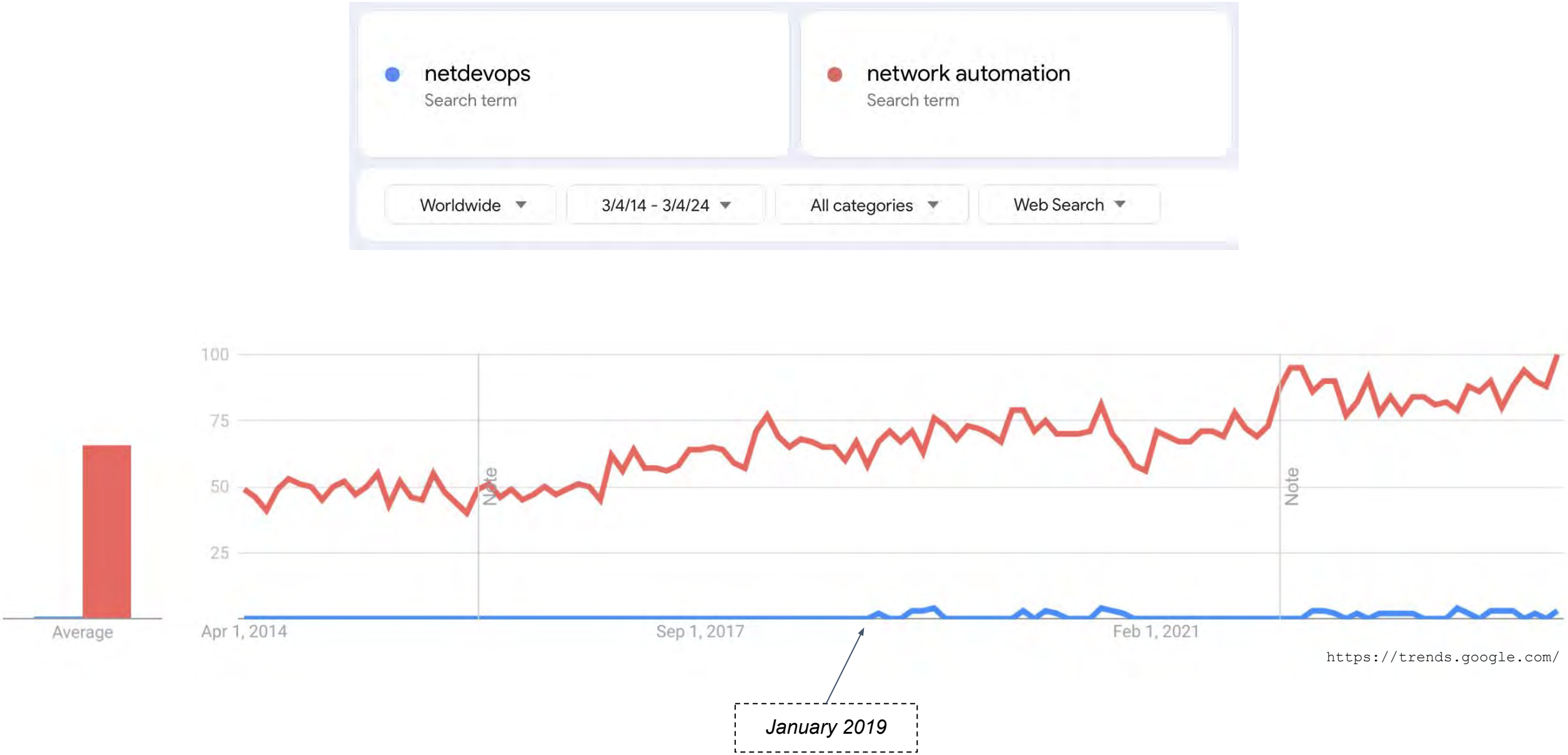
Agenda

- Reminder: What is NetDevOps?
- Stories:
  - University of Arizona with Dustin Mouton
  - GlobalNOC at IU with AJ Ragusa
  - Internet2 with Mike Simpson
- Q&A

What is NetDevOps?

Featuring: Broken-down Buzzwords

Shannon Byrnes, Sr. NetDevOps Engineer
Infrastructure Systems & Software
Internet2


netdevops
Search term
network automation
Search term
Worldwide
3/4/14 - 3/4/24
All categories
Web Search
100
75
50
25
Note
Note
Average
Apr 1, 2014
Sep 1, 2017
Feb 1, 2021
https://trends.google.com/
January 2019

Obligatory “What is a NetDevOps Team” Slide

A **network automation team that applies DevOps principles** to support and drive automation and orchestration.

Has skill sets like:

- Software Development
- Network Engineering
- Systems Administration
- Cloud Administration

Does things like:

- Infrastructure as Code (IaC)
- Version Control
- Continuous Integration (CI)
- Continuous Deployment (CD)

Not *exactly* like this. But close enough.

Why the hype?

The combination of these skill sets can result in network automation software that helps you manage your network safely, securely, and at scale.

Side effects include:

- Guarding against employee turnover.
- Confidence during security audits and cyber threats.
- Reduction in cognitive load across your team, increasing efficiency and morale.
- Reduction in manual work load across your team, giving time back to your employees to do greater things.
- Career growth opportunity.
- **Mitigated or removed problems that exist when trying to automate your network without dedicated staff.**

A Loaded Example: An attempt at removing the black box.

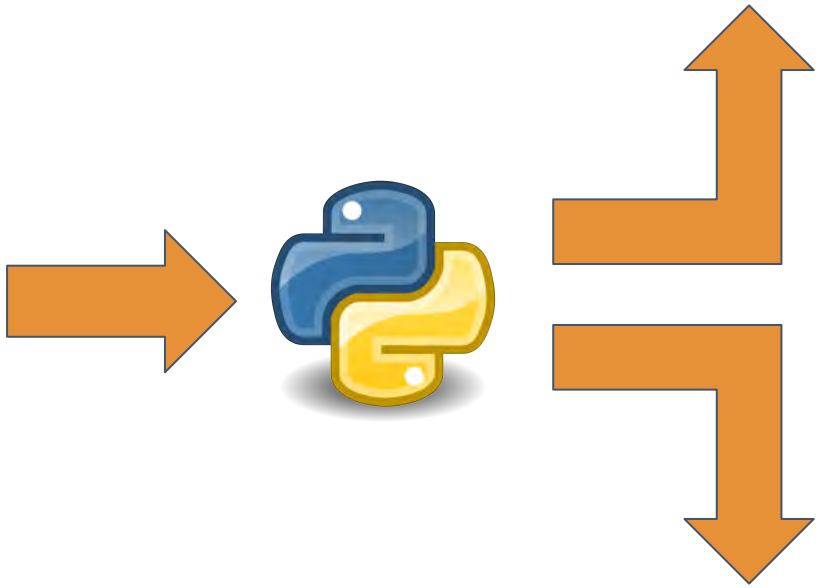
I Spy:

1. A software developer
2. A network engineer
3. “Infrastructure as Code”
4. “Version Control”

```
Router(config)# router bgp 65001
Router(config-router)# neighbor 192.168.1.2 remote-as 65002
```

CISCO

```
bgp_configuration:
  - as_number: 65001
    neighbors:
      - ip_address: 192.168.1.2
        remote_as: 65002
```



