



MINI-BOOK

MB02 - April 2020

How to choose tools for DevOps and Continuous Delivery

Optimizing software
tool selection for
learning, collaboration,
automation, and team
dynamics

Overview

Modern approaches to software tool selection go way beyond the **simple checklists of yesteryear**.

When choosing tools for DevOps and Continuous Delivery (CD), high-performing organisations know to take into account the **social dynamics** of the organisation and the trajectory of the **rapidly-evolving public cloud** vendors.

In this report, we help you to choose tools in a way that emphasises **collaboration, automation, learning, reproducibility**, and the relationship between teams and software (**Conway's Law**).



Contents

- 3 Overview – choosing tools for DevOps and Continuous Delivery
- 4 Choose tools in a way that enables collaboration
- 6 Favour tools with APIs for automation
- 8 Understand the learning challenges for staff
- 10 Avoid special tools that exist only in one environment
- 12 Making use of Conway's Law for tool choices
- 17 Next steps

Tools to help collaboration

Key takeaways:

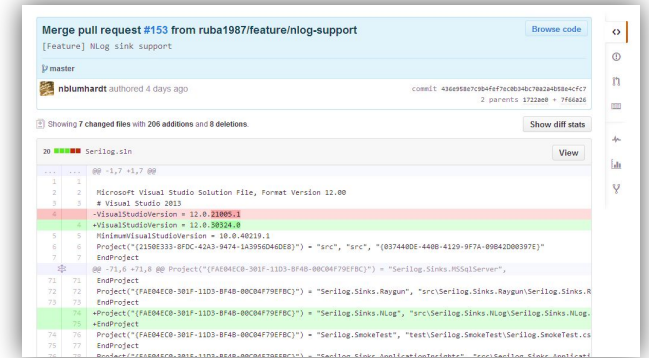
- Value collaboration as a key selection aspect
- Look behind the tool's main purpose to find collaboration opportunities
- Ask “how does our use of this tool help or hinder people in collaboration?”

Tools to help collaboration

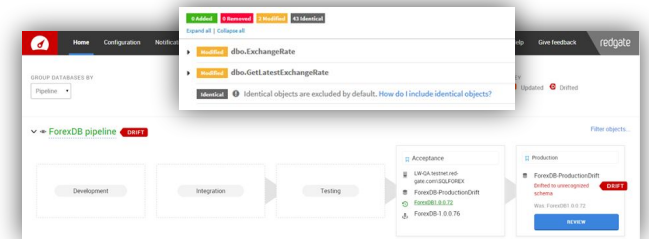
We don't need "a collaboration tool" to get collaboration benefits from our tools. Subtle differences in **the way we deploy or use tools** can reap huge rewards in improved working patterns.

Access to tools can have a major impact. If only developers can login to the version control system, that hinders collaboration with other groups. But if access is available to IT operations, product managers, and testers, **rich collaboration opportunities** open up.

Consider the **User Experience (UX)** of different groups when choosing how to use tools: a simpler tool may be better.



GitHub's in-browser visual diff helps coders and non-coders to work together on software changes



DLM Dashboard from Redgate helps database administrators and developers to collaborate on database changes



Favour tools with APIs

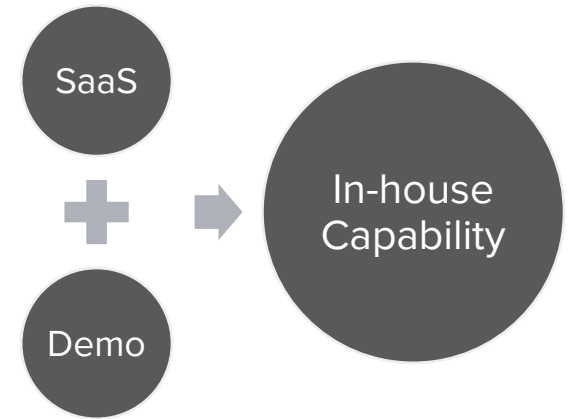
Key takeaways:

- Choose tools that expose APIs
- Point-and-click is no longer acceptable
- Aim for composition of new capabilities from multiple API-driven tools
- Build and deployment are first-class concerns



Favour tools with APIs

Modern software development needs the delivery tooling to be highly automatable. This requires a **fully-featured API** for each tool, preferably HTTP-based. Expect to **compose capabilities** by gluing together API-rich tools, enabling easy wiring for alerts and other events. Tools such as Slack provide a **rich integration of human and machine** activities for rapid decision-making and flow.



