Introduction to

BY, CHINNAJEE RAO

About V. Chinnajee Rao

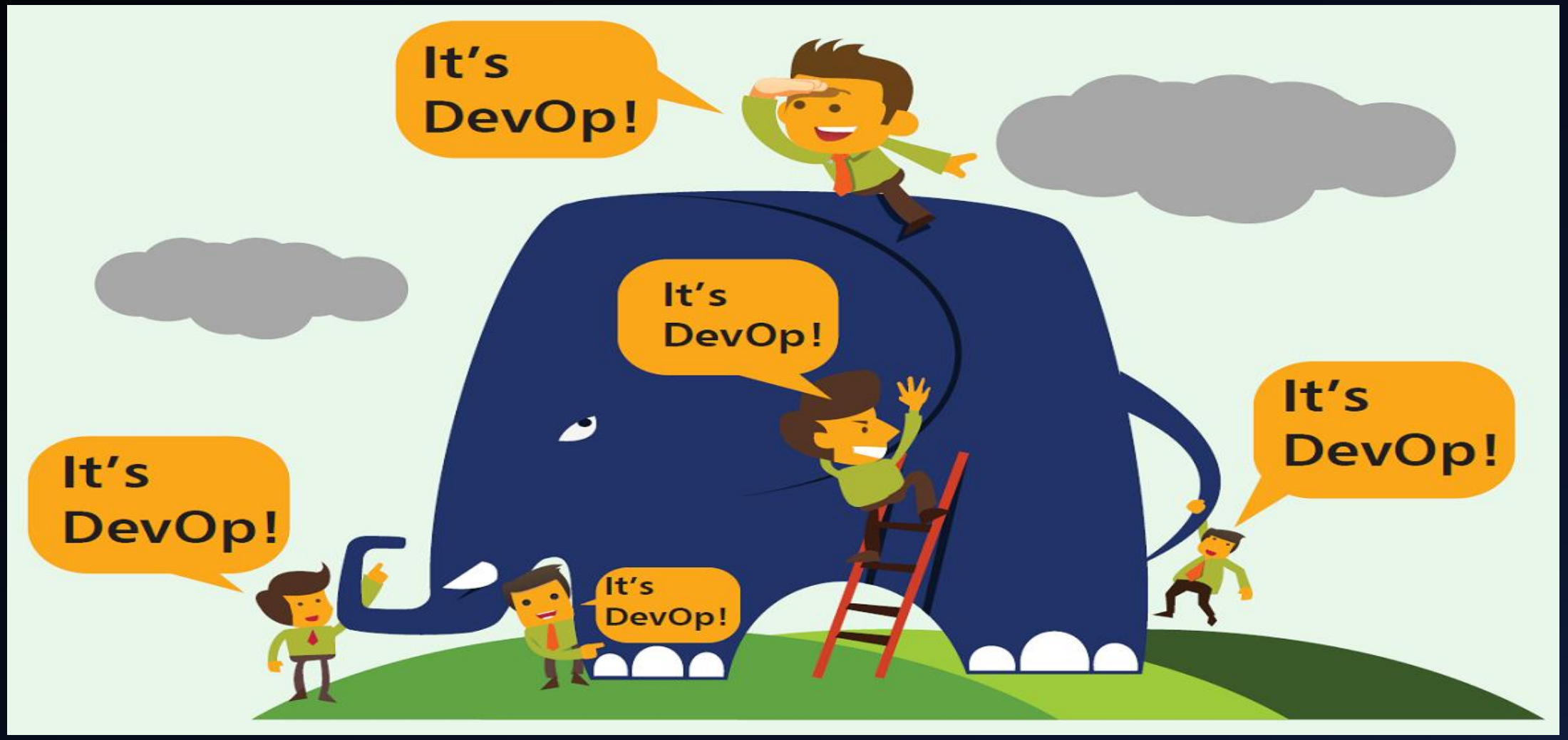
- IT Infrastructure trainer with 18 + years of experience
- Core – U & I
- Certified – DevOps, AWS, RHEL,
  Solaris , AIX, VMware

Contents

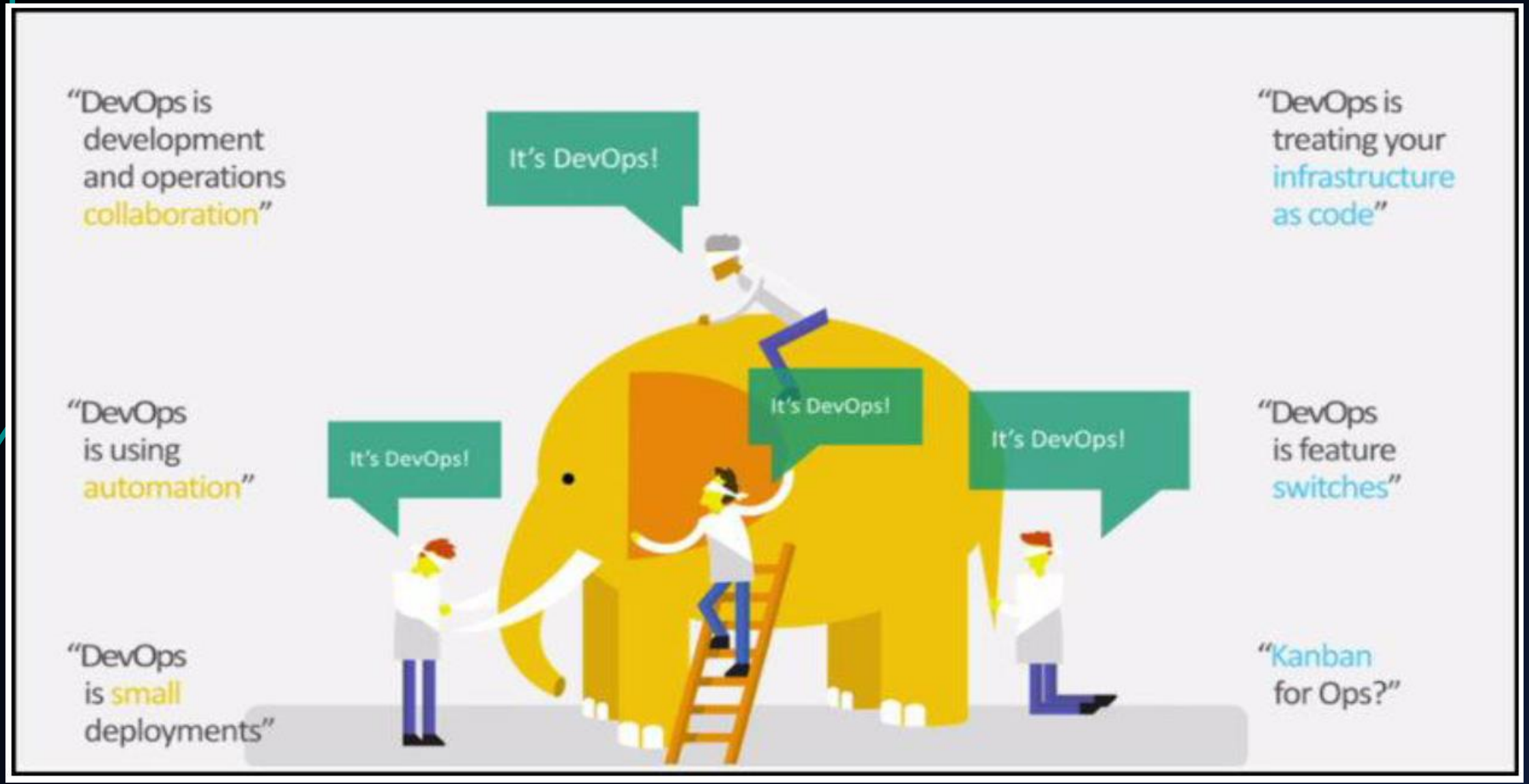
- Introduction
- What Is DevOps?
- Do We Really Need DevOps?
- Survey of DevOps Quantifiable Benefits
- How Does DevOps Work Anyways?
- Challenges and Problems Out There
- 7Cs of DevOps
- How Will You Benefit From DevOps Approach?
- Conclusion