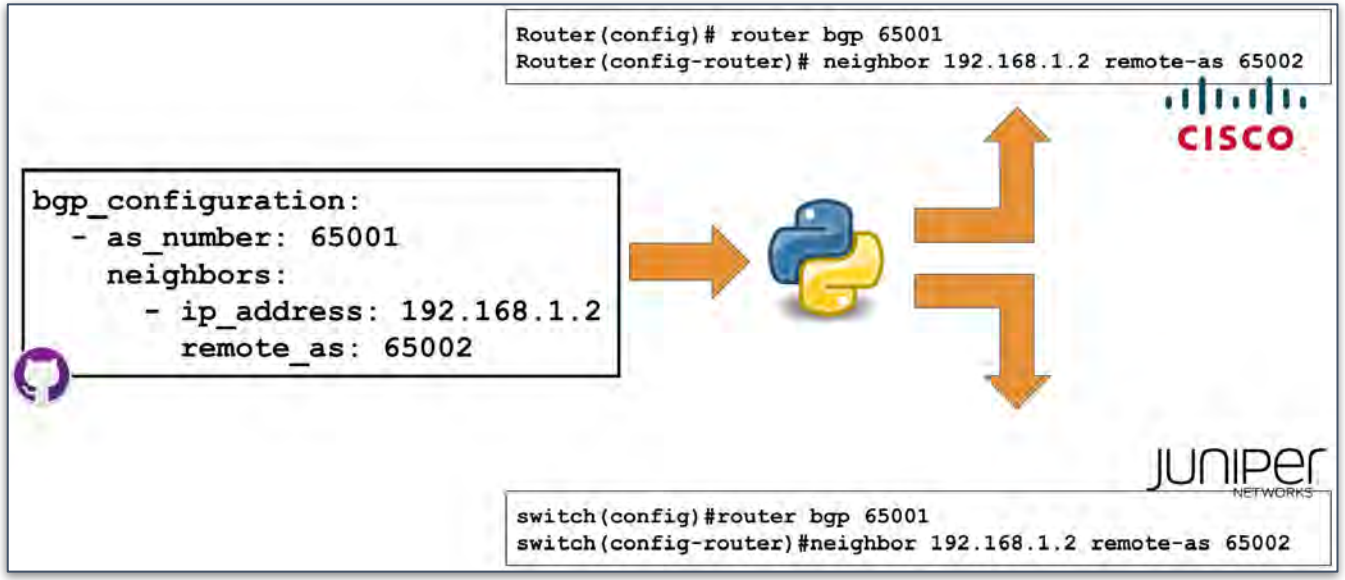
JUNIPER NETWORKS

```
set protocols bgp group my_group type external
set protocols bgp group my_group peer 192.168.1.2 remote-as 65002
```

An Oversimplification

I Also Spy:

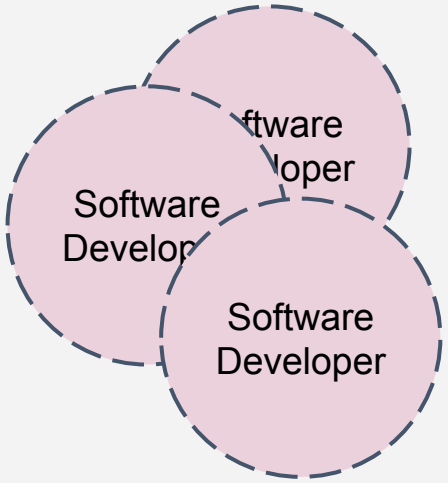
1. Continuous Integration
2. Continuous Deployment
3. Systems and/or Cloud Administration



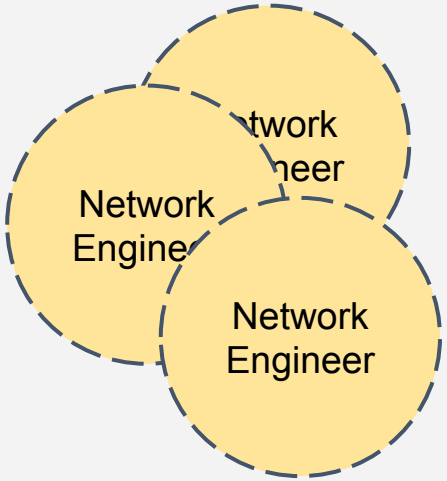
CI Says:
Let my engineer add a new BGP neighbor into the file, press a button to launch scripts, then press a button to deploy it onto the router, then press a button to verify all is well.

CD Says:
Let my engineer add a new BGP neighbor in the file… and then grab a cup of coffee.

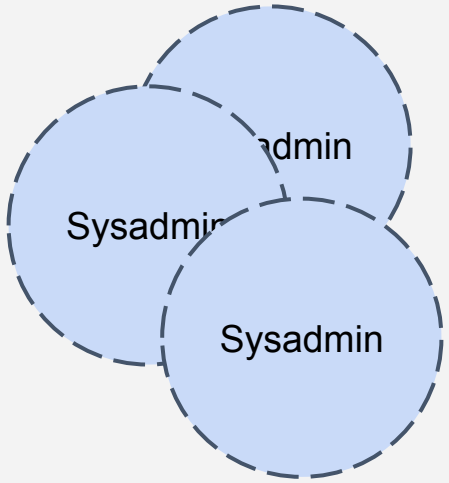
Software Development Team



Network Engineering Team



Systems Team



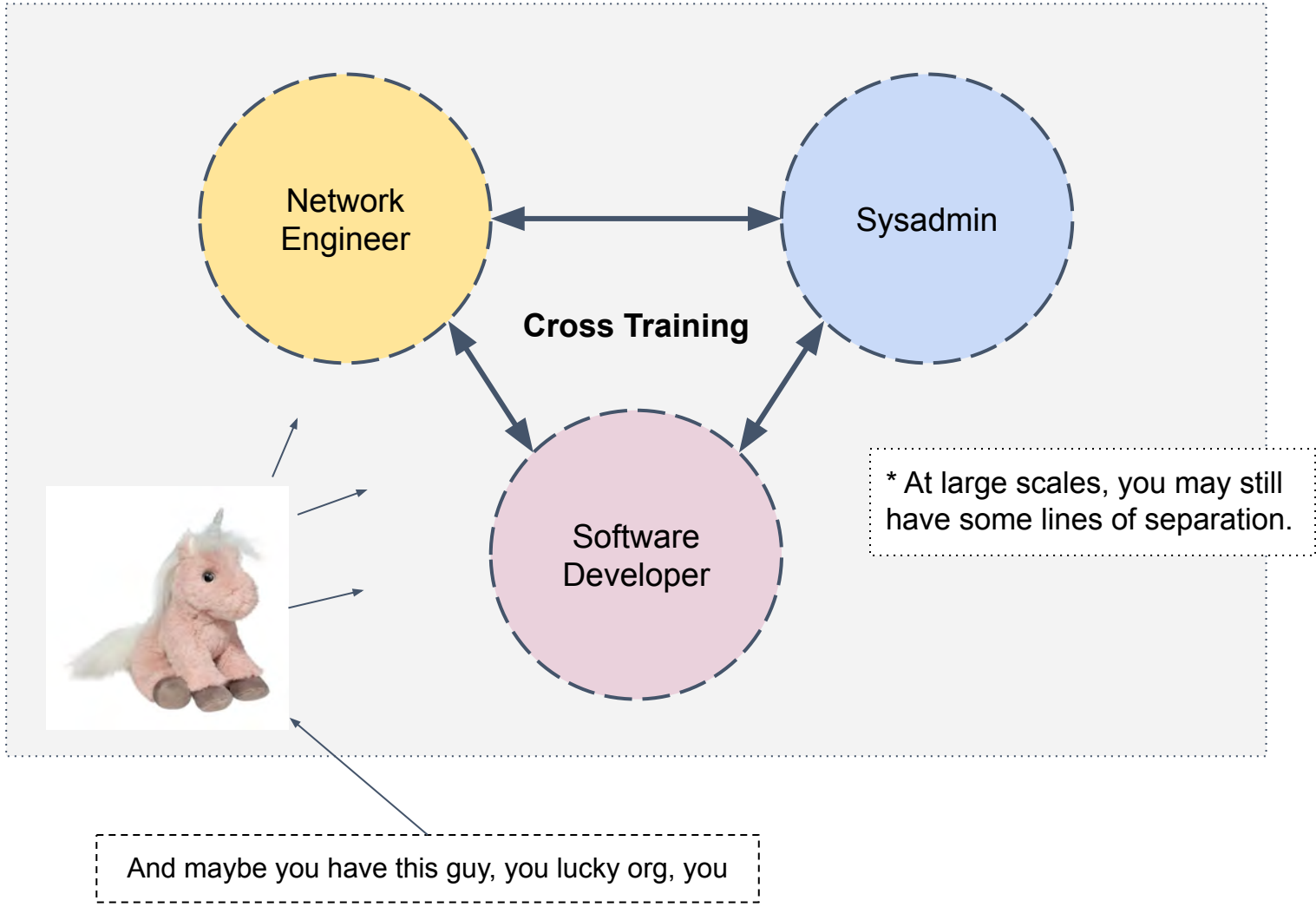
In the middle: Some managers and projects competing for prioritization

This is another over-simplification

- Varying team sizes
- Individuals with more than one skill set
- At very large team scales, may require reorganization.

