



JOURNEY TO Network Reliability Engineering (NRE) and DevNetOps

James Kelly



ons

EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate

JUNIPER
NETWORKS

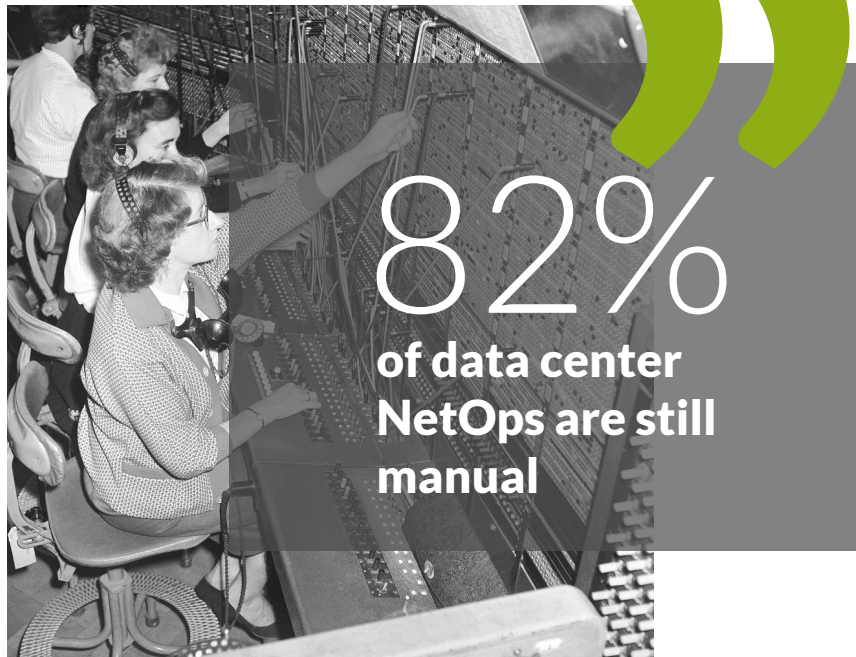
Engineering
Simplicity

NETWORK AUTOMATION



AUTOMATED NETWORKS

CONSUMPTION IS THE CONTEMPORARY CHALLENGE



Mistakes of the past

- Forcing engineers' rebirth as a "developer"
- The "pornography of tech" has left behind transformation of people & processes

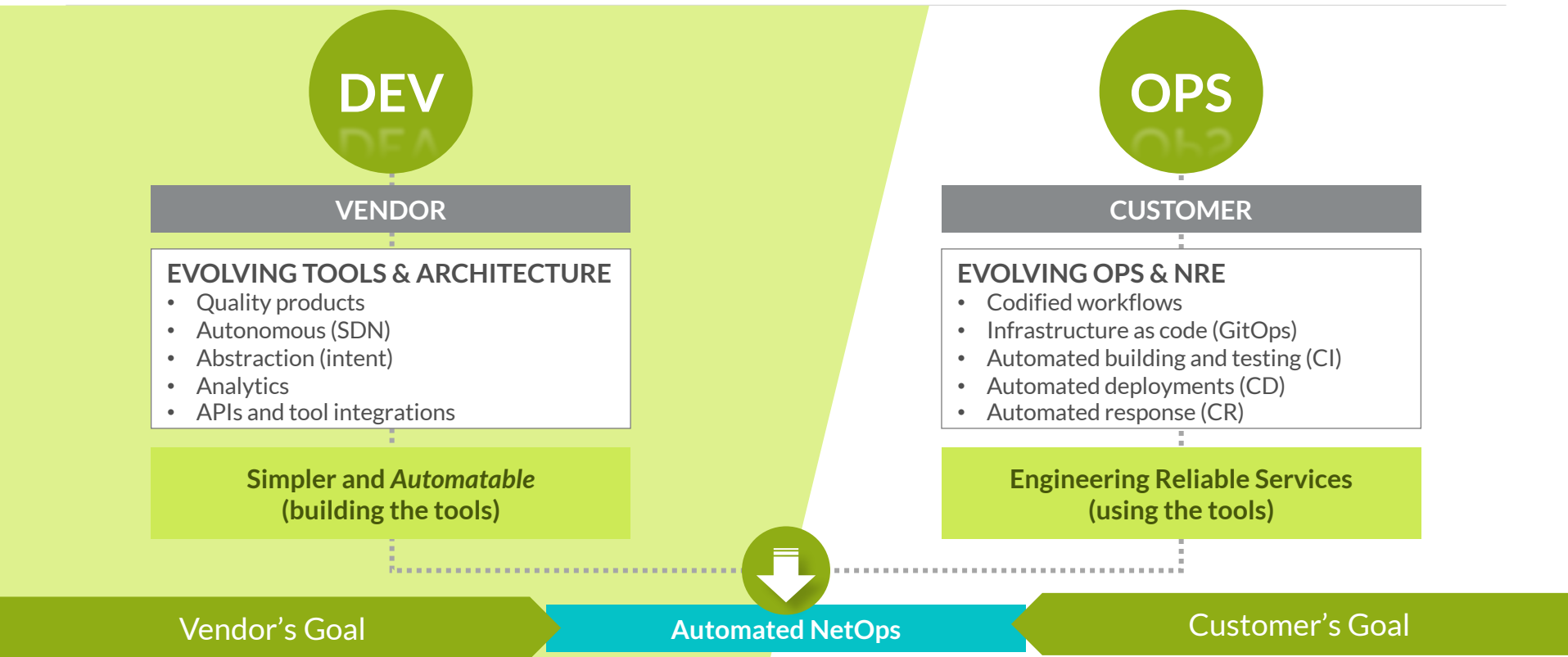
Ironically...
humans are the heroes
in the journey to automated NetOps