DevOps

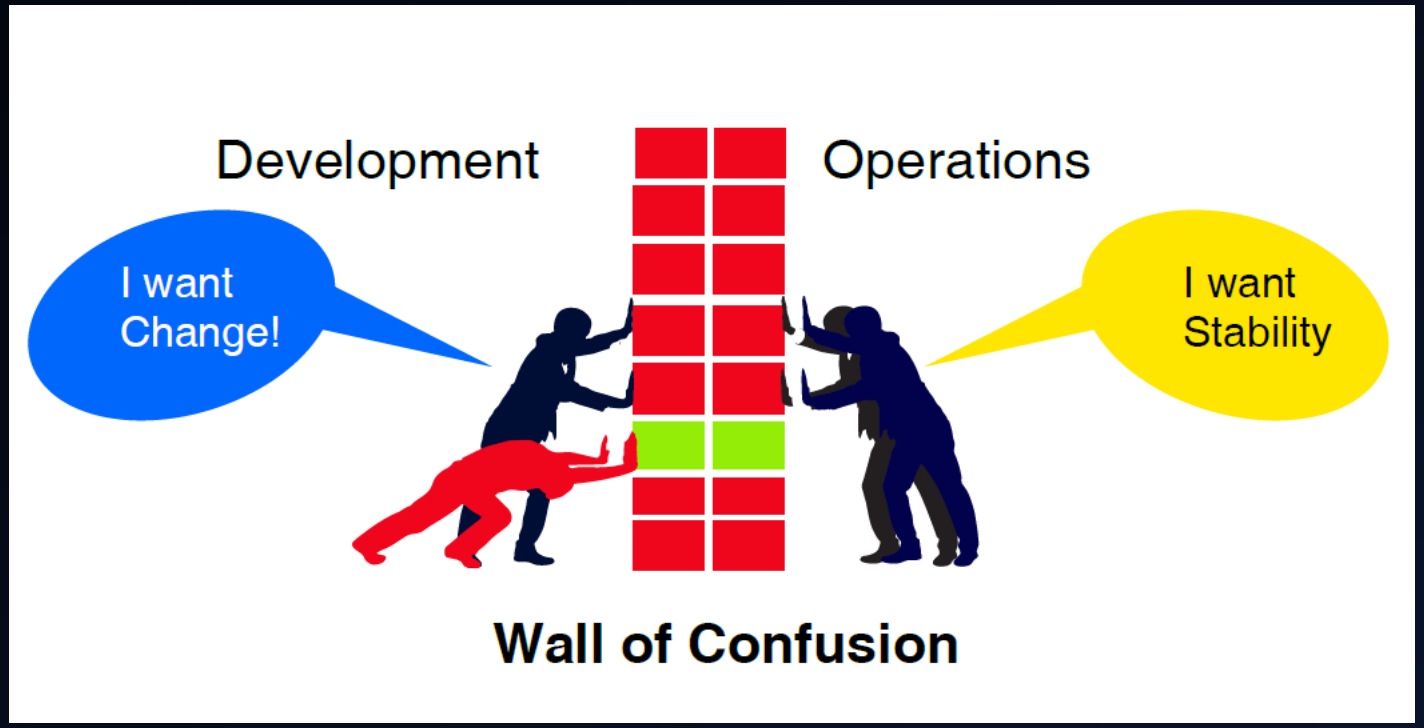
Why DevOps exist?