<u>But most important: Remove silos and barriers to collaboration</u>

NetDevOps Team



University Information Technology Services

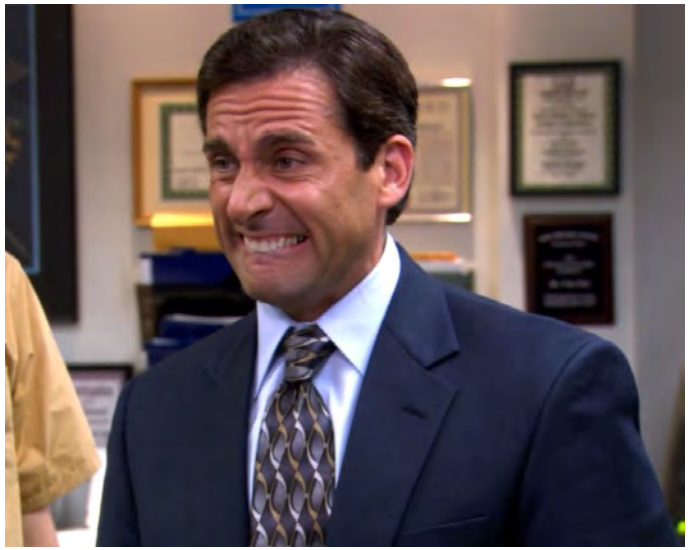
Building a NetDevOps Team

Dustin Mouton - Assistant Director, Network Services, University of Arizona

Campus Perspective - Before NetDevOps Integration

- Large Campus of 50k+ Students a Semester
  - 70+ different platforms (now over 90)
  - 1800+ wired devices (now over 3000)
  - 9000+ access points (now over 11000)

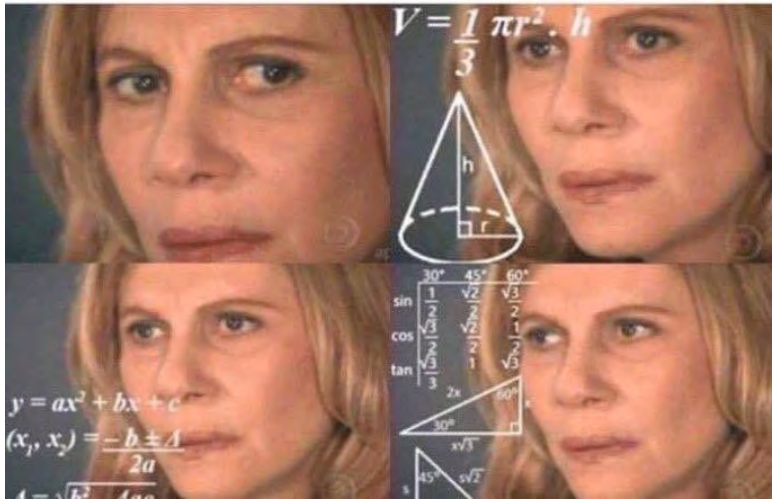
- Challenges
  - Security
  - Diversity
  - Scalability
  - Global configuration consistency



HELP!! Where Do I Start!!

- The scale of campus infrastructure was becoming too much.
- The idea of a network engineer who programs was not foreign. Old scripts and software lying around that no one understood, written by a retired engineer.
- “A programming person sounds like a good idea for our backfill…”

Note: NetDevOps was not a well-known concept at the time.



Personnel…Who?…Dilemma

What are the options?

- Do we repurpose Network Engineers?
- Do we force Network Engineers to take on additional responsibilities?
- New hire for (what would be) a NetDevOps role?
  - Who the heck is going to fund this?

We don’t want this.

Attempt #1 at a Network Automation Team:
Adjusting the Backfill Position

1. Determined specific goals for the position
   - Automate device image upgrades
   - Locate underutilized infrastructure
   - Standardize network configuration
   - And other repetitive (boring) work

2. Rewrote the entire job description
   - Networking was not the priority
   - Added buzzwords like Programming and Automation

Things that went well

- Foundation of a Source of Intent
- Major accomplishments
  - Automated image upgrade tooling
  - Automated device onboarding into NetBox
  - Mass configuration changes
  - Project-specific configuration deployments, such as wired dot1x, IP helper migrations, config standardizations
- Leadership saw tangible benefits from reports and automation. Dramatically increased support.
- Maturity of network automation rose with time and effort.

Things that did not go well

- Automation engineers hopped projects without documenting or refining that was implemented.
- Not enough experience to provide educated time estimates. Many corners cut.
- Engineers doing automation required a lot of focused time, and could not maintain their original day-to-day duties. Workload ended up unexpectedly moving to other team members.
- Without mentorship, automation team lacked foundational knowledge on how to operate effectively. Ex., sprint meetings, effective usage of version control, code review, etc.
- The lack of a formal NetDevOps team was limiting.

Awwww nuts.

The Great 2022 Mass Exodus

- Lost four skilled staff members to significant career opportunities through professional development
  - Had operated independently
  - All left within a 1.5-month period
  - Documentation and tools became unmanageable for remaining Network Engineering staff.

Attempt #2 at a Network Automation Team:

1. Took all the lessons learned from #1
2. Form a ‘Real’ team
    - Propose new positions at different skill levels (Jr., Sr., Pr.)
    - Define a new pillar within Network Services to focus on NetDevOps
    - Utilize a current staff member to function as a team lead
3. Have new/current staff review current and previous tools (What can be deprecated vs. refactor)

How's attempt #2 going?

- What is better?
  - A well-defined and structured NetDevOps team. Code planning, creation, review, and deployment becomes well understood by many instead of one.
  - More refined training provided by the NetDevOps team to other Network Services staff.
  - Tools are easier to understand and use by the Network Engineering centric staff.
  - Reports to present to Senior Leadership are more easily generated and data collected is consistent.
  - Plenty more!

Goals and Lessons Learned

Culture Shift - A must!

- Transforming as a team
- Demonstrating benefits
- Creating a new mindset
- How do we get the bad things to not happen again?

Staff Training, Development,...oh, and hiring…probably

- Outline plans to train current Network Engineers
  - Dedicated time for training
  - Knowledge Transfer among team
  - Set aside training budgets
- Continued education for developers and engineers.

Case studies and Future trends

Engaging your Senior Leadership

- Provide examples of organizations or universities that have succeeded in deploying NetDevOps initiatives.

- Highlight benefits those organizations have achieved
  - Increased agility
  - Reduced downtimes
  - Improved user/network engineer experience

- Discuss emerging trends in network automation and tools that can further increase team productivity and efficiency.

Building a NetDevOps team

How we built a Network Automation team inside of the GlobalNOC.