BRIGHT SPOT:

NRE

Network reliability engineer(ing)

TWO SIDES TO AUTOMATED NETWORKS



SPEED, AGILITY, EFFICIENCY &
OTHER SUCCESSES
ARE INCIDENTAL

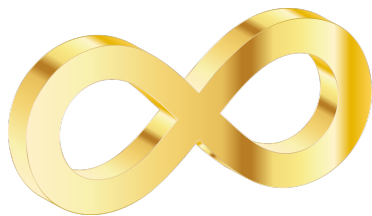
FOR THE NRE THAT HOLDS
RELIABILITY
PREREQUISITE TO OTHER
USEFUL ECONOMIES

FOR THE
NRE

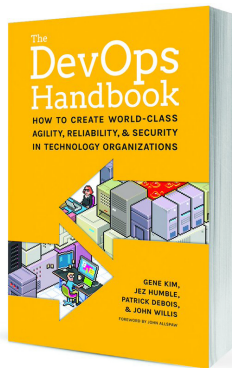
OPERATIONS
SIMPLICITY

IS CREATED THROUGH
ENGINEERING

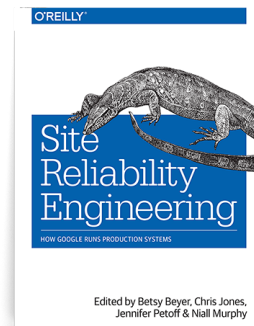
INSPIRATION



DevOps is Coined
2009



DevOps Handbook
2015



SRE is published by Google
2016

DEVNETOPS
OR
NRE?

WHAT IS (AN) NRE?



Defⁿ An NRE

The professional that implements network reliability engineering

Defⁿ NREing

Engineering an automated network to deliver measurable *reliability* (SLO/SLA of MTTF, MTTR, etc.) under measureable conditions (scale, rate of change, performance, etc.)

Defⁿ DevNetOps

Like NRE, *engineering* automated network, but more explicitly says:

- Take a developer (software engineering) approach
- The application of the approach is to NetOps
- Focus on shorter cycles and lead time in code-to-prod pipeline
- Our work begins in pre-production and follows CI/CD/CR

WHY N-RELIABILITY -E



1. Reliability is at the base of the hierarchy of needs

It's prerequisite to security, velocity, agility or efficiency.

2. Reliability must be ensured before acceleration

"It's not how fast you drive, it's how you drive fast"

3. Reliability forces us to automate and simplify

Encompassing NetOps goals: resiliency, security, metrics...

Higher-order DevOps principles...