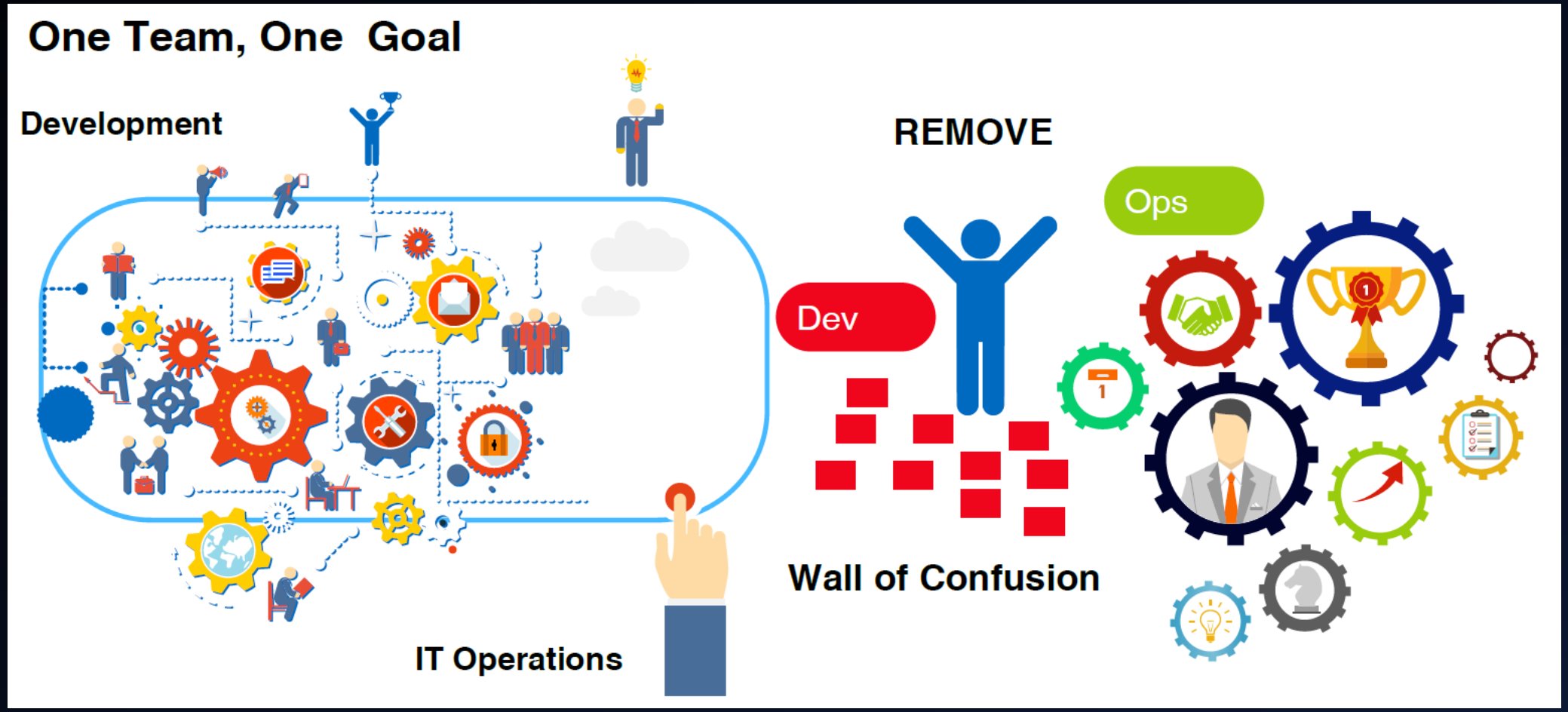
Why DevOps Exist



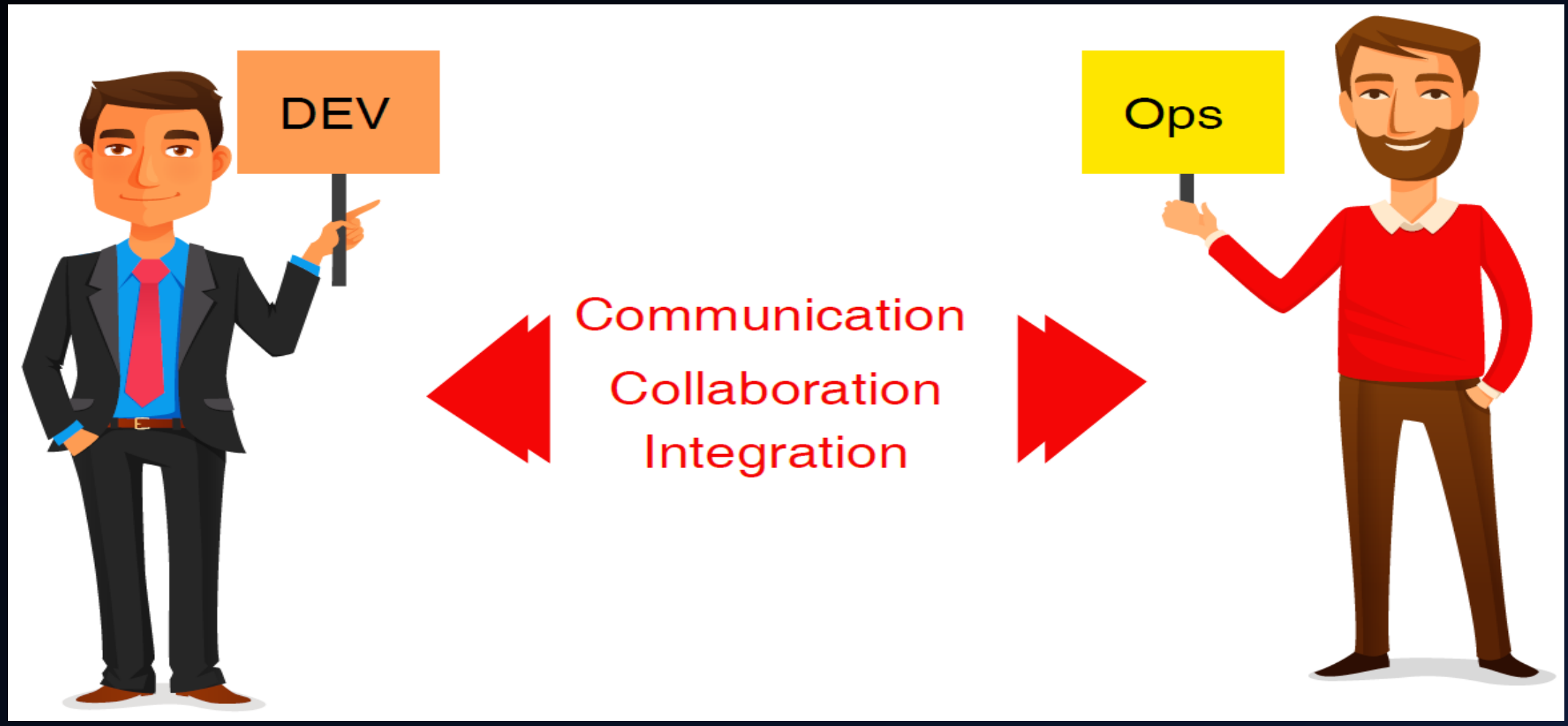
Do we really need devops?



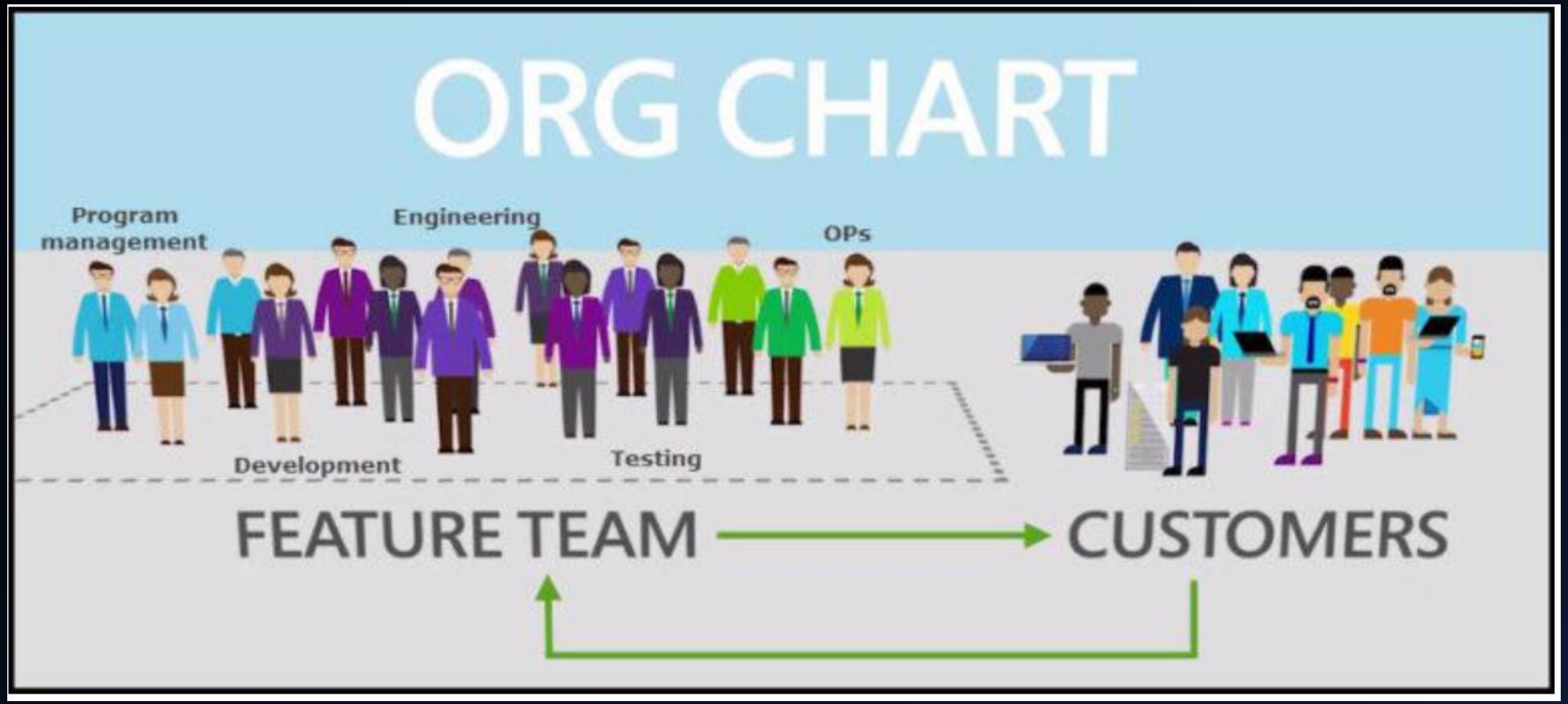
DevOps break down the walls between development and operations team