AJ Ragusa - Manager - Network Analysis and Control

2019/2020 the GlobalNOC Renewal Program

- In 2019 and 2020 the GlobalNOC Decided that one of the key tools that was needed was a Network Automation Suite. To accomplish this Luke (Executive Director), David Ripley (Director of Systems Engineering) and myself (not even a manager at the time) were tasked with building a Network Automation team.
- The goal for us was to evaluate existing tooling (from Ansible to Python, Perl to NSO) to design and build a set of tools to drive the GlobalNOCs Network Automation solution.
    - There were lots of different decisions here but mostly those aren't of interest for this discussion :)
- The good news, we had a lot of experience to build on
- We also had a lot of existing frameworks, policies, and other tools to build on
- Need a goal and a Vision

We Operate different from single network NetDevOps teams

- GlobalNOC supports many networks of many different sizes and types
- Our tools must be able to support these different networks
  - EVPN, ISIS, OSPF, MPLS, etc…
  - Cisco IOS, Cisco IOSXR, Juniper, Arista, HP, etc…
- What works for us might not work for a single University
- My focus is going to be mostly on building a Network Automation team however it is not possible to do it without all the other supporting teams

The beginning... kinda - Sherpa - 2008

GRNOC Sherpa: Vlan Provisioner

Step 2 of 6: Please provide the first endpoint, called the A End

/ Sherpa Home / 1: Add Vlan / 2: A End / 3: Z End / 4: Path / 5: Root Bridge / 6: Provisioning

Summary

Tag : 121

BW: 3.2 Gbps

Desc: test

Root:

Ent: PSC

Endpoints

A: pitt.layer2.nlr.net -

Z:

Select a node *(green dot)* on the map, then an interface

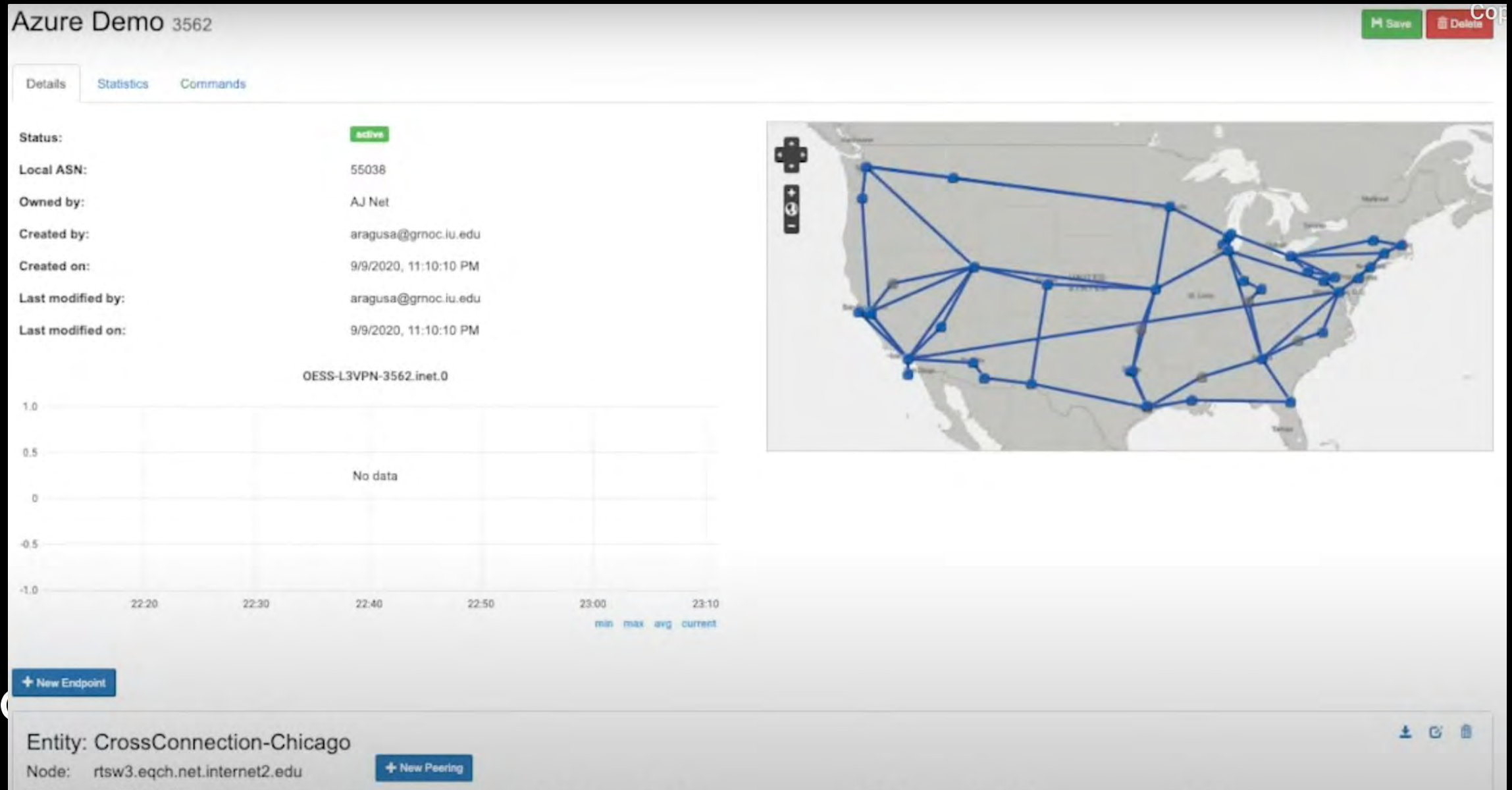
Interface	Description
GigabitEthernet6/1	
GigabitEthernet6/2	SUP 6 to Racklan f0/5
GigabitEthernet9/14	
GigabitEthernet9/8	
GigabitEthernet9/5	PSC
GigabitEthernet9/2	PSC-PNW-TR
GigabitEthernet9/12	
GigabitEthernet9/4	PSC
GigabitEthernet9/3	"3ROX-ESNET [NO-MONITOR]"
GigabitEthernet9/24	
GigabitEthernet9/17	
GigabitEthernet9/18	
GigabitEthernet9/7	NLRview PC (to wash) [NO-MONITOR]
GigabitEthernet9/6	available port with SX sfp loaded [NO-MONITOR
GigabitEthernet9/20	
GigabitEthernet9/21	
GigabitEthernet9/19	
GigabitEthernet9/1	PSC-NEF

Next Action:

Proceed to Step 3: Specify Z endpoint

GlobalNOC
at Indiana University

OESS - 2012-2023



Where to start - the people

- We already had a team working and running OESS basically full time with assistance from others. That will make up the core of the team, with AJ as the manager.
- The NAP (Network Automation and Performance (yes we did perfSONAR as well)) would be 4 people and a manager
- 3 people were already essentially already filling these roles so we only needed to hire 2
    - This is still a long process (Syseng was hiring several positions at this time) in total it took almost a year to fill the team

Hiring

- GlobalNOC Syseng teams generally are DevOps teams
  - Focus on both operational support and development skills
  - For hiring we try and find well rounded individuals who have the basics of both Software Development and Linux Administration skills
  - Network knowledge is **NOT** a requirement
  - Computer Science degree is a must
  - Most of our applicants are IU Computer Science graduates (masters / bachelors)
- Our interviews:
  - Software Development questions (what is a thread)
  - Linux Administration questions (how can you tell what process is using up memory)
  - White board software development (write an infinite loop)
  - Linux Administration practical (you could login to this server yesterday, but today you can not how do you troubleshoot)
  - Soft skills (you have a disagreement with your manager how do you proceed)