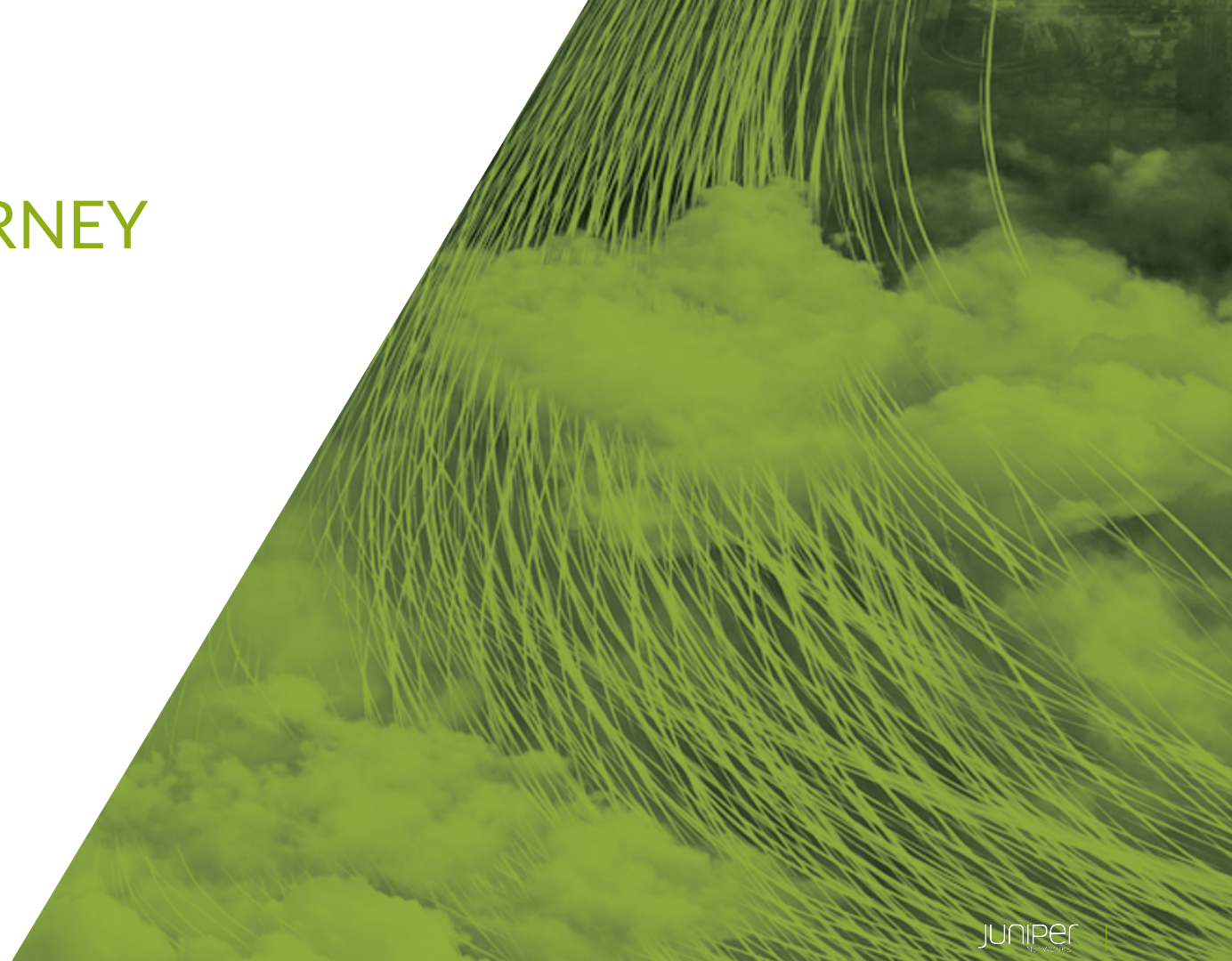
1. Eliminate toil and technical debt with automation
2. Truth and transparency. Management by metrics.
3. Allow for failure; Iterate and evolve with Agile; then triangulate...
4. Continuous improvement: turn local lessons into global ones
5. Continuous learning (kaizen)