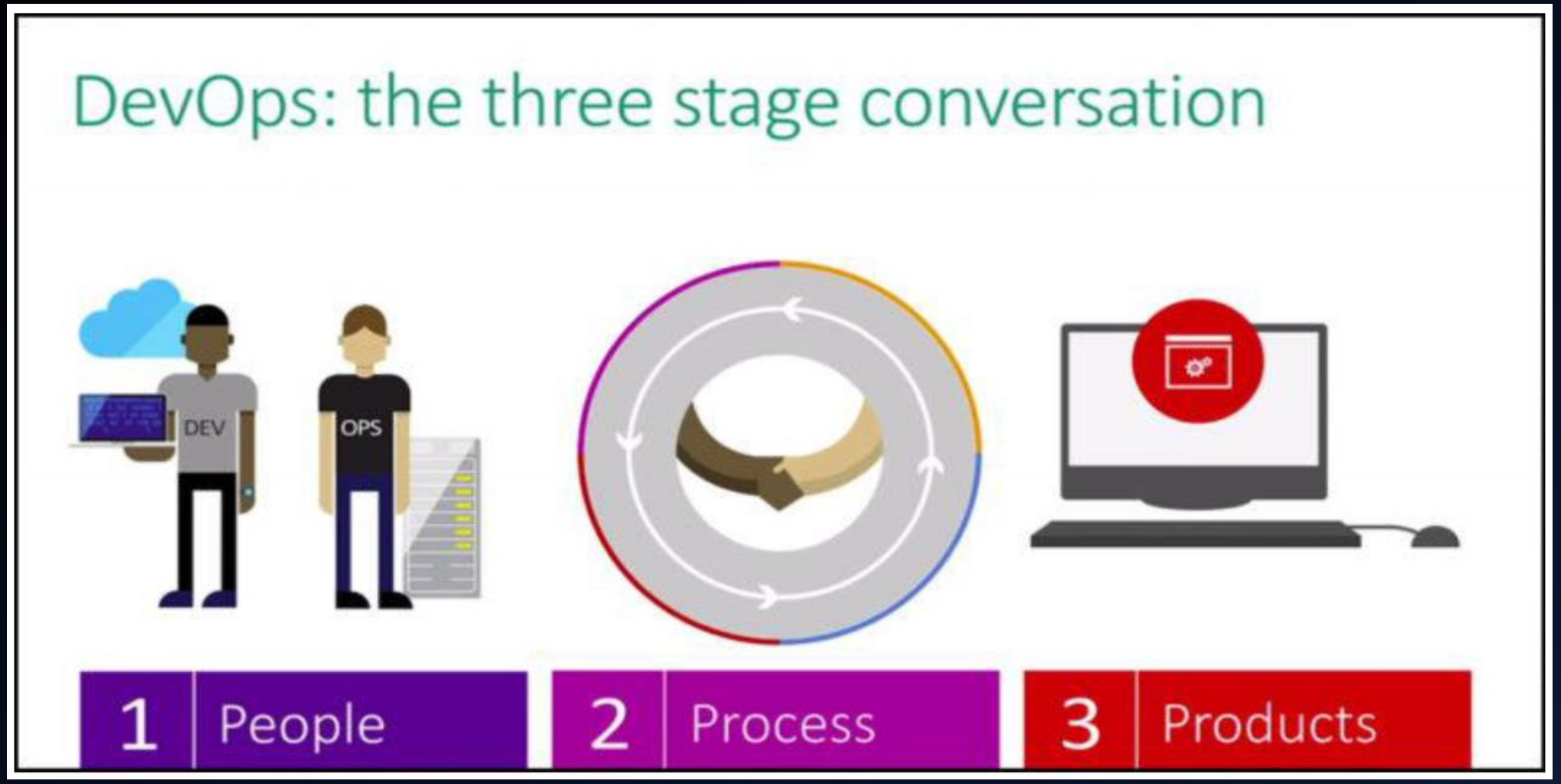
What is DevOps?



Bridge the gap between dev and ops



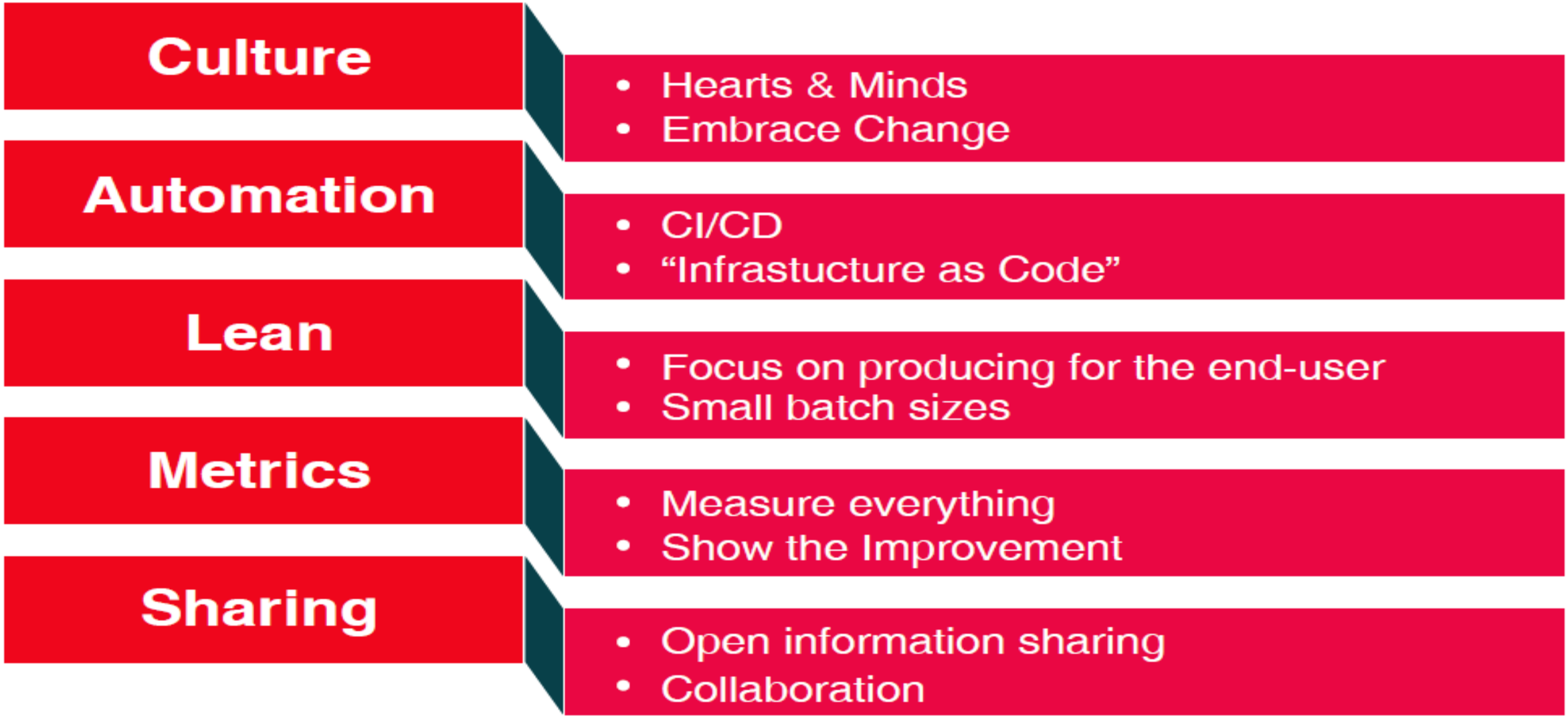
Correlation between PEOPLE, PROCESS AND PRODUCT



Survey of DevOps Quantifiable Benefits

Metric	Percent Improvement
Increased collaboration between departments	23%
Improved quality of our deployed applications	22%
Increased numbers of customers using our software/services	22%
New software/services that would otherwise not be possible/explored	21%
Fewer employees working on developing and deploying our software/services	21%
Reduced time-to-time market for our software/service	20%
An increase in revenue	19%
Our software/service made available across more platforms	19%
a reduction in spend on developement and operations	18%
Increased frequency of deploymwnts of our software/services	17%

DevOps is a way of thinking.



