Desperately Seeking DevOps
Summit 2019

Presenters:

Matt Micene
Kevin Franklin

Notes:

- How to drive culture shift from conventional operations?
  - Create a devops team, leads to fighting
    - Fighting over tooling
  - Some luck with having an SME team that has buy in from other teams
  - How do we drive the culture change
    - How do we handle the shift in approach of operations
  - How to make operations into devs
    - Some accept some don’t
- Why is the traditional environment that much different?
  - Most devs buy in to automation for cloud, but not traditional
  - Devs forget current environment and chase new, shiny
  - What always gets out, Windows
    - Difficulty automation
    - Better tooling with Linux
  - Difficulties with heterogenous environments
- How to get buy-in?
  - Often times executives are pushing so hard for cultural changes
    - Can be hard to adapt so quickly
    - Execs trying to flip devops lightswitch
    - Execs saw increase in op-ex after pushing so hard
○ Difficulty in government
  ■ Supporting legacy systems, mainframes
  ■ Security issues with using hybrid cloud
  ■ Need another step in the evolution
○ Can't turn all your sysadmins into developers
○ Not normally executive buy-in
  ■ People normally agree this is the path to use
  ■ Have legacy
    ● Mainframes
    ● Large databases
  ■ Executives may agree, but constrained by rapid change
    ● Trying to see where to go forward but also keep legacy in view
○ Need to drive team buy-in
○ Complexity due to size of organizations as well
○ Any success with getting buy-in?
  ■ Successfully deployed devops for legacy system
  ■ People who played ball got to get more work going ahead, devs happy
• You’ve decided on a DevOps approach, how do you start?
  ○ We did stuff, weren’t perfect. (PROCESS based)
    ■ Was scary at first
    ■ Setup JIRA
    ■ Saw efficiencies
    ■ Saw progress
    ■ Didn’t start with measuring team velocity, but eventually moved on to third or fourth iteration
    ■ Started to track burn down and velocity
  ○ Trying to come up with some sets of applications (RISK based)
    ■ Not being process focused but app focused
    ■ Worked with developers to identify candidates
    ■ Tackling low hanging fruit first
      ● Lowest impact
  ○ Tried to go for microservices approach and control domain (ARCHITECTURE focused)
    ■ Don’t want to tackle monolith that affects everything
  ○ DevOps is people, process, and tools
    ■ Tools is easier than people & process
    ■ People & process is hard
    ■ Need to consider how much change your employees can handle
      ● Everyone is going to lose some sort of expertise and skillset in a transformation
      ● There needs to be some sort of psychological safety in an organization
    ■ Accept that transformation is hard and difficult
● Fail faster
● Learn from failures
● Celebrate failures

● Use cases for DevOps, when to do it, and when not to do it
  ○ If you don’t have buy in to support  
    ■ Executive level support  
  ○ Don’t pretend to be agile  
    ■ Agile-fall  
    ■ Be honest with yourself and what you are doing  
    ■ Don’t use buzzwords to cover current practices  
  ○ Don’t use magical devops to fix infrastructure issues  
  ○ Is it possible to go to far into DevOps?  
    ■ Every change that was made they would rebuild everything  
    ■ Would kill current environment and rebuild  
    ■ Treated entire environment as a monolith  
    ■ Desire to move quickly has lead to shedding of frameworks  
      ● Using devops to circumvent procedure

Recommended Reading:

● DORA Research Papers:  
  ○ https://devops-research.com/research.html
● DevOps Reading List:  
  ○ http://uncontained.io/articles/devops-reading-list/
● DevOps on the Services Blog:  
  ○ https://servicesblog.redhat.com/?s=DevOps
● Open Practice Library, Frameworks for changing people, process, and culture  
  ○ https://openpracticelibrary.com/