DrupalCon

NASHVILLE **2018**

APRIL 9-13

DrupalCon
NASHVILLE 2018
APRIL 9-13

What mix of DevOps “things” is right for your needs?

Aimee Degnan

Aimee Degnan

CEO / Principal Architect
Hook 42

cuteaimeeh

aimeerae

hook42inc

Topics

What & why?

How we got to DevOps.

How?

Concepts, Keywords, Tools

Choosing DevOps.

Assess you, your product, and your implementation.

Disclaimers & Goals

- This is a broad topic. This is not a deep dive.
- Topics warrant sessions or days of training.
- If you are expecting a tool and I haven't got there, we will get there.
- There are a lot of words. There will be a PDF version for reference.

Goals:

Gain Understanding -- Consider Impact -- Identify Relationships

Think Differently

Who am I? Why am I talking about this?

My Experience / Evolution

- Child of a Network & Systems Specialist
- Systems Administrator (multiple platforms + custom hardware)
- CMS Specialist / Web Application Developer (multiple platforms)
- Enterprise Architect (Online Software Applications)
- Project Manager (analyze and manage all the things)
- Process Improvement Engineer (efficiency!)
- Strategic Direction

Motivation:

Be in the most effective position to influence informed change.

Removing “word baggage”.

All the “things”!

Terms used in DevOps may mean different things to different people.

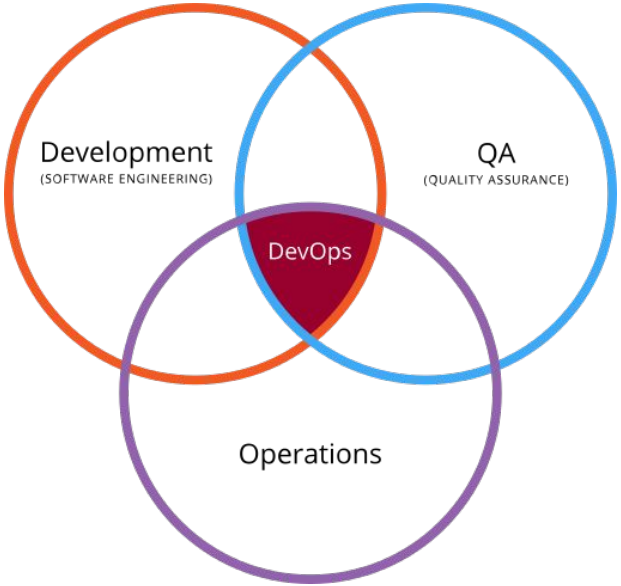
Let’s clarify DevOps terminology in context of **concepts.**


“Cloudy with a chance of DevOps.”
Bridget Kromhout
(Pivotal)

What is DevOps? Why did it evolve?

Software **DEV**elopment + IT / Software **OP**eration**S** (name)

But it also contains QA for business alignment.

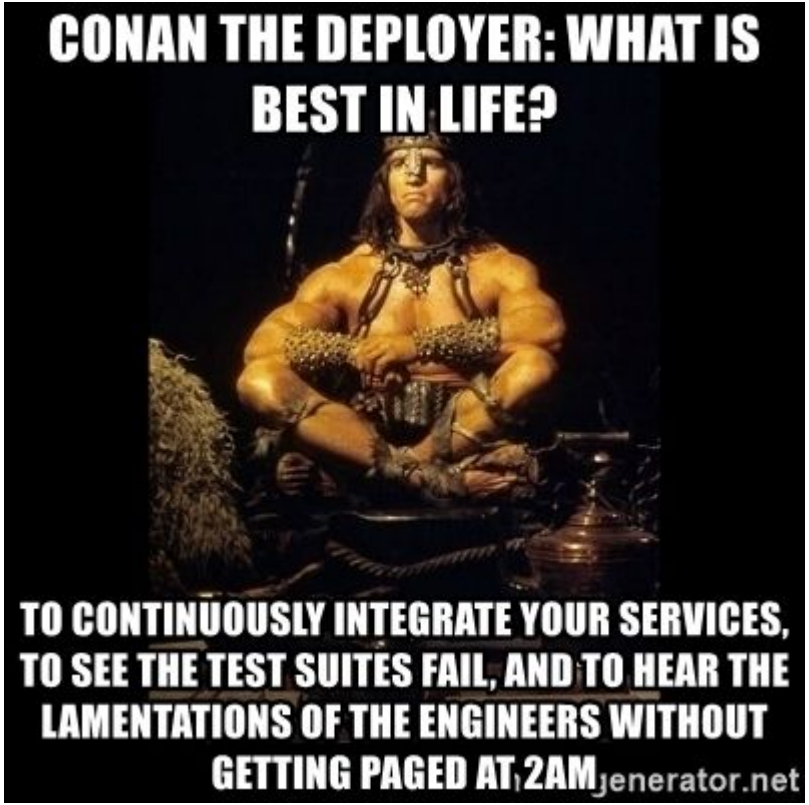


What is DevOps? Why did it evolve?

Systems Admins:
Why don't the engineers write code that will not continually break production at 3 am?

Developers:
Why can't I have root to make all the things just work?

What is DevOps? Systems Admin Goal



What is DevOps? Business goals.

An efficient and strategic use of your available resources (time, money, team, tech) to achieve:

- Faster time to market.
- Improved / increased deployment frequency.
- Shortened lead time between deployments.
- Lower failure rate on releases.
- Faster time to recovery.

Remember these.