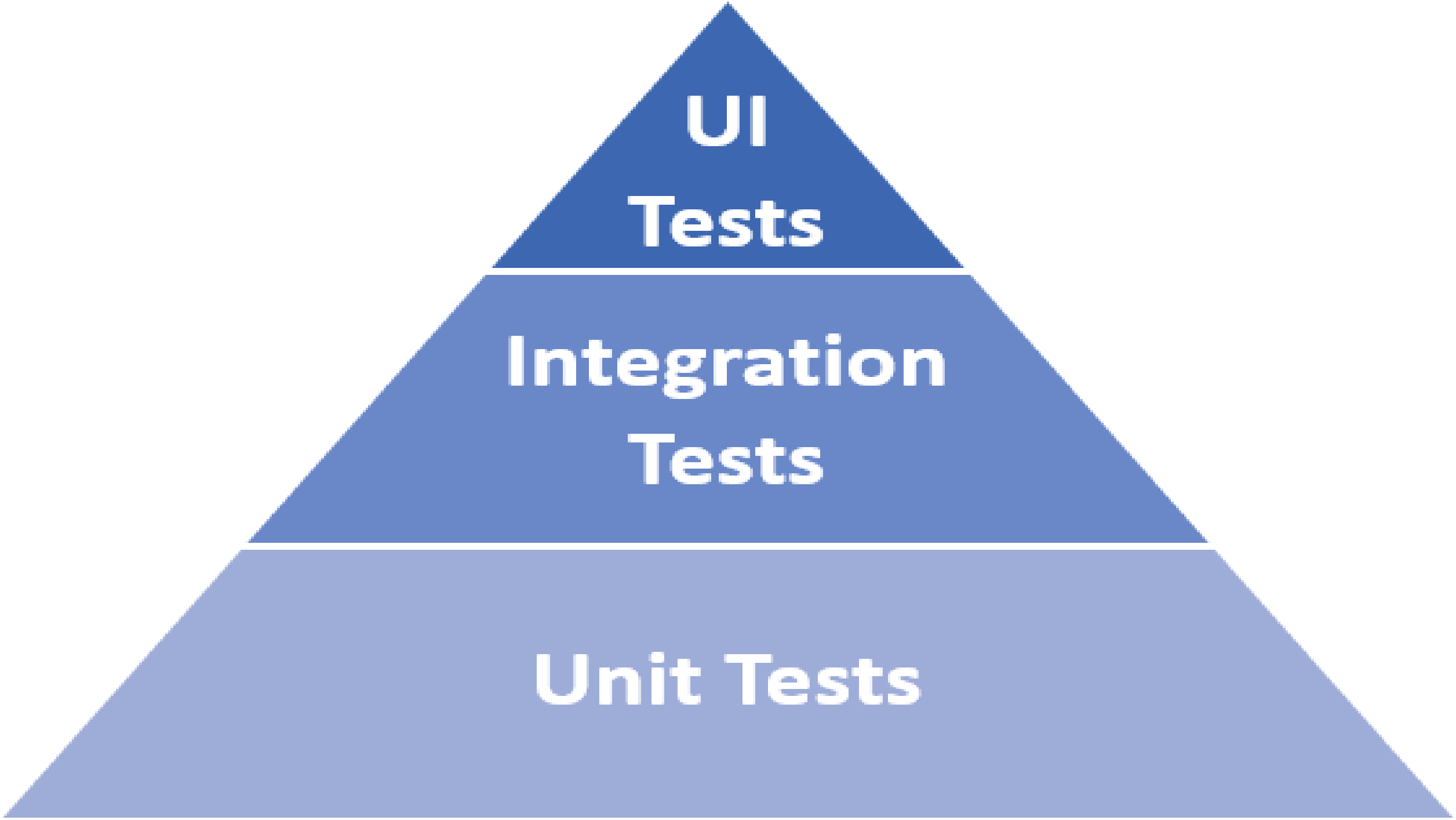
CALMS Model

Culture

- A DevOps culture changes the way businesses work, enabling teams to produce great results that make users happy.
- A big part of the culture is breaking down the silos that exist within an organization, so that Dev and Ops can work better together towards the same goal – those happy users.
- There's no longer an *It worked on my machine* or *It's not my problem* atmosphere. Now, it's *Let's work together to make this the best experience for our users*.

Automation

- Automation is key when it comes to DevOps.
- If you want to release more frequently, the release pipeline is a perfect candidate for automation.
- It’s completely repetitive and automation will help eliminate any manual errors.
- Continuous Integration with a thorough set of automated tests is important for automating releases.