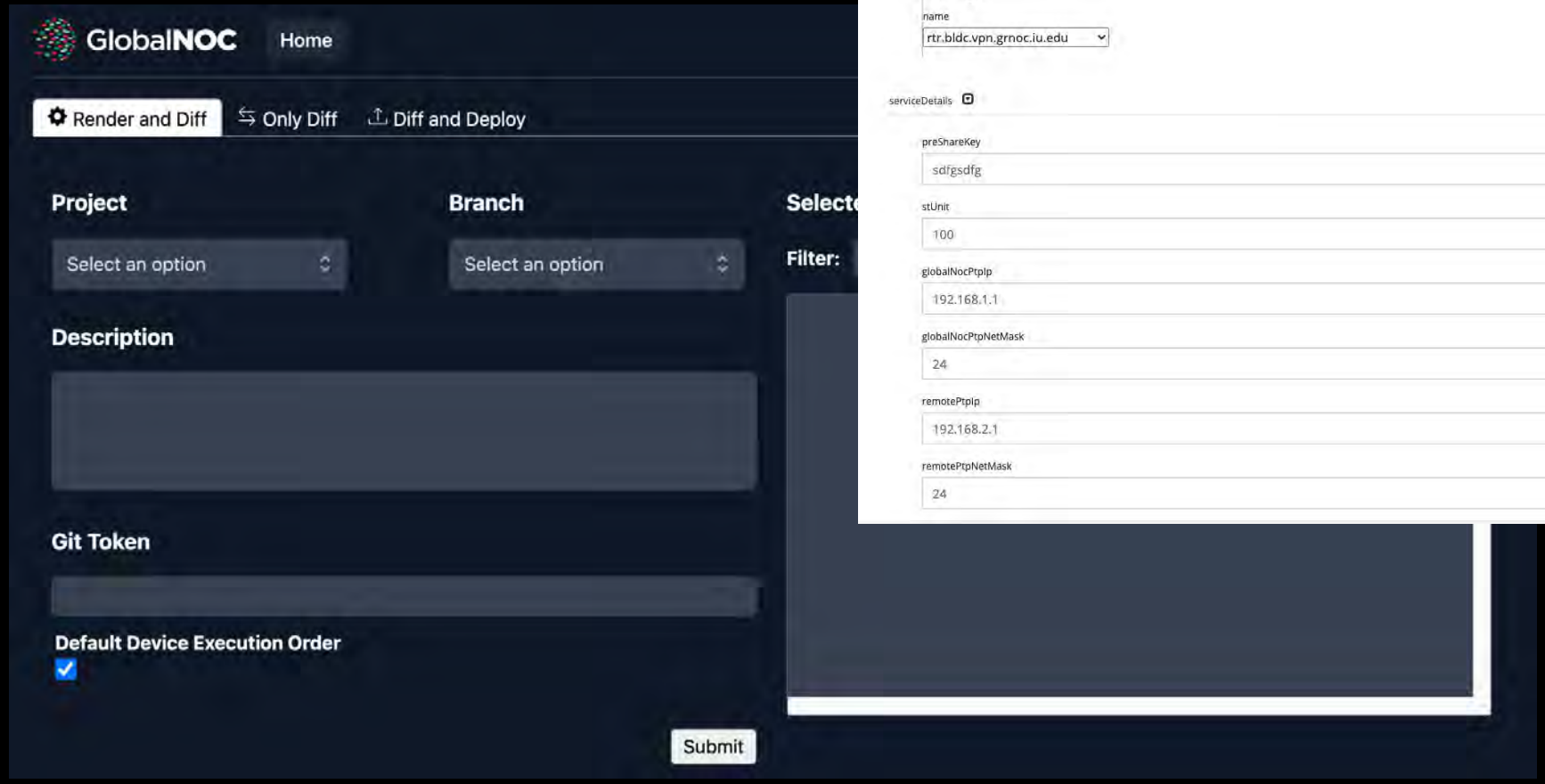
Project and Vision

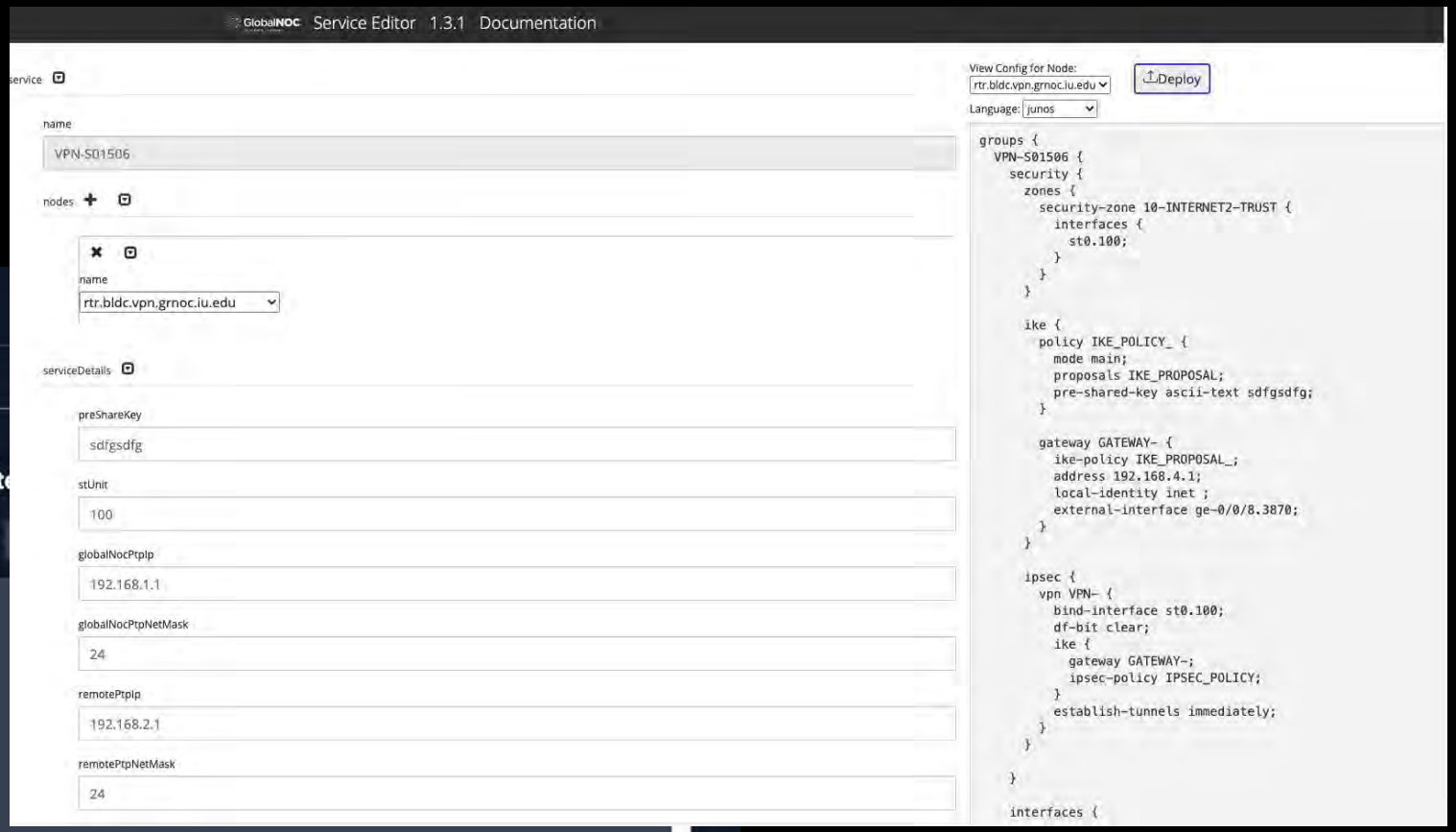
- Previously AJ had played with some basic automation using Ansible and GitHub to automate the network, this was our framework to start with. The first step was to define the application (build a set of requirements and then a design)

Working with Network Engineers to understand what they want

- During the Requirements and Design phase (early in the project start) interacting with the network engineers was critical to make sure we had the right set of requirements. GlobalNOC had also hired a dedicated Network Automation Engineer (Grover Browning) to assist (essentially filled the SCRUM “Project Manager” role for us.
  - Grover helped communicate with other engineering teams to help us gather requirements
  - Grover was also a test user for us
  - Grover also helped customers “onboard” to our tools one they were available to them
- The Network Automation Engineer role was critical to the start of these tools, but was temporary, lasting two years. However this role had great impact for our customers to get the most of our the tools quickly.