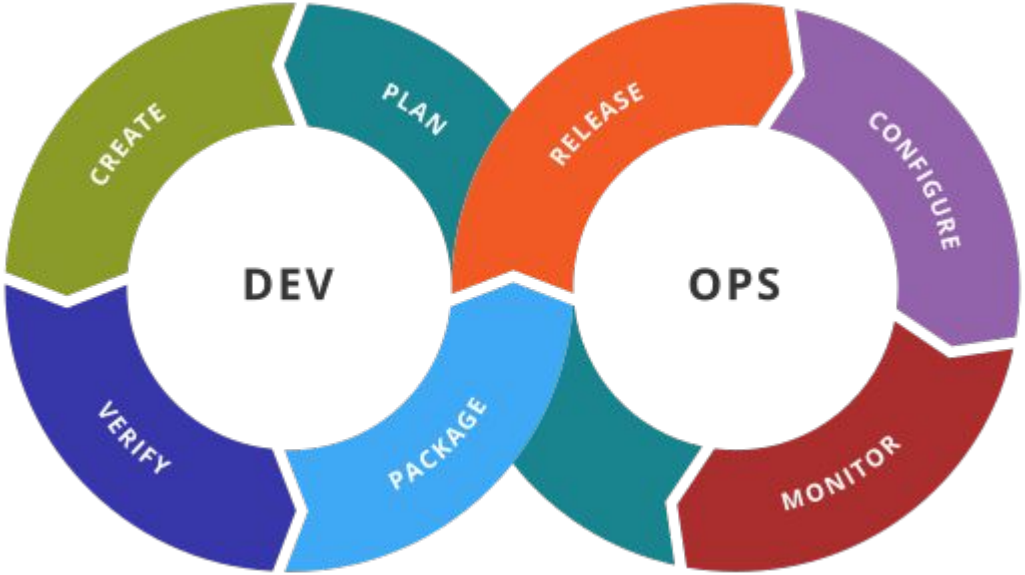
These are core goals that drive which DevOps “things” that you *choose* to use.

What is DevOps? Why did it evolve?

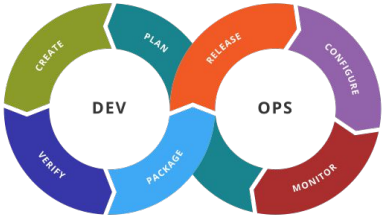
Cross-team collaboration is necessary to reach business goals.

- Applications grew more complex.
- Applications needed more computing power.
- Support bigger, complex environments.
- Movement from build & ship software (media) to continuous online software (web).
- Support Internet of Things. (hybrid hardware + online)
- ***Support more with same amount of people (or less).***

What is DevOps? The Process.



What is DevOps? The Process.



Process Mgmt 101:

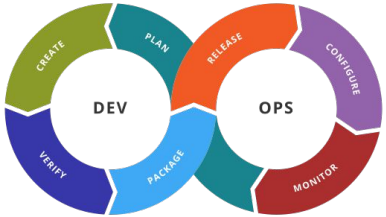
The people and systems performing actions in a repeatable order with predictable outcomes.

“Things” are transferred between steps.

Metrics are collected throughout the chained actions for review and adjustment within the Planning phase.

Basic process structure drives characteristics of DevOps.

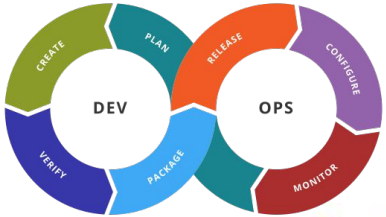
What is DevOps? Characteristics.



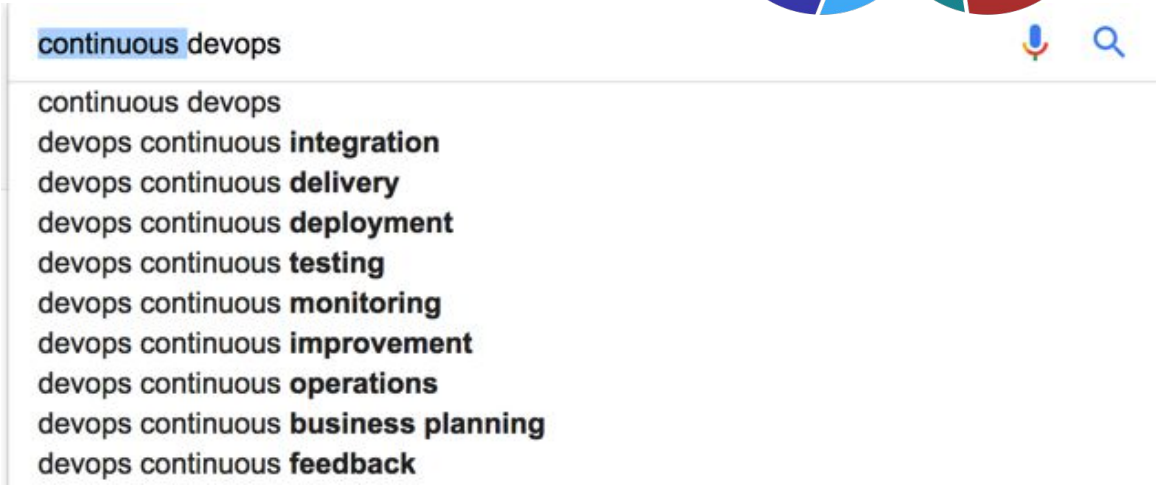
- Connected
  - Cross-team
  - Cross-role
  - Cross-system
- Parity / equality
- Velocity / Movement

- Continuous
- Repeatable
- Automated
- Early error detection

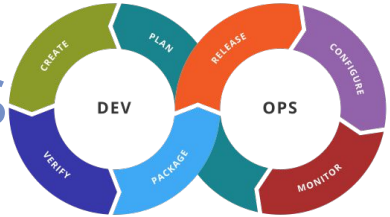
What is DevOps? Continous-ness.