WHY NR - ENGINEER

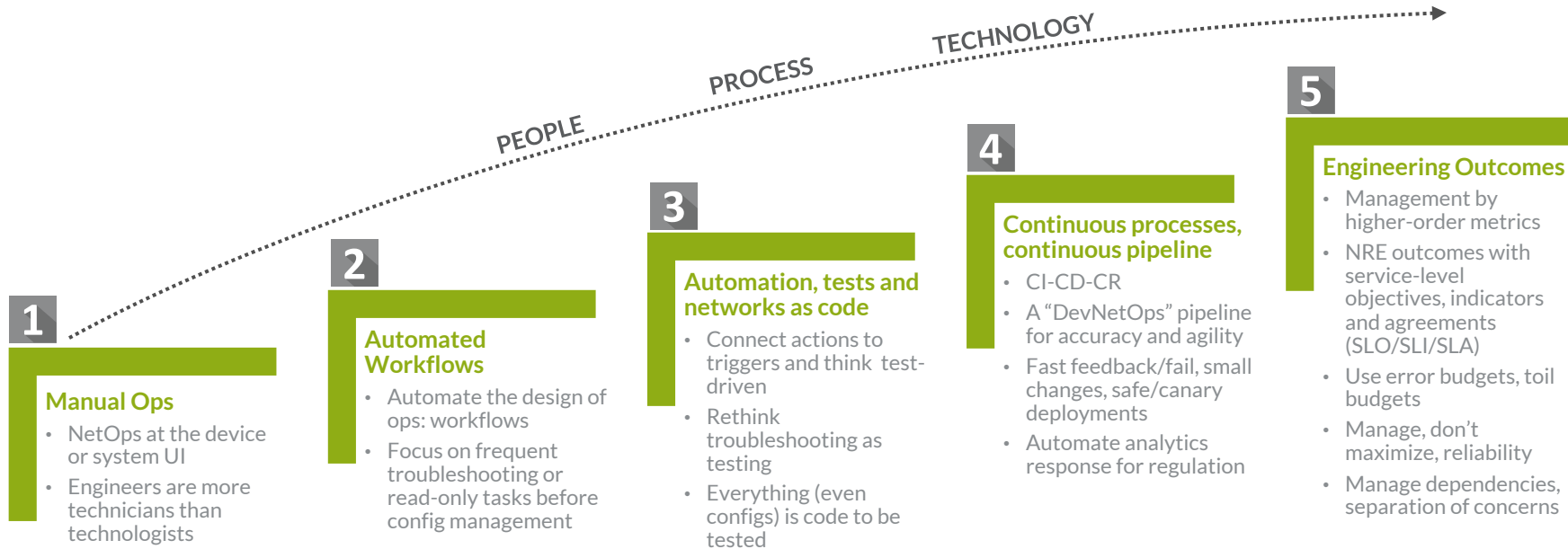


- 1. It's not just about network automation or technology:**
"Network automation does not an automated network make."
- 2. NRE focus sidesteps DevOps vs. DevNetOps confusion**
There are clear NetOps projects outside of software teams, but some confusion on terms remains. NRE is more straightforward.
- 3. Engineers are builders with structure and rigor**
Engineering picks up where vendors leave off
- 4. More to it than in-production NOC dashboards**
More creative, more satisfying, more money
- 5. By building, testing, stressing, staging what you build...**
you prepare for better operations and better outcomes

5-STEP JOURNEY



AUTOMATED NETOPS 5-STEP



DESTINATION



People:
Network Reliability Engineers (NRE)




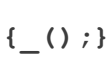





























































Process:
“DevNetOps” And NRE’ing




Technology:
Autonomous and automated NetOps

TECHNOLOGY LANDSCAPE*

STEP 1			+ STEP 2			+ STEP 3			+ STEP 4			+ STEP 5		
 Device CLI	 Product GUI	 Documents	 Code	 Markup/ Modeling	 Ansible	 git	 GitHub	 BitBucket	 Bamboo	 Jenkins Pipelines	 Spinnaker	 Optimizely	 VividCortex	 Catchpoint
 Workflow Docs	 Workflow pseudocode	 openstack mistral Mistral	 Chef	 Puppet	 Salt	 Robot	 Jenkins	 GitLab	 Travis	 GitLab Runner	 Build Kite	 Statistica	 Netsil	 NetBrain
			 OpenConfig	 RFC 6241 NETCONF	 Terraform	 Serverless	 Salt EDI	 StackStorm	 Open AI	 Tensor Flow	 theano	 tableau	 Spark	 MapR
			 NAPALM	 openstack heat Heat	 HELM Helm	 Prometheus	 ELK / Elastic	 splunk > Splunk	 ZUUL			 Chaos Monkey	 Istio	 Avalanche
						 Graphana	 ToDD	 InfluxDB						
 Products			 JET	 PyEZ	 Junos Scripting	 AppFormix Events	 JTI, OpenNTI	 Ixia	 JLABS	 Cloud CCL	 NITA	 HealthBot		

* It's not what you use, it's how you use it.

A DEVNETOPS PIPELINE



	TOOLING	PROCESSES	PEOPLE
Network as Code	Repos of config, secrets, artifacts + gitOps	branching, reviewing, pairing, Agile	Code skills (not necessarily programming)
Pipeline Orchestration	Pipeline CI/CD tools, test frameworks	TDD, measurement judgements	Build and debug skills, pipeline pros
Micro & Immutable Architecture	Baking deliveries for ZTP, vendor refactors	Small-step commits/deployment	Hands off CLI/TTY
Orchestrated Upgrades	ZTD, virtualization, labs, traffic draining	Staging and simulation, canary analysis	In-hours maintenance, maybe roll forward
Resiliency Design and Drills	Traffic generation, DoS, chaos monkey	Chaos windows, document limits	Force failure for understanding
Continuous Measurement	Big data analytics, ML, ITops integration	Incident playbooks, capacity planning	Management by stats, metrics, efficiency
Continuous Response	Auto-remediation, FaaS, predictive stats	Supervise self-driving	Drink tea, meditate
Continuous Improvement	Upgrades, features, fixes, changes	Record local lesson into global knowledge	Active open-mindedness, post-mortems



THANK YOU

JUNIPER
NETWORKS

Engineering
Simplicity