GNAT/GSCS 2019





Policies and Consistency for our Engineers

- We try to have basic policies to protect the networks and the engineers
  - Code Policies
    - No committing to Master/Main
    - Code Reviews by the rest of the team
    - Requirements and Design documents for “large” features/bugs or new tools
    - All commits require a ticket number
    - All development happens in a branch
  - Service Policies
    - 3 business day notifications for maintenances (Syseng and Neteng)
    - 24x7 On-Call with Escalation
    - AARs for high-impact / long duration outages (or upon customer request)
    - Changes for changes to actually be pushed to the network

If the Software Engineers don’t understand networks how can they automate a network?

- We rely on more senior engineers to understand the basics of networking
- We rely even more on the Network Engineers to know the networks they are working on
  - Since we can’t understand and keep in our head all of the different networks we really do need the Network Engineers to help us understand the technology
  - The hardest part is really communicating between Syseng and Neteng
- This worked for smaller more dedicated projects like Sherpa and OESS however trying to do this automation across all of our customers led us to a new “class” of engineer
- The Network Automation Engineer was extremely helpful with dealing with these issues but was only a temporary position to help us get off the ground we needed something else
- **Software Super Users Group**
  - Network Engineers from across the teams who specialize in the GlobalNOC tools
  - Are interested in learning Network Automation skills such as Ansible
  - Bridge the gap between Syseng and Neteng

Software Super users - Experts at GlobalNOC tools and their Networks

- Software Super Users are a class of Network Engineer inside of the GlobalNOC. Each team provides a Network Engineer to the team.
- This person interacts with Systems Engineering to get assistance using our APIs, writings scripts, and is the “expert” on the GlobalNOC Tools
- They also provide uses cases and problems for the systems engineering teams to solve
    - New features in existing tools
    - New workflows
    - Help script and clean-up data on the network
    - Understand the network needs (they are a network engineer)
- This role is critical to help the Network Automation team implement and improve our tools to work with each of our network customers

Network Operations Automation vs. Configuration Automation

- One request from Network Engineers was to assist with troubleshooting network outages. This lead to the Network Troubleshooter tool
  - This was a great example of working with the Network Engineers to fill a need to make their jobs easier
- The Network Automation team was able to quickly pick up this tool, work with the engineers to gather requirements and put out an Alpha of the tool in 2 weeks.
- Quickly able to take end-user feedback (Network Engineers and the Service Desk) to have a Beta of the current tool in another 2 weeks
- Fully deployment available for most GlobalNOC Customers in less than 2 months
  - While continuing with the operational and development of our other applications in a small team

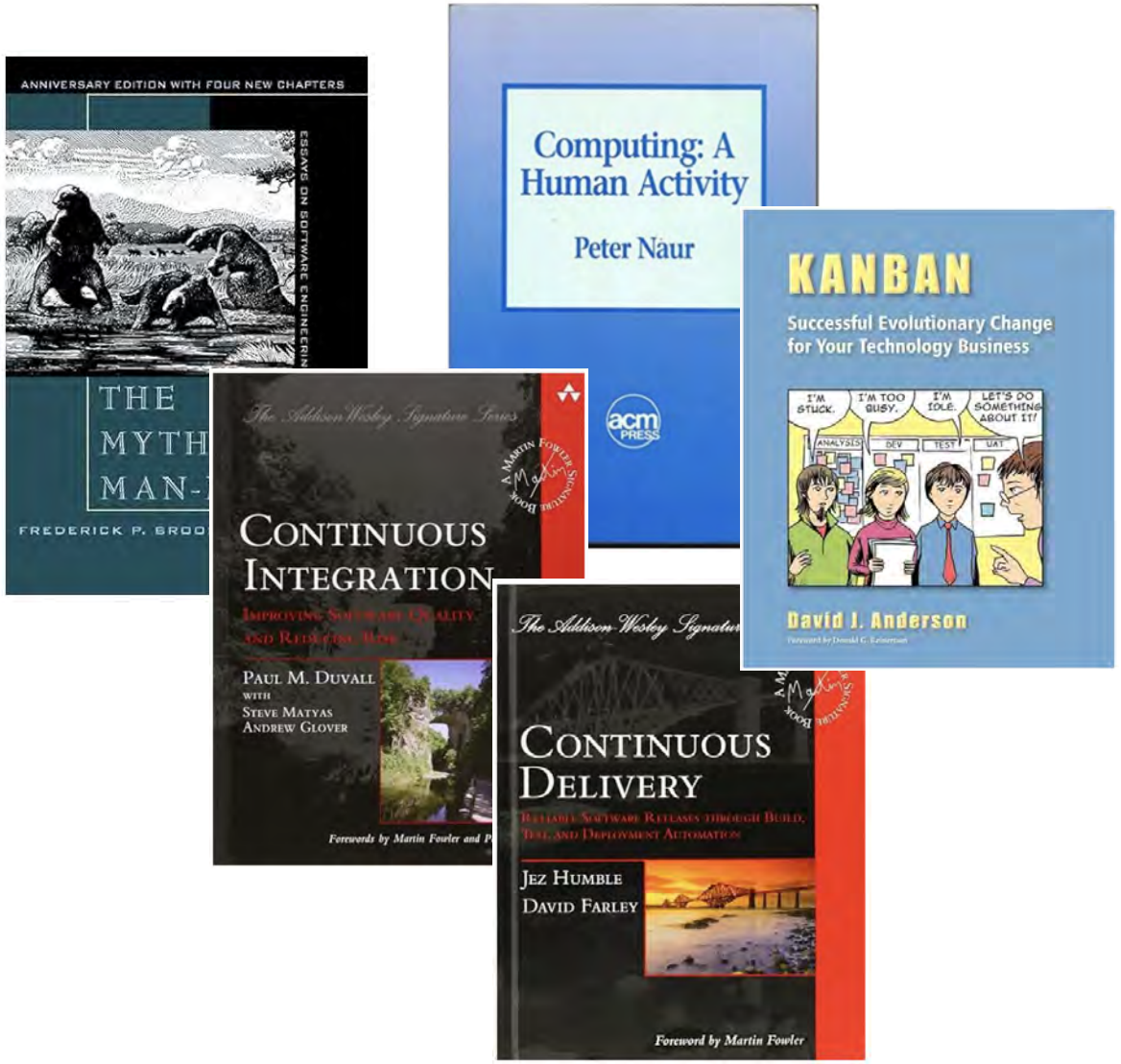
Bringing in the Service Desk for Network Automation Operations

- As we continue to improve our Network Operations Automation we have now started to include the Service Desk, by working with their manager, and SSTs for networks to improve the functionality of our Operational Automation tools for them
- We are still learning how to do this and improvements will continue to come to these processes
    - We have started embedding engineering dedicated people into the Service Desk (for example with ServiceNow) and we expect this to grow over time to improve our communications and interactions
- One of the future features for the Troubleshooter is the ability to have a “run book” with separate instructions for Service Desk users and Network Engineers allowing the Service Desk to start the troubleshooting process before they call a Network Engineer