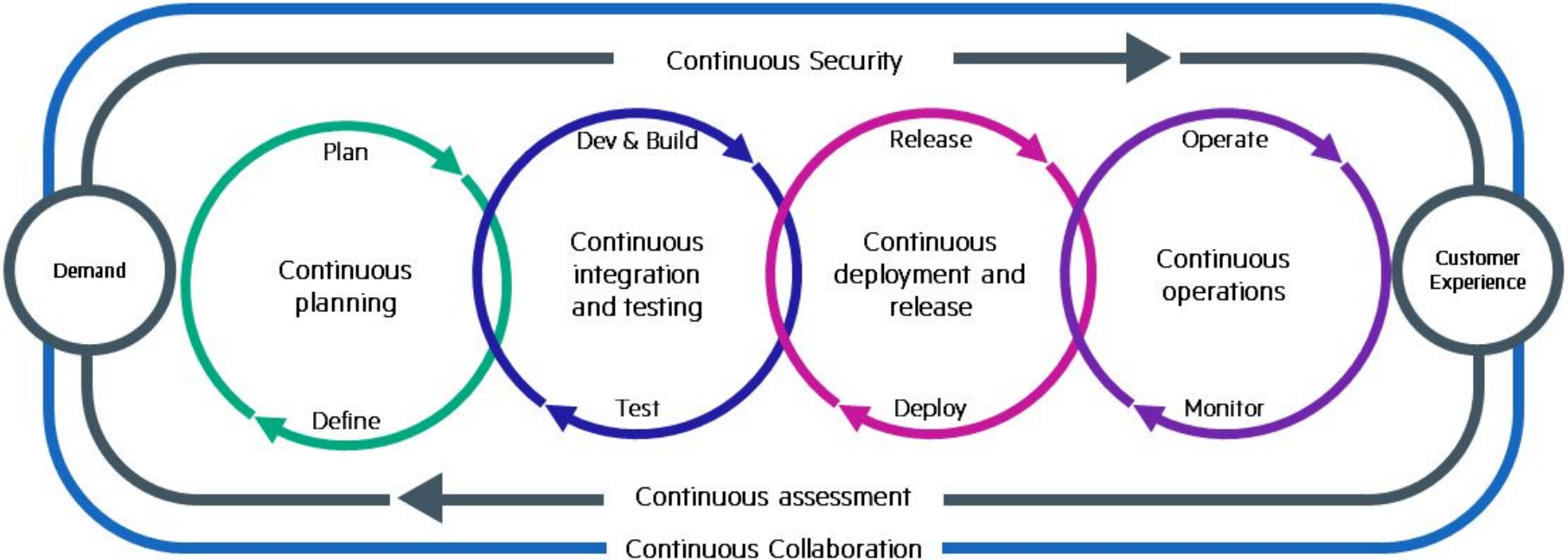


- Development
- Integration
- Quality / Testing
- Delivery
- Deployment
- Response / Monitoring
- Learning
- Operations
- …

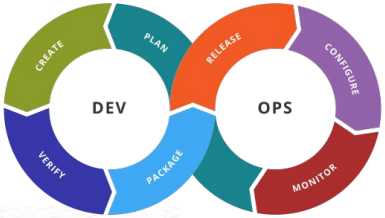


What is DevOps? Continuous-ness

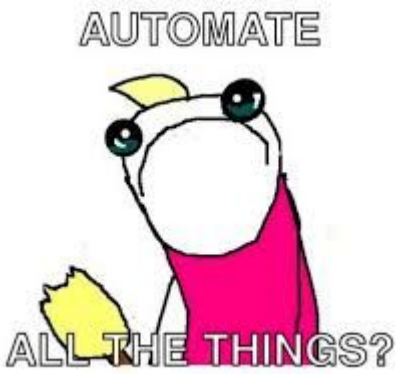


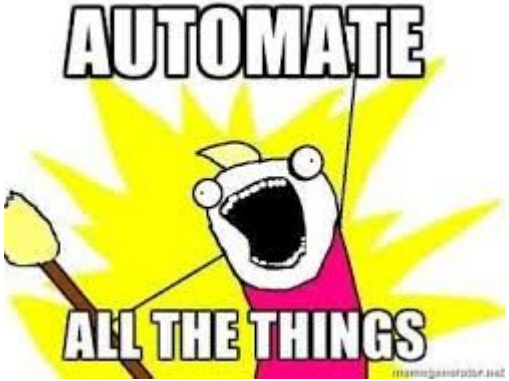


What is DevOps? Automation.



- Efficiency.
- Consistency.
- Reduce error.
- Force multiplier.
- Reduce time to marke
- Capture metrics.





Challenges of DevOps

What do I choose? It is so confusing.

Quickly changing and vast tool market.

All of the words. What do they mean?

What is the hippest thing and is it right for me?

What is DevOps? NoOps movement.

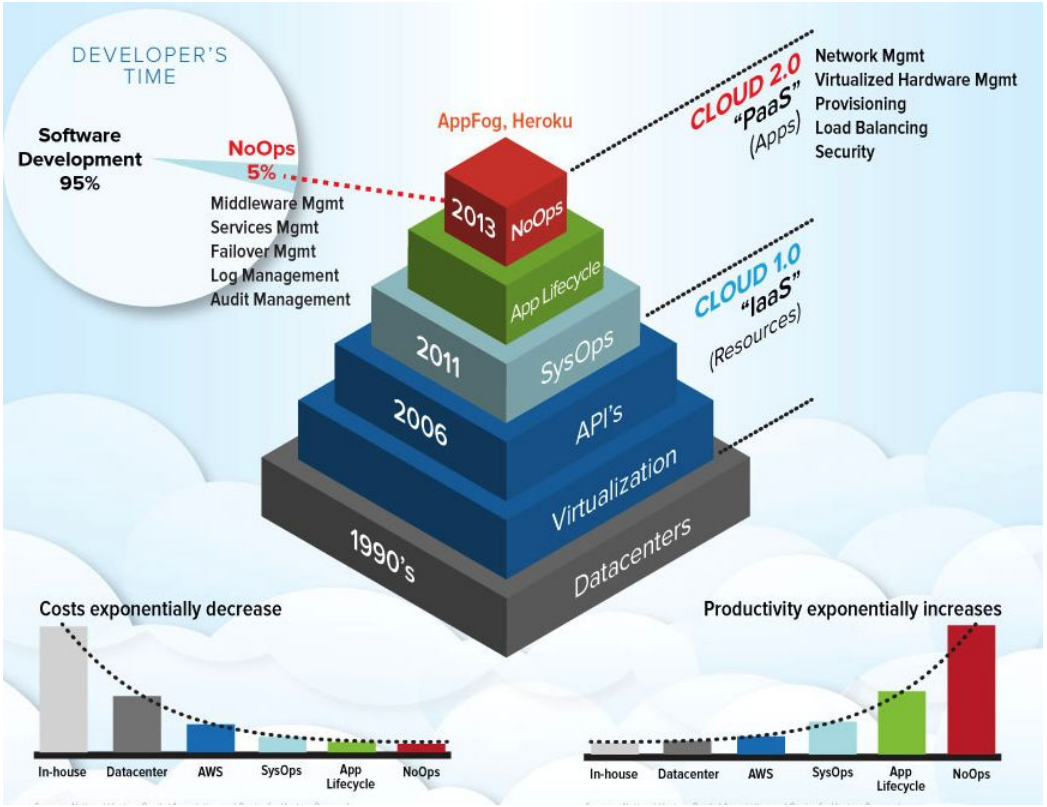
It is an evolution of Automation to Intelligent Operations

Philosophy:

IT environment is abstracted from the infrastructure that you do not need a dedicated team to manage software in-house.

