

Effectively Organizing DevOps

Igal Koshevoy, Pragmaticcraft
Business-Technology Consultant
igal@pragmaticcraft.com
@igalko

What is devops?

- "Devops is a unified development and operations team, working together with shared goals."
- "Devops is the agile development methodology applied to operations."
- "Devops is about bringing a dev perspective to ops and an ops perspective to dev."
- "Devops is big picture thinking, using development to automate operations."
- "'Devops' is a new word for the old word 'systems engineering' in a information technology context."
- ...?

This talk is about:

- How to setup a new devops team
- How to improve the overall performance of dev and ops teams by merging them
- What technologies and social changes will make your team more successful

Do your research

What are your organization's:

- Wants (what they claim they want)
- Needs (what they actually need)
- Prevailing attitudes
- Stakeholders, contributors and customers
- Power structures
- Available resources

Do more research

- How have things worked in the past?
- What's the background of participants?
- What supporters can you enlist?
- What are the obstacles?
- Who can you have help overcome your own limitations?

Groundwork

- Make the human connection
- Bring diff. people together (e.g. lunches, pairing, etc) and rally around shared goals
- Educate devs, ops, users, their managers, executives, etc.
- Explain the benefits
- Explain the costs
- Explain the risks and their mitigation
- Be realistic, it's not a free lunch
- Get buy-in, each may need own incentive
- Beware of impacts on existing powers

Changing an organization's mission

- It's no longer a partitioned world, divided into "we build apps" or "we run servers"
- It's a unified "we create and deliver services" world
- Think bigger! Why do these apps and servers exist? Seamlessly supporting that is your mission

Heal the rifts of the past

- Why do your devs dislike ops? Why do your ops dislike devs? What problems is this bad relationship causing?
- Address the complaints with meaningful actions, collaborate for solutions
- Stop saying, "No! It's not my problem!" and start trying to understand, discuss and solve problems together

Strategic changes

Seek out **automation** because it provides:

- **Safety**, reduces risks and eases recovery
- **Reliability**, builds trust and consistency
- **Efficiency**, make better use of resources
- **Repeatability**, easily do it again correctly
- **Documentation**, describes your setup
- **Training**, helps add staff and cross-train
- **Foundations** for future, its an investment

Strategic changes

- Seek **agility**: Deliver value in short iterations that don't incur new technical debt. Review current needs. Repeat.
- Seek **effectiveness**: Make good use of your staff, time, resources, etc.
- Seek **empowerment**: Distribute responsibility and authority.
- Seek **innovation**: Don't get buried and miss own and others' improvements

Tactical changes

- Use agile methodology to improve team's trust and confidence while creating value
- Cleanup messes, this really helps morale
- Work in mixed teams
- Have team members really work together
- Cross-train members to share knowledge
- Praise those that contributed to success of the project, not just the apparent hero
- Make better use of people's abilities and interests to improve results, avoid burnout

Change management

- Make it easy to report issues without shooting messenger or placing blame
- Keep your issue tracker healthy
- Evaluate quickly to determine urgency and have way to expedite urgent work
- Schedule work and resources
- Involve affected parties
- Discuss risks and plan mitigation
- Review and trial, return for improvements
- Apply work and monitor results

Change management approaches

- Regularly-scheduled all hands meetings
- Ad hoc meetings with just the involved parties
- Asynchronous workflow with the involved parties
- Etc.

Technical changes

- Get code into revision control
- Get data backed up
- Automate
- Virtualize
- Modularize your work into reusable pieces
- Develop in per-user sandboxes
- Monitor availability of services

Technology changes

- Automate builds
- Automate testing
- Automate deploys
- Automate exception reporting
- Automate monitoring
- Automate provisioning
- Automate setup
- Automate maintenance
- Automate ...? :D

Change attitudes

- We're all in in this together
- Stop blame games and finger pointing – work together to solve those problems
- Understand your impact on others: e.g. don't login to production machines to make changes, this hurts everyone
- Improve the workflow rather than skipping it
- Practice continuous improvement
- Participate in internal and external training
- Pay attention to and deliver quality

Questions & Answers

Igal Koshevoy
Biz-Tech Consultant
igal@pragmaticcraft.com
@igalko