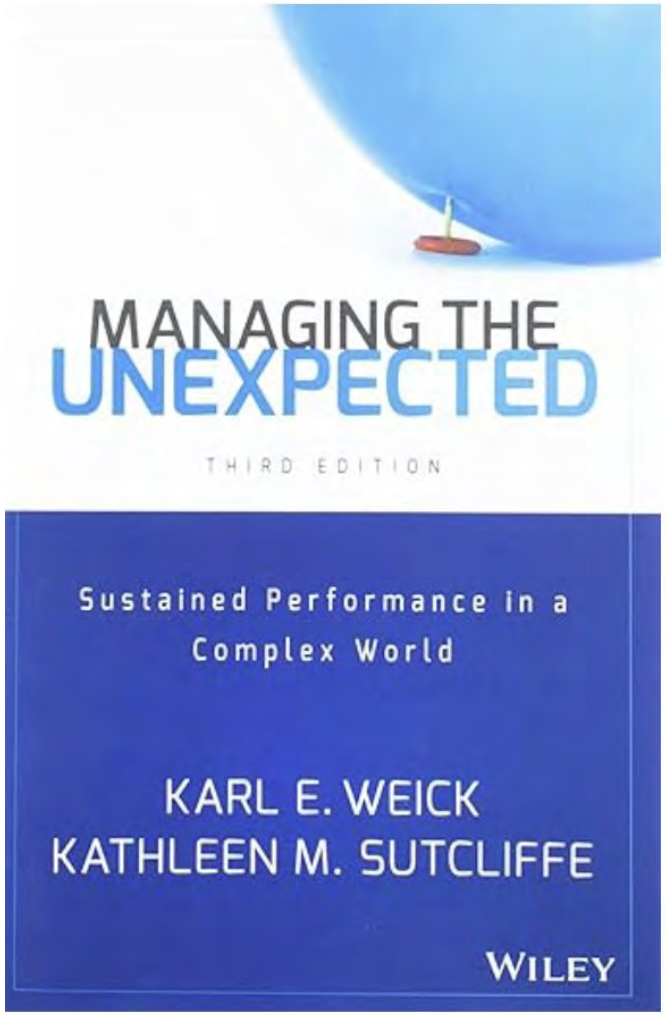
What I Wish You Knew About NetDevOps (Mike S)

None of this stuff is new.

- **Frederick Brooks**, *The Mythical Man Month: Essays on Software Engineering* (1975).
  - "The Surgical Team" essay
- **Peter Naur**, *Computing: A Human Activity* (1992).
  - "Programming as Theory Building" essay
- **Agile development** movement:
  - "Manifesto for Agile Software Development" (2001).
- **Continuous integration/continuous delivery** movements:
  - Paul Duvall, *Continuous Integration: Improving Software Quality and Reducing Risk* (2007).
  - Jez Humball and David Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* (2010).
- **Lean methodology** (going back to the 1950s):
  - W. Edwards Deming's "System of Profound Knowledge".
  - Ohno Taiichi's "Toyota Production System".
  - David Anderson, *Successful Evolutionary Change for your Technology Business* (2010).



Super last-minute addition @ 8:59am, Thu 3/7.



The absolute foundational requirement:

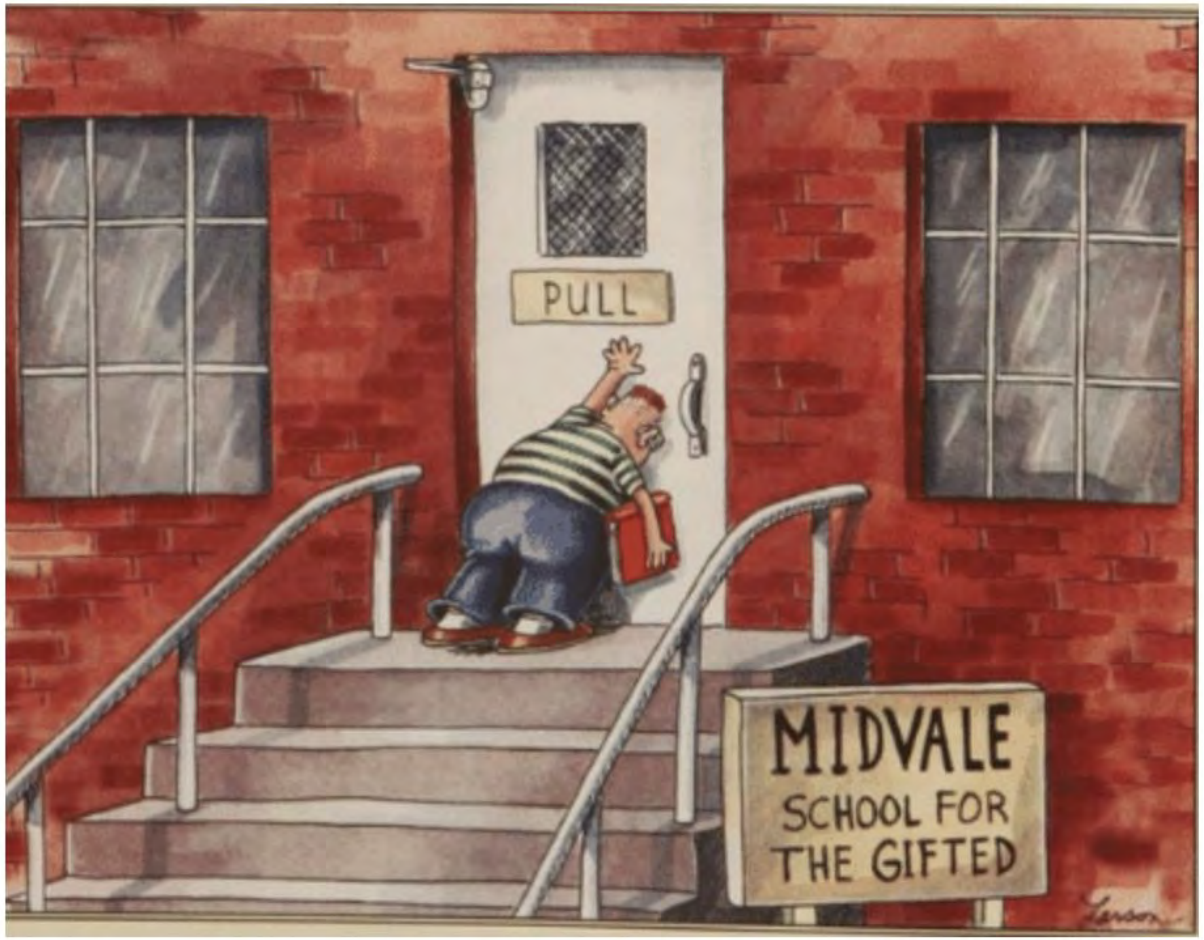
BE WILLING NOT TO DO DUMB THINGS.

- **We do Dumb Things all the time.**

- **Specific definition of a Dumb Thing:**
  - Doing something the way we've always done it …
  - … Despite the fact that the entire context and environment of doing the thing is completely different than when we originally decided to do it that way.

- **It is incredibly easy to do Dumb Things:**
  - **Because** it feels like a lot of effort to do something new, and it's incredibly easy to just go with the flow.
  - **Because** cognitive dissonance is a real thing, and the easiest person to fool is yourself.
  - **Especially** if you have a lot invested in Being Right.

- In a context of **intense, rapid, unceasing change** – imagine how easy it is to wind up doing Dumb Things all the time.



Here’s how you know you’re getting it wrong:

- You’re building cross-functional project teams made out of slices of people from your REAL teams, which are all based on occupational specialties.
    - INSTEAD, MAYBE: Build permanent cross-functional teams oriented around what it takes to completely deliver a service or suite of related services.