- Serverless Programming (removes provisioning / config management)
- Containerisation (extended use)
- Microservices Architecture (challenges current CI/CD approaches)
- Intelligent and Unified Operations (across all tools - monitoring/deployment/logs/incident)
- Self Healing / Auto Remediation (auto-rollback, failover support, built in recovery)

What is DevOps? NoOps movement.



Where can I find some NoOps?

The market is maturing quickly. Visit some hosting vendors here!

- NoOps vision is driving growth in classic DevOps environments.
- You will see improvements across the process phases.
- Will the IT / Operations people lose jobs?
  - A bunch of IT / Ops people help create NoOps infrastructures.
  - A shift of need from the customer environment to the providers.
  - Responsibilities will change to approach the topics.


DevOps Culture
Don’t just DO DevOps,
LIVE DevOps.

The People.

Developers

Build forward changing features for the business.

May not understand network computing.

Use tools created by DevOps and IT engineers.

IT / Ops

Intensely cautious.

Paranoid that production will break. It will. And they will get a call.

Understand full impact of systems changes.

DevOps

Represent crossover of the two roles.

Create tools that help improve confidence across teams.

Balance stability with velocity of forward change.

What is DevOps? Skills Assumptions.

All technical staff supporting DevOps processes need to know more.

Some of the technical staff may not have skills for DevOps tasks.

Some may not have the desire to be a DevOps engineer.

All technical staff

should be able to use the tools

created by DevOps engineer.

What is DevOps? The Skills Continuum.

Developer Evolution:

1. How to Code
2. Dev tools (within context)
3. Full stack application
4. Integrations
5. Systems / Networking / Security
6. DevOps

SysAdmin Evolution:

1. Understand operating systems
2. Networking / Security
3. Application layer interactions
4. Full Stack Debugging
5. Application Development
6. DevOps

Cross-team Alignment Matters

Measuring & Rewarding Success:

- Incentives may be different.
- Success measures may be different.
- Accountability may be different.

Measures:

- **Development:** How many features can I get out the door?
- **Operations:** How stable is the overall system? Uptime metrics.

Problem:

A lot of unstable / buggy features is great for numbers (“good” for dev), but bad impact on the runtime environment (bad for “ops”).

Impact to culture:

“Wall of Confusion”. Finger pointing. No one wins.

Cross-team Alignment Matters

Bridging the gap:

- There are no silos.
- There is only the total team.
- Full transparency of information.
- Open and frequent communication is rewarded.
- Each team’s performance metrics need to be tied to others.
- Data-driven accountability of individual, team, and overall process.
- Cross-team collaboration success measured on overall performance.

Business Support

DevOps will fail without leadership.

Company must encourage communication.

Embrace an agile / nimble approach.

Invest time, money, team, and faith.

Use data-driven decisions to guide growth.

DevOps is work and must be addressed as such.

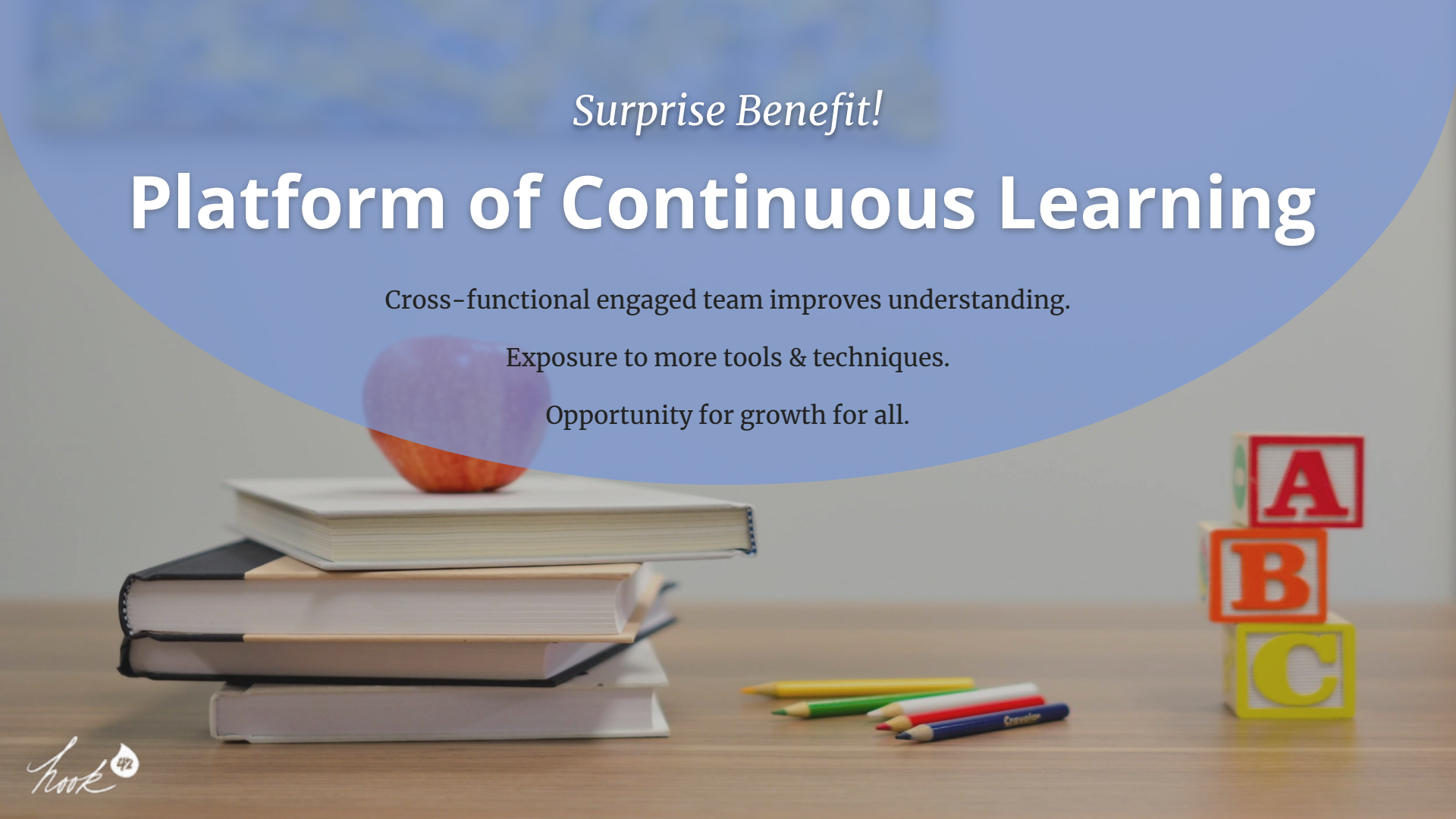
Surprise Benefit!