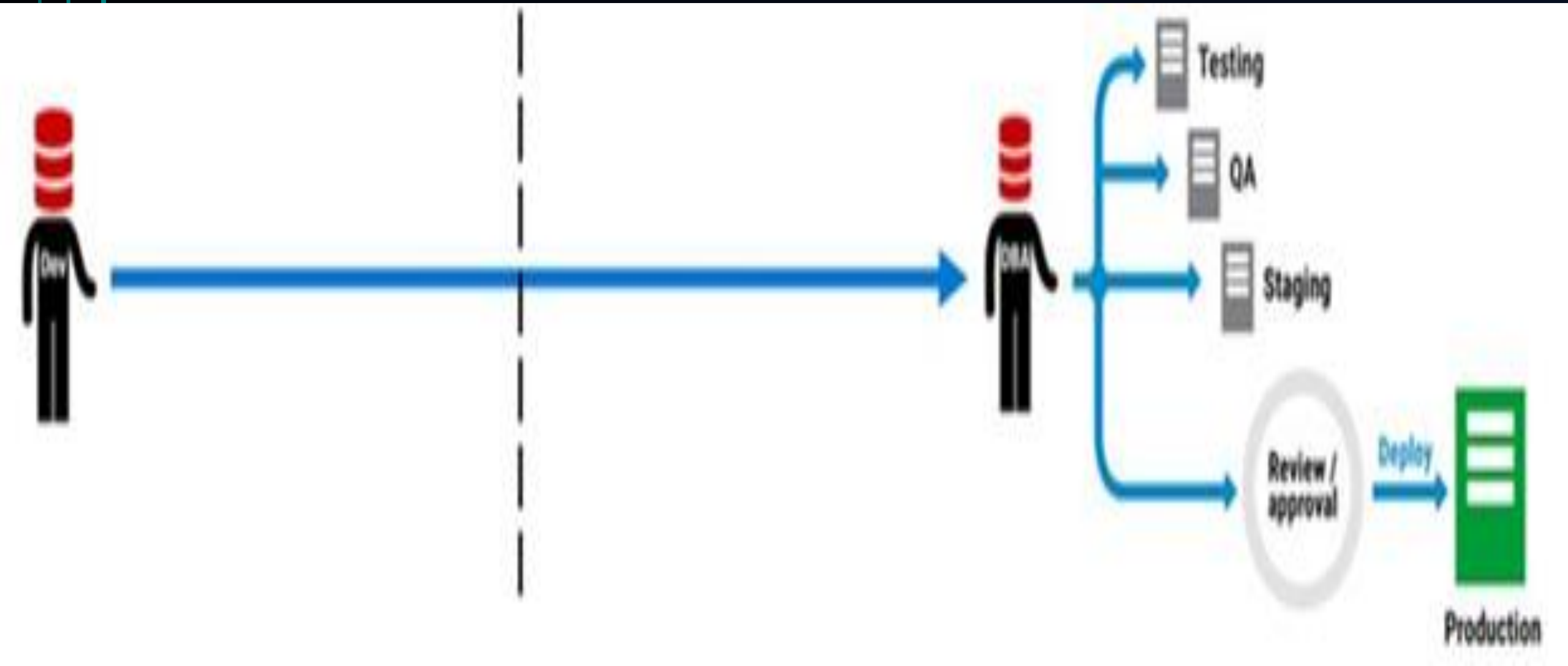

UI
Tests
Integration
Tests
Unit Tests

Continuous Integration

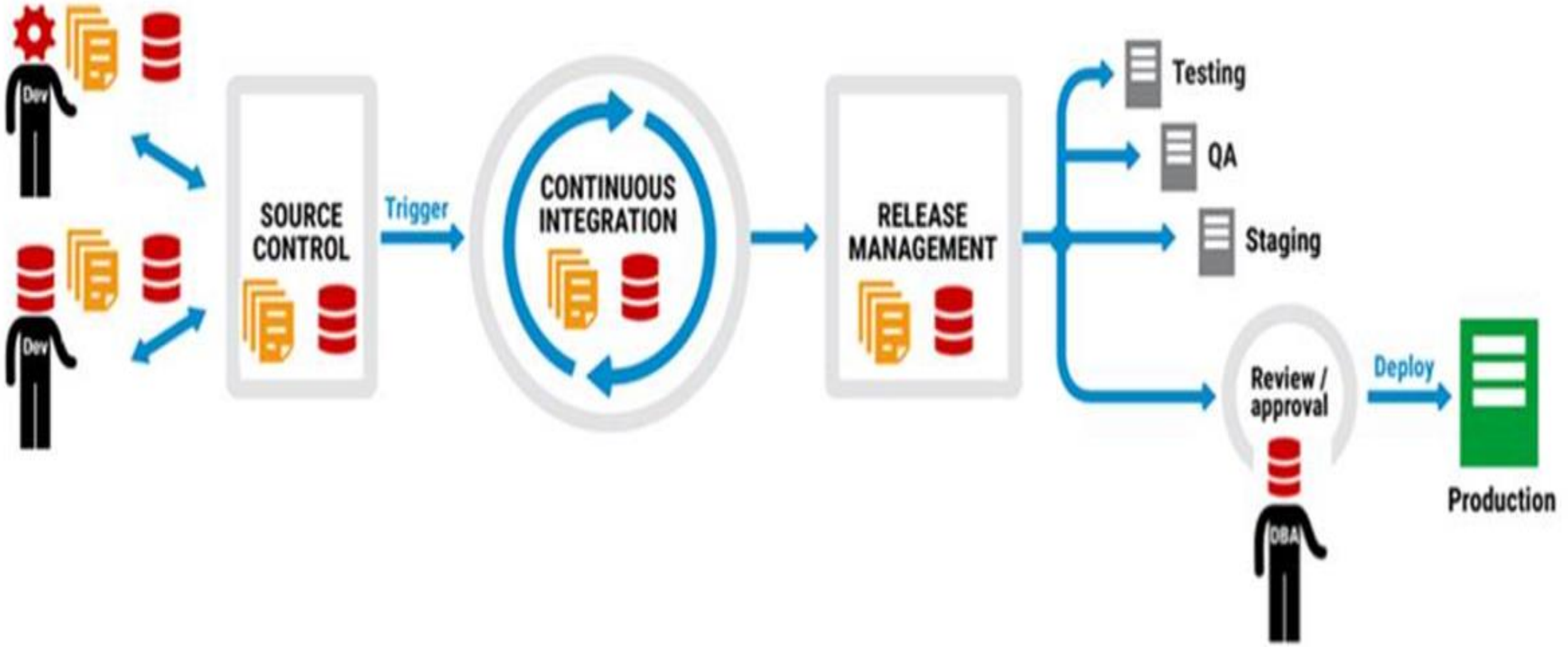
Continuous Delivery

Continuous Deployment

A typical deployment pipeline as a siloed process



A typical deployment pipeline as a continuous delivery process



Lean

- Lean is focused on incremental improvements and splitting the work into small batches.
- The small batches allow you to release frequently as you develop.
- This is important so that you can get real user feedback on your work and learn from it so that you can pivot and adjust according to what you learn as needed.

Measurement

- Measuring is important to get quick feedback and continuously improve, because you can only know if you're making a difference and improving if you have a baseline to measure against.
- The first area to start measuring is your internal processes,
  - like how long it takes for committed code to be running on production,
  - the frequency of releases,
  - the failure rate, and t
  - he mean time to recover from a failure.

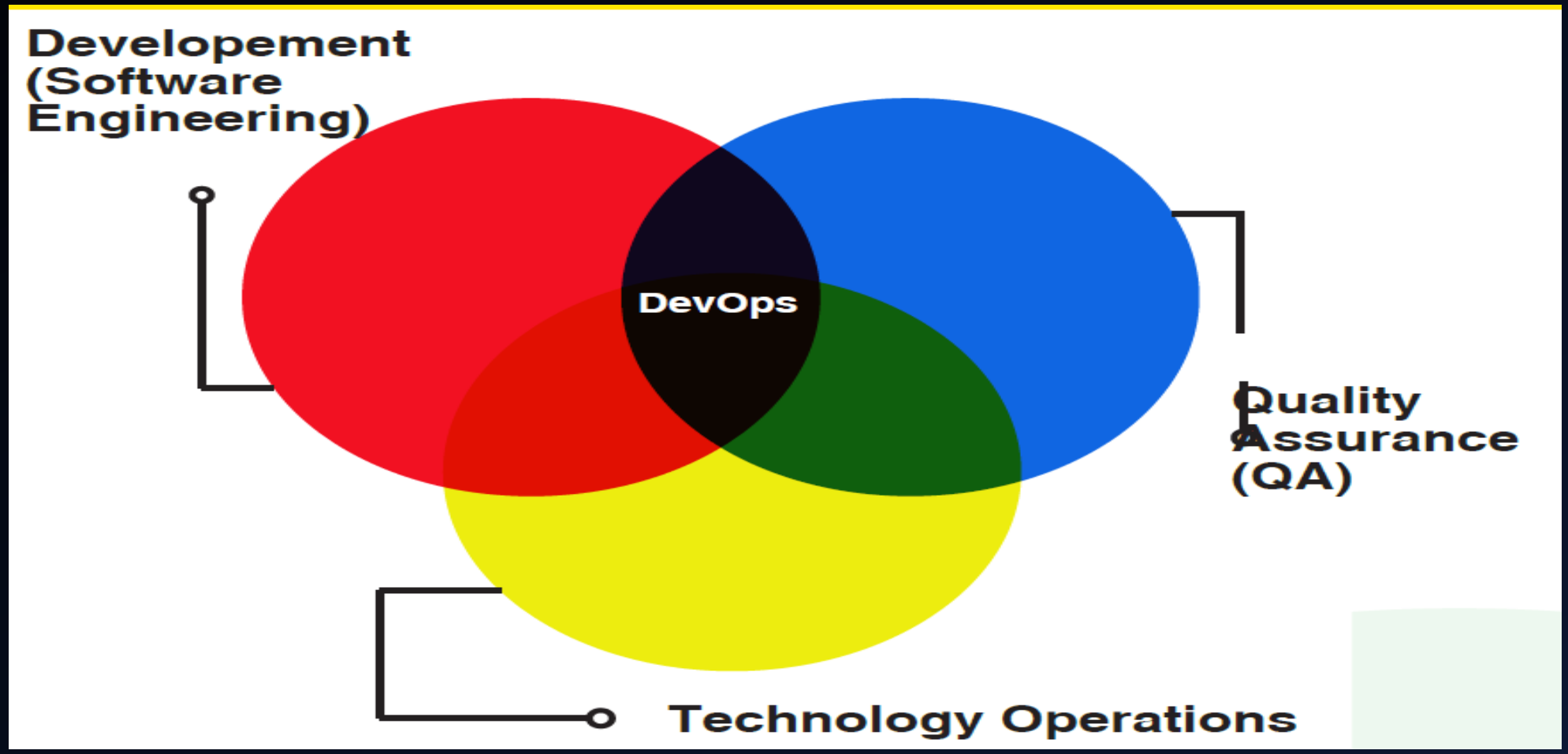
Sharing

- Sharing brings us back to the first value, Culture.
- If you want to make a huge impact on the culture, it’s important to share the lessons you’ve learned and your best practices with the rest of your company. This helps to break down those silos so that people work together to continuously improve and do what’s best for your users.