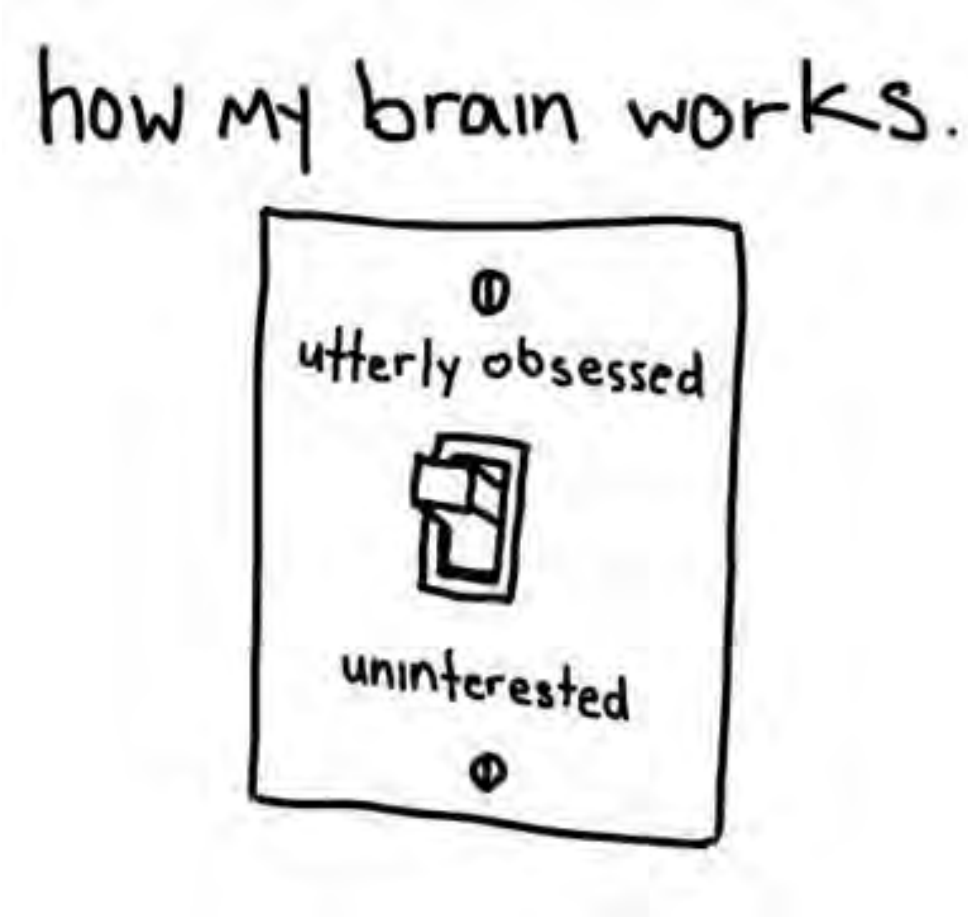
- You’re focusing on how many different chunks of work you can simultaneously cram through your team in parallel, because that just has to mean you’re being super-productive.
    - INSTEAD, MAYBE: Focus on how smoothly, efficiently, and predictably you can move a single complex piece of work from concept to completion.

- You’re diligently fixing problems with individual people to improve their individual productivity, because that will lead to overall increased productivity of the team.
    - INSTEAD, MAYBE: Work to understand and fix problems with the systems within which people have to try to get their work done.

Two more things to obsessively think about, part 1:

- **Thing One: Shorten your control loops to match the rate of change of the thing you need to control.**
  - Imagine a thermostat that turns on the heat or the air conditioning based on what the temperature was 12 hours ago.
  - Now imagine a hiring process that takes 12 months to produce a hiring decision, in an environment where expertise needs change every quarter.
  - Now imagine a tightly-centralized purchasing approval workflow that requires high-level clearance to buy rapidly-consumed office supplies.
  - Now imagine a high-functioning agile team implementing their continuous delivery toolchain for a rapidly-evolving software product where the EULA for every tool that the team wants to try out must be reviewed and approved by the legal department before it can assessed for utility.

- **Pragmatically:**
  - Minimize barriers to communication and coordination.
  - Drive locus of control out to the leaves of the tree.



Two more things to obsessively think about, part 2:

- **Thing Two: Pay attention to the only supply chain that actually matters.**
  - **Sourcing**: Build a big pipeline of opportunities to find awesome people.
  - **Acquisition**: When you find someone good, get them. Period. Without hesitation or opportunistic negotiation.
  - **Development**: Give them everything they need to get their job done, and to grow in confidence and expertise.
  - **Retention**: Give them reasons not to leave – losing their health insurance doesn't count.

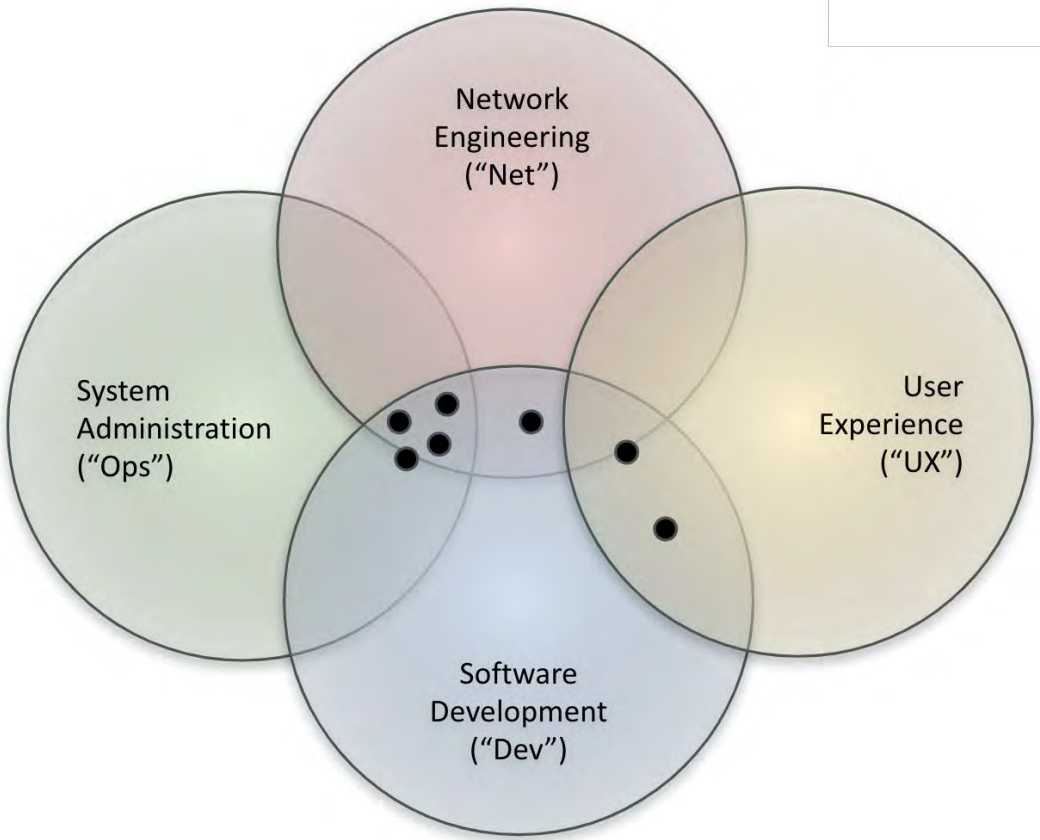
- **Pragmatically:**
  - People are either assets or expenditures, pick which one you think they are and act like you believe it.
  - If you just thought to yourself, "people are expenditures": I give up, you win, go on with your life.
  - Otherwise, consider expending at least as much effort on figuring out the SADR stuff above as you spent on your last technology RFP process.

Last thing for real I promise:

- **One type of NetDevOps team could be:**
  - A couple of network engineers
  - A couple of system administrators
  - A couple of software developers
  - etc.

- **Another type of NetDevOps team could be:**
  - Someone who's worked as a network engineer and a system administrator.
  - Someone who's worked as a system administrator and a software developer.
  - Someone who's done network engineering and software development.
  - etc.

- **On the right:** a snapshot in time of the skill sets of the people on my team from a couple of years ago. So: right now, at this moment in time, I MUCH PREFER the second model over the first, even though it makes building the team a LOT HARDER.



Thank you!