Platform of Continuous Learning

Cross-functional engaged team improves understanding.

Exposure to more tools & techniques.

Opportunity for growth for all.




The DevOps
Tool Chain

Tool Chain: Overarching concepts

- Self-hosting vs. Hosted Solutions
- Single-function tools vs. Solution Suites
- Activities within phases of the DevOps process
- Documentation matters

Self-hosting vs. Hosted solutions

Self-hosting is the installation, configuration, and upkeep of a tool on a server completely managed IN and BY your organization. This is in addition to the DevOps work required to create a DevOps process.

Samples: Self hosted Atlassian Stack, Redmine, Installed Git servers, Jenkins, GitLab

Hosted solutions are a pay for services model where the toolkit and infrastructure is supported by the hosted company. You just focus on your Business Needs.

Samples: GitHub, Pantheon, Acquia, Jira/Confluence (hosted), Circle CI, Travis CI, GitLab

Single-function tools vs. Solution Suites

Single function tools serve one focused part of the DevOps process, but may be leveraged across multiple phases of the DevOps cycle. Testing Software.

Solution suites that provide many DevOps services ready for integrated process management.

Cloud based platforms provide solution suites that provide many DevOps services preconfigured for their platform tools.

The glue: Most platforms and tools have APIs to integrate into dashboards and deployment tools using APIs.