Conclusion

- CALMS are important DevOps values,
  - which enable you to break down silos,
  - work better together, and
  - provide frequent value to users that can be continuously improved.

Collaboration of dev ops and qa



DevOps combines the best of all teams

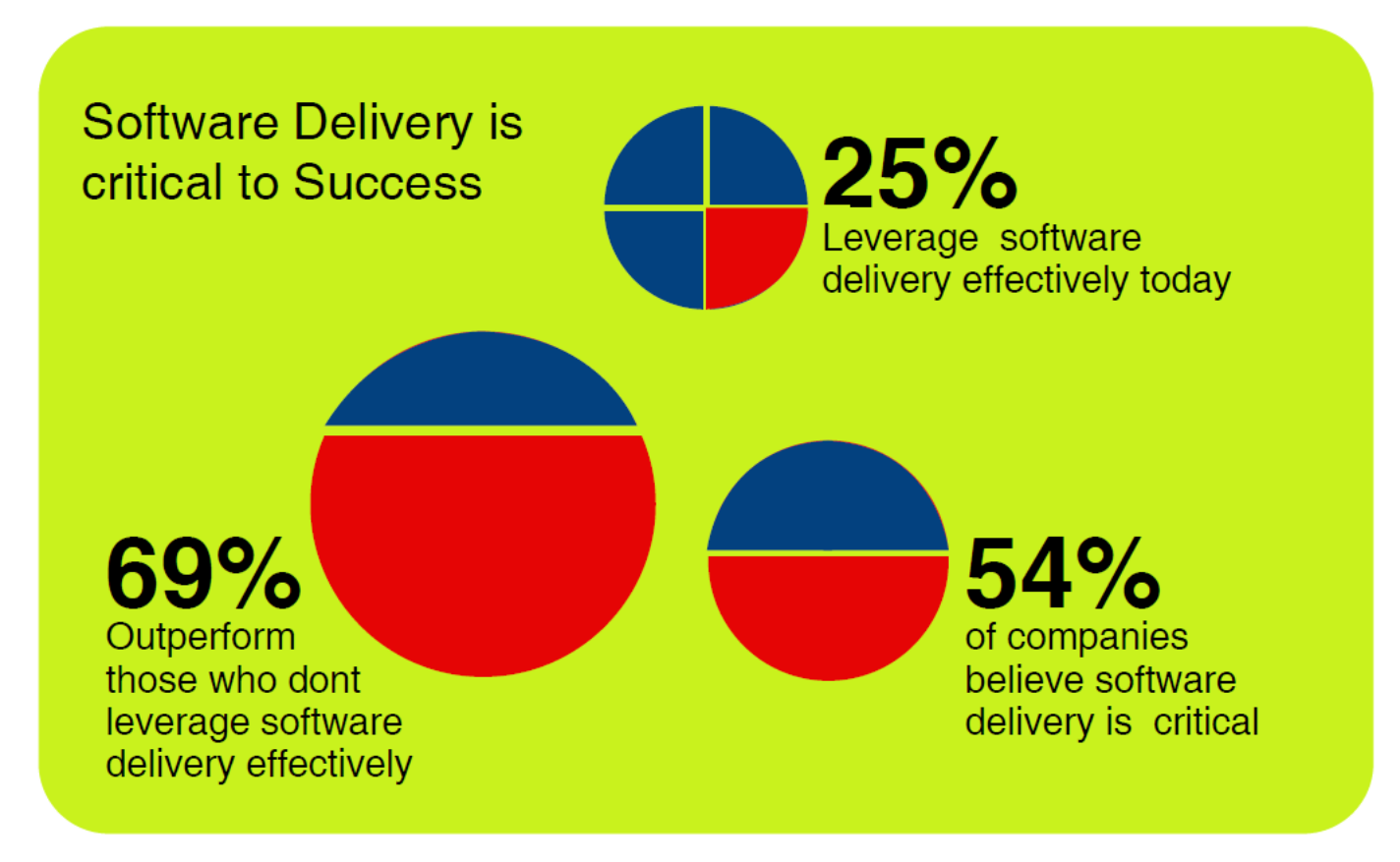
As following:

- Reduces cost/time to deliver
  - Deploy often, deploy faster with repeatable, reliable process
- Increases Quality
  - Automated testing, Reduce cost/time to test
- Reduces Defect cycle time
  - Increase the ability to reproduce and fix defects
- Increases Virtualized Environments utilization
- Reduces Deployment related downtime
- Minimizes rollbacks

5 Basic principles of DevOps

- Eliminate the blame game, Open post-mortems, Feedback, Rewarding failures
- Continuous Delivery, Monitoring, Configuration Management
- Business value for end user
- Performance Metrics, Logs, Business goals Metrics, People Integration Metrics, KPI
- Ideas, Plans, Goals, Metrics, Complications, Tools

Challenges and problems out there



Some of the serious issues blocking your software delivery are:

- No proper configuration management
  - Discrepancies in managing configurations
- Deployments are a blocker
  - Upgrade risk due to manual management of multiple application configuration and versions - Dependency on specific deployment engineer
- Production downtime
  - Due to lack of improper deployment instructions / checklist
- Hacking
  - Fixing directly in production (instead of a proper hotfix process) and forgets to check-in into source control

Some of the serious issues blocking your software delivery are:

- Building and maintaining servers
  - Time consuming and unproductive
- No environment management
  - Differences in development and production environments
- Slow deployments
  - Costly error prone manual process and efforts
- No shared ownership
  - Lack of feedback and proper metric leads