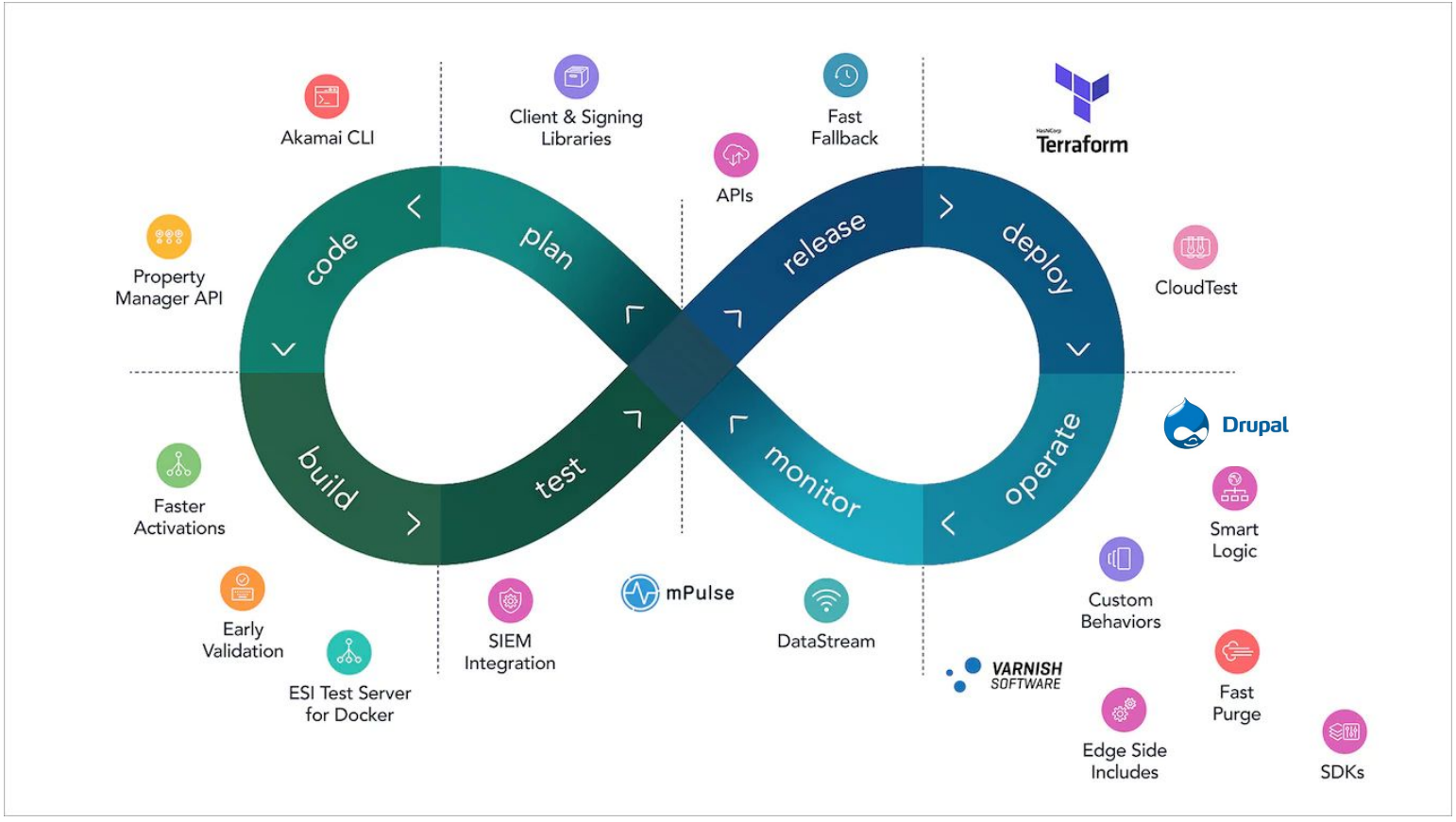
Documentation Matters

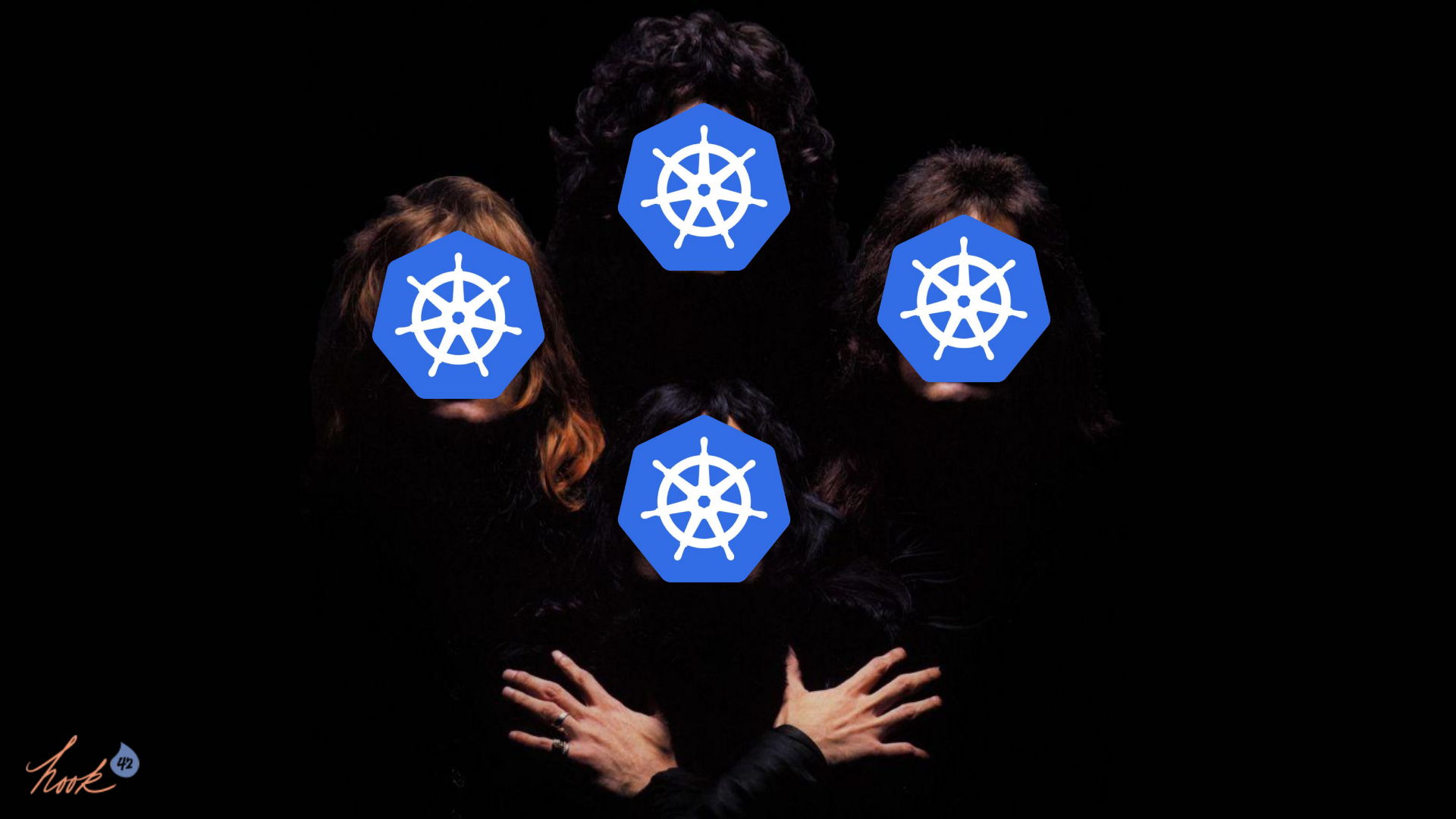
But it doesn't solve all problems....

Key documentation:

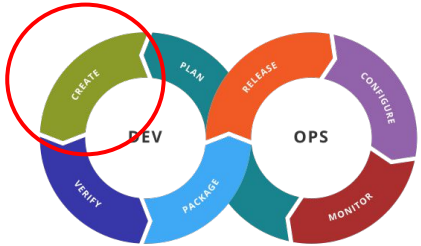
- Usage of Development tools (within context)
- Process metrics captured (for each phase)
- Incident management, remediation, and escalation steps
- Diagrams: infrastructure, application, integration architecture
- Communication expectations

Tool Chains in the process. The Activities.





Tool Chain: Create / Code



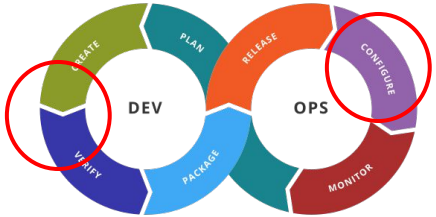
Activities:

- Code development and review
- Source code management tools
- Code merging
- Primarily on the developers environment.

Tools:

- **Repositories:** Git, Perforce, Team Foundation. GitHub, GitLab, BitBucket
- **Development:** IDEs, code debuggers, code quality (Lint, PHP CS).
- **Compiling Tools:** In IDEs, Front End Dev / SASS suites (Node/Gulp).
- **Environments:** Virtual Machines / sandbox environments.

Tool Chain: Build (for Test)



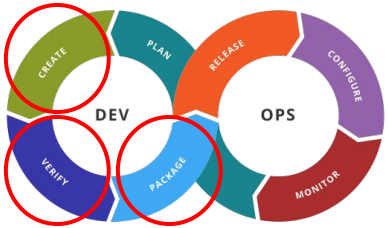
Activities:

- Continuous integration
- Build status

Tools:

- **Container images:** Docker, Drupal VM, Vagrant
- **Provisioning system / container orchestration:** Terraform, Kubernetes
- **Continuous integration / deployment:** Jenkins, GitLab (CI/CD), Travis CI, CircleCI, AWS Code Pipeline

Tool Chain: Verify - Test



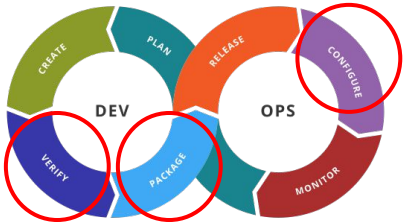
Activities:

- Continuous testing tools provide feedback on business risks.
- Test coverage - the amount of tests written within your business application.

Types:

- **Code level (unit test)** Behat+Cucumber, PHPUnit, Code Quality
- **Functional (procedural)** Selenium, Puppeteer / Mocha, Saucelabs
- **Browser / platform compatibility** (Browserstack, Saucelabs)
- **Load (infrastructure)** (LoadRunner, Apache Bench, JMeter)

Tool Chain: Package (Build for production)



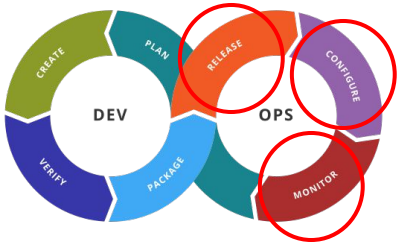
Activities:

- Artifact repository.
- Application pre-deployment staging.

Tools:

- **Container images:** Docker, Drupal VM, Vagrant
- **Provisioning system / container orchestration:** Terraform, Kubernetes
- **Continuous integration / deployment:** Jenkins, GitLab (CI/CD), Travis CI, CircleCI, AWS Code Pipeline

Tool Chain: Release ("Pressing the button.")



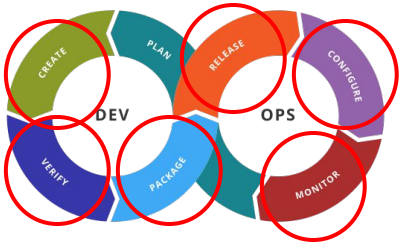
Activities:

- Change management / release approvals.
- Release automation.
- Continuous delivery.

Tools:

- **Productivity apps:** ticket tracking system (Jira)
- **Container images:** Docker, Drupal VM, Vagrant
- **Provisioning system / container orchestration:** Terraform, Kubernetes
- **Continuous integration / deployment:** Jenkins, GitLab (CI/CD), Travis CI, CircleCI, AWS Code Pipeline

Tool Chain: Configure (All the things.)



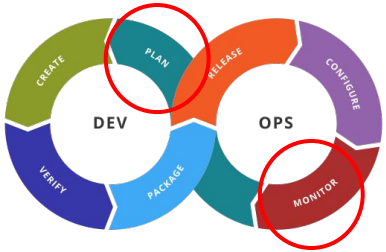
Activities:

- Infrastructure configuration and management
- Infrastructure as Code tools
- Multiple layers of config: system, application, environment
- *Keywords:* variables, consistency, parity across systems, don't mess up.

Tools:

- **Development productivity:** Gulp / FED SaaS / Linters
- **Configuration management:** CFEngine → Chef → Puppet
- **System images:** Docker, Drupal VM, Vagrant
- **Infrastructure as Code:** Container management, Terraform (internally hosted), AWS (all aspects), Google (Kubernetes (helm)), OpenStack, GitLab

Tool Chain: Monitor



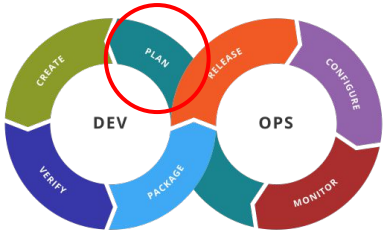
Activities:

- Applications performance monitoring, End–user experience
- **Inside-out monitoring:** Applications performance, on server, runtime, logs
- **Outside-in monitoring:** End-user experience / monitoring as a customer
- **Keywords:** telemetry, TSDB (time series), historical vs real time data

Tools:

- **In-app monitoring:** Drupal, custom events, error management
- **Server/system/application/event level:** DataDog, New Relic, Prometheus
- **Log Aggregators:** LogStash, SumoLogic, Splunk
- **Dashboards:** Grafana (raintaink), SNAP (intel)
- **External monitoring:** Pingdom, GitLab

Tool Chain: Plan



Activities:

- Retrospective and continuous improvement.
- Review metrics (all of them) and identify improvements.
- Prioritize tech debt, DevOps, and internal build requests.

Tools:

- **Productivity apps:** ticket tracking system (Jira), roadmaps, priorities
- Feedback from the team.
- Process monitoring (metrics of overall system)
- Feedback from product (increase in sales, leads, etc)
- **All the other data.**

Project Management & DevOps

Agnostic:

- DevOps work is work.
- Use tickets to track work.
- DevOps needs to be included.
- Integrated triggers / events.
- Retrospective learning.
- Evolution through planning.

Waterfall:

- Dependency order of work → Product Backlog management

Agile / Scrum / Kanban:

- DevOps leverages Agile approach
- Imbedded DevOps team member
- Parallel scrum teams / scrum of scrums


Choosing your DevOps
Keep it simple.

Choosing DevOps? Considerations.

Getting Started:

- Time
- Money
- Business Support

Understanding:

- Product Needs
- Legal Constraints
- Infrastructure Constraints
- Available DevOps tools
- Investment to Reward (ROI)

Execution:

- Skills
- Team
- Tools
- Configurations
- Functional Process
- Metrics

No disruptions while disrupting!!

Know your Constraints

Why can't I use the cool tools?

Legal Compliance

Self-driving Privacy

Who are you?

Individual

Responsible for all parts of the DevOps to support personal work.

Type, experience, professional goals, and skills drive tool choices.

Business

Small: a single dev team. Business organization is “simpler”.

Large: Multiple dev teams supporting a single, collaborative product or multiple products.

Complex or Simple

Agency

Needs to support the individual and the business.

Keep costs down with high efficiency.

Serve their clients’ needs.

Choosing DevOps? Individual?

- Use containers.
- Saas Solutions (100%).
- Keep is simple.
- Keep it “Good Enough”.
- Reduce the number of disparate tools.

Choosing DevOps? Small Business?

- Small team / “big” site → SaaS 100%*
- Small team / small site → SaaS 100%
- Both:
  - Use containers.
  - Reduce the number of disparate tools, if possible.
  - Choose the tools that best fit your company’s needs.
  - Adhere to the set business objectives for the DevOps performance success.

Choosing DevOps? Larger Business?

- SaaS until it internal hosting + resource costs are cheaper.
- Internal hosting for data security or legal constraints.
- Internal hosting considerations
  - Still use cloud / provisioning tools, but you have to make your own cloud.
  - Limited toolset to those that can be self-hosted.

Choosing DevOps? Agency?