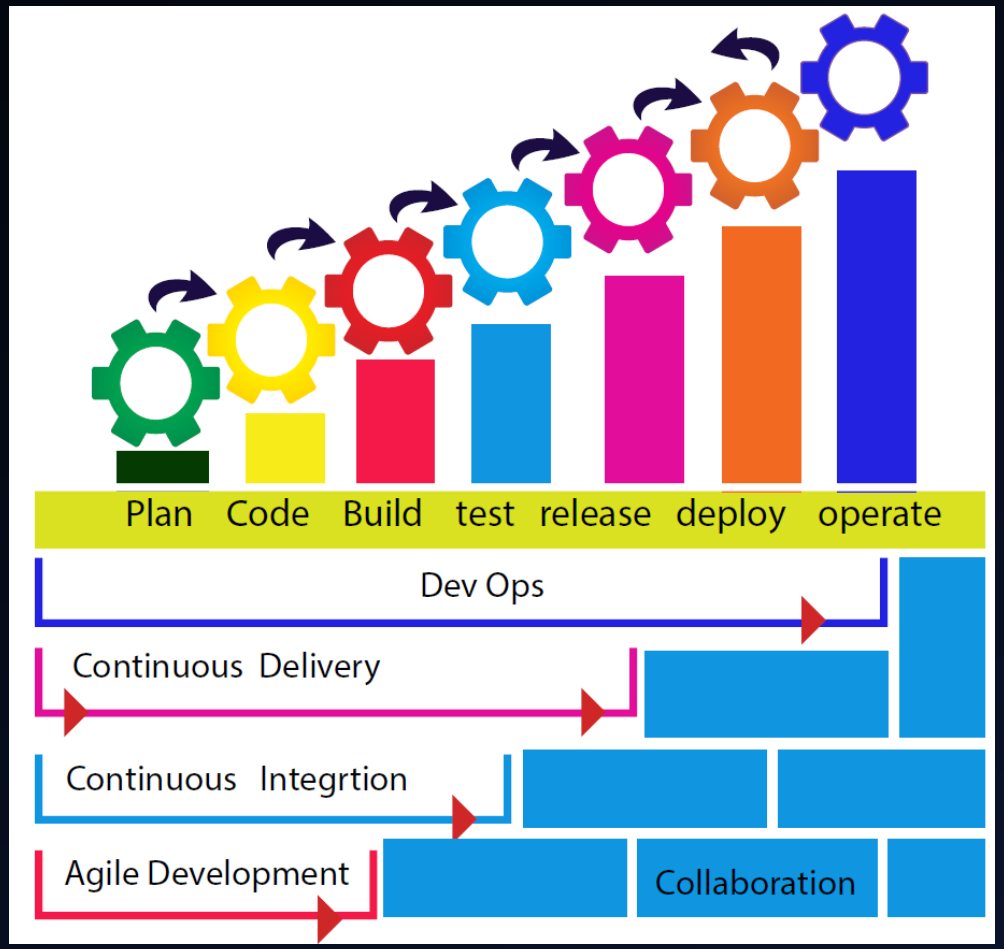
So,

- **Realize your entire business gets impacted if you do not have Continuous Delivery.**
- To enable that, you must adopt DevOps

7Cs of DevOps

- Communication
- Collaboration
- Controlled Process
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

Towards Automation



Without automation there is no DevOps.

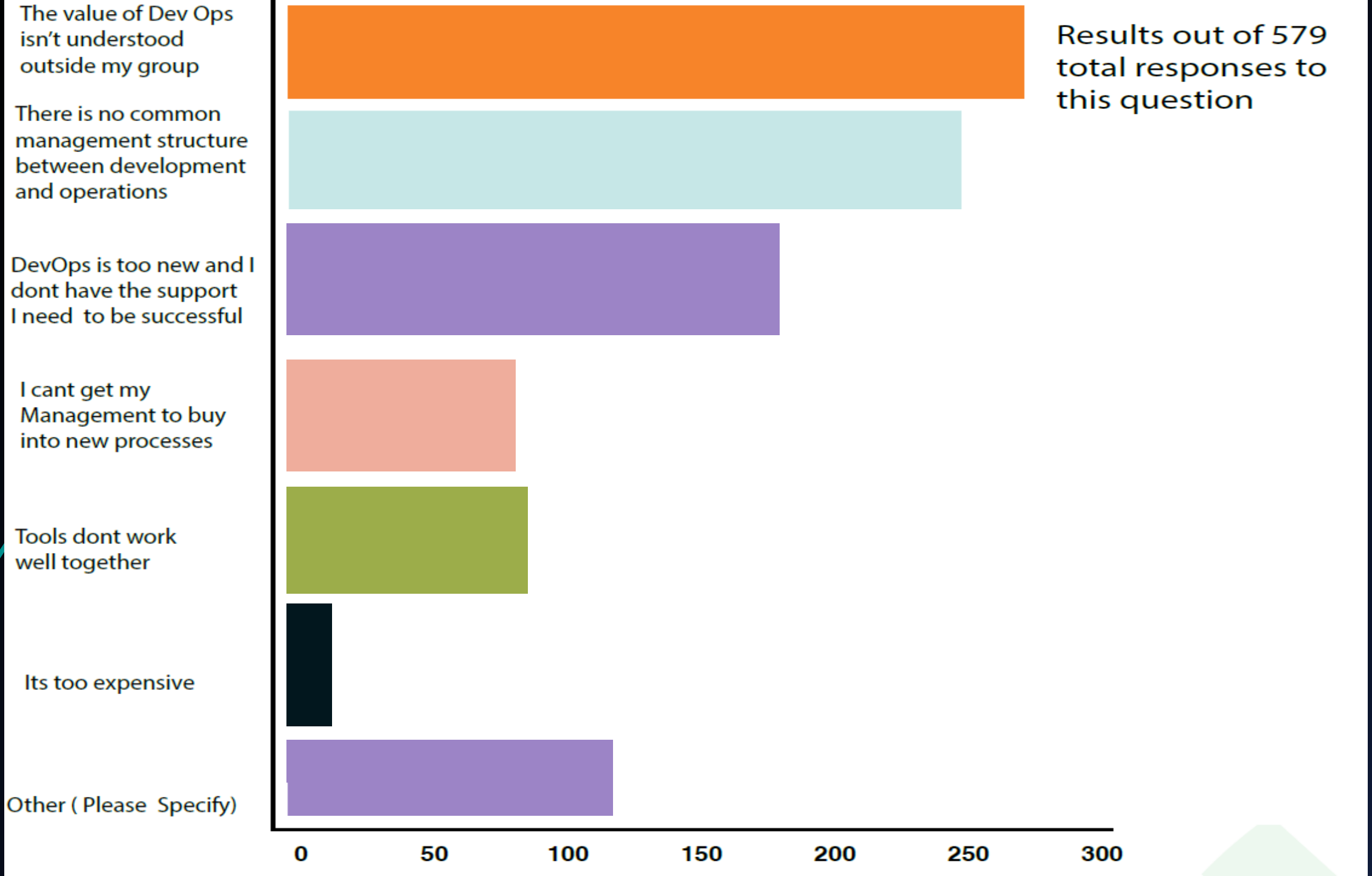
- Automate Provisioning
  - Infrastructure as Code
- Automate Builds
  - Continuous Integration
- Automate Deployments
  - Defined Deployment Pipeline and Continuous Deployments with appropriate configurations for the environments

Without automation there is no DevOps.

- Automate Testing
  - Continuous Testing, Automated tests after each deployment
- Automate Monitoring
  - Proper monitors in place sending alerts
- Automate Metrics
  - Performance Metrics, Logs

How will you benefit from DevOps approach

- DevOps Approach defines the **DevOps lifecycle as a Continuous on-going interactions and feedback** between the Customers-Business-Development-Operations.


The value of Dev Ops isn't understood outside my group
There is no common management structure between development and operations
DevOps is too new and I dont have the support I need to be successful
I cant get my Management to buy into new processes
Tools dont work well together
Its too expensive
Other (Please Specify)
Results out of 579 total responses to this question
0
50
100
150
200
250
300

Some Companies who got DevOps Right

Job Roles in DevOps

Job Roles

- Traditional IT Job roles
- DevOps Job roles

Traditional IT Job roles

Classic IT Job roles

- Architect
- System Administrator
- Application Developer
- Network Administrator
- Database Administrator
- Storage Administrator
- Security Administrator

DevOps Job Roles

Common Duties in the Cloud

Duties	Cloud Infrastructure Role	Application Role
Design/validate/expand solution-independent architectures and requirements	Cloud Enterprise Architect	Cloud Enterprise Architect
Design/validate/expand solution-dependent architectures and requirements	Cloud Infrastructure Architect	Cloud Application Architect
Build the infrastructure/application	Cloud Operations Engineer	Application Developer
Specify security requirements	Cloud Security Architect	Cloud Security Architect
Manage, monitor, and enforce security	Security Operations	Security Operations
Build and manage fast and scalable workflows	DevOps Engineer	DevOps Engineer
Design and build automation solutions	DevOps Engineer	DevOps Engineer
Perform cost coding and cost optimization	Financial Manager	Financial Manager
Manage operational teams	Program Manager	Program Manager

Questions

Queries

Feedback


Thank You