Know your business:

- May have multiple clients of various sizes.
- Support multiple technologies (PHP / Java).
- Staff's skill level and ability to support infrastructure tasks.
- Hybrid of self-hosted and SaaS solutions.

Know the client's needs:

- Stamp-em-out vs. complex client infrastructures.
- Legal requirements. Uptime expectations. Budget.


“How much does a DevOps cost?”
Adam Bergstein
(@nerdstein)
hook 42

BizOps: When to invest in DevOps?

You are probably doing some DevOps already.

- Productivity tools
- Git / Source Code Management
- Front-End development (SaaS tools)
- Hosting platforms
  - SCM, deployment, backups, monitoring
  - Pantheon, Acquia, Platform.sh

We're up all night to get lucky. -- Daft Punk

Work it, Make it, Do it, Make us Harder, Better, Faster, Stronger. -- Daft Punk

BizOps: Expected Investment Types

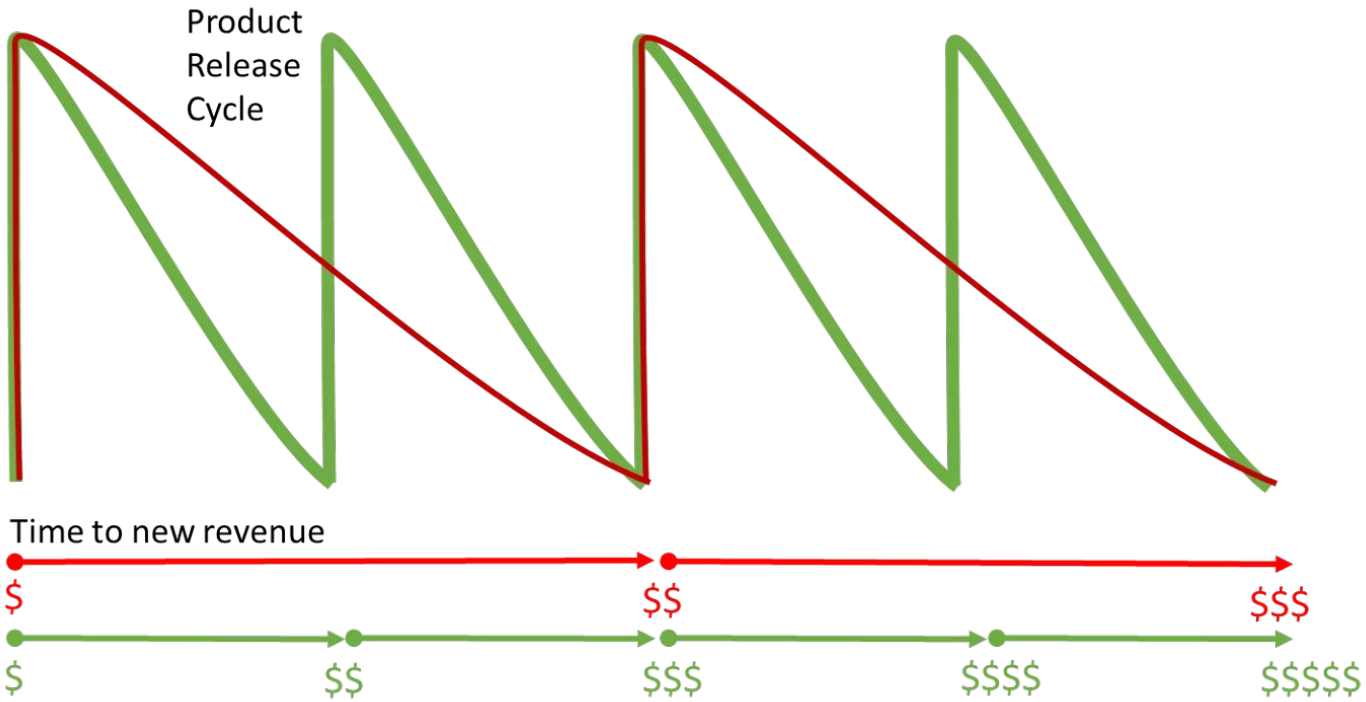
https://devops.com/the-roi-of-enterprise-devops/

BizOps: Costs and Investment

Benefit/Metric	Improvement
New software/services that would otherwise not be possible	21%
A reduction in time spent fixing and maintaining applications	21%
Increased collaboration between departments	21%
An increase in revenue	19%
Improved quality and performance of our deployed applications	19%
A reduction in spend on development, testing or operations	19%
Our software/services made available across more platforms	18%
Reduced time-to-market for our software/services	18%
Increased numbers of customers using our software/services	18%
Fewer employees working on developing and deploying our software	18%
Increased frequency of deployments of our software/services	15%

https://devops.com/the-roi-of-enterprise-devops/

BizOps: Costs and Investment



https://devops.com/the-roi-of-enterprise-devops/

Implementing DevOps tools

You can do it!!

Keep it simple.

Keep it focused.

Choose one tool at a time.*

Evolve it over time.

Align tools to company needs.

BizOps: Is my DevOps working?

Symptoms your DevOps “things” need work:

- Site going down. Infrastructure unstable.
- Too long between deployments. Slipped release schedules.
- Feature development not getting done.
- Fear of change vs. confidence in innovation.
- Internal battles / finger-pointing.
- “Works on my machine.”
- “I need root on production.”

Takeaways

Be realistic with your needs.

Understand concepts and terms.

Foster the humanity in DevOps.

Enjoy the shiny tools. :D

Special Thanks

- My dad.
- Hook 42 team. Adam Bergstein.
- DevOps peers: Brad Degnan, Marc Pernia.
- San Francisco Drupal Users Group.
- All of the sysadmins, developers, and PMs from Before.
- All of the people making great strides in the DevOps tool space.

DrupalCon
NASHVILLE 2018
APRIL 9-13

Join us for contribution sprints

Friday, April 13, 2018

Mentored Core sprint

9:00-12:00
Room: Stolz 2

First time sprinter workshop

9:00-12:00
Room: Stolz 2

General sprint

9:00-12:00
Room: Stolz 2

#drupalsprint

What did you think?

Locate this session at the DrupalCon Nashville website:

http://nashville2018.drupal.org/schedule

Take the Survey!

https://www.surveymonkey.com/r/DrupalConNashville