When adopting new tooling, consider starting with SaaS-hosted offerings whilst running internal prototypes/demo versions before building an internal capability



Older-generation tools had limited options for configuration via text files, relying on manual operator setup. Modern tools should be **configurable via text-based config files** that can be stored in version control, and **installation should be fully automated via packages**.

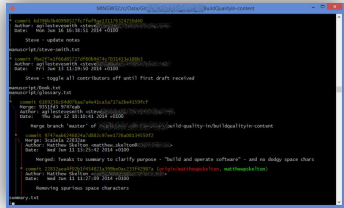


Encourage learning

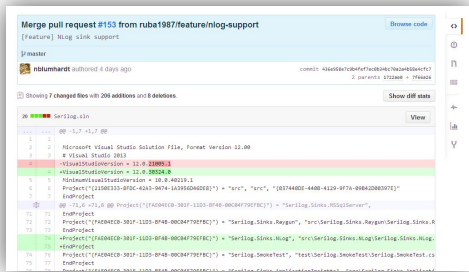
Key takeaways:

- Bring people with you on your DevOps & CD journey
- Appreciate current skills
- Prefer achievable gains now over possible future state
- Avoid fear of too-scary tools

Encourage learning



Command-line tools can be daunting for some people and may hinder collaboration unless training is provided



Tools with a more friendly UI can help to bring people onboard new ways of working, giving them the confidence to adopt command-line tools later

Some organisations make the mistake of selecting "best of breed" tools without considering their **ease of use**. Assess the skills in your organisation and **devise a tooling roadmap** for moving teams to new and improved ways of working.

Select tools that offer more than one way to use them (GUI, API, command-line) so **people can self-pace their learning**.

Focus on the **User Experience (UX) of learners** and meet their needs.

Avoid leaving people behind on the "climb" to more advanced approaches by holding **regular show-and-tell sessions** from peers and senior staff to demonstrate tools and techniques.



Avoid special 'singleton' tools

Key takeaways:

- Production-only 'singleton' tools break feedback loop (learning)
- Singleton tools make CI/CD more difficult
- Singleton tools underestimate the value of collaboration and learning



Avoid special 'singleton' tools

It is tempting to run special tools or special versions of tools in Production only due to high license costs. These 'singleton' tools tend to accrue an **aura of magic**, leading to people thinking "Production is special" and becoming disengaged: a bad outcome.

For effective DevOps and Continuous Delivery, we need to **enhance team learning and engagement**. Choose tools that can be rolled out widely, and that **help teams to understand how Production works**.

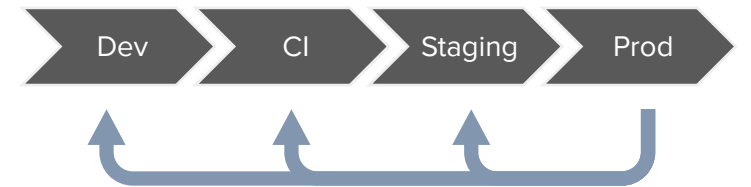


Select tools that work well and can be **easily instantiated** in non-Production environments (Dev, CI, Staging, etc.)

Optimise globally across the teams that need to collaborate, not just locally for Production.



Production-only tools prevent teams from learning because Production is treated specially.



Running the same tools in Production as in all other environments enables rapid learning and increases engagement within teams.

Conway's Law: communication drives architecture

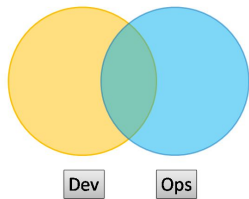
Key takeaways:

- See the whole organisation as a system that we're building
- Separate tools for separate teams
- Shared tools for collaborative teams

Communication drives architecture

“organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations”
– Mel Conway

Type 1 – Smooth Collaboration

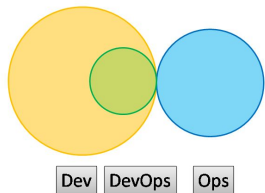


Collaborating teams need some shared tooling to be effective

Conway's Law has been observed and measured in many studies in recent years. The **communication paths in our organisation drive the resulting system architecture.**

Avoid selecting a single tool of one type across the whole organisation without considering the **team inter-relationships** first.

Type 3 – Infrastructure-as-a-Service



Teams providing or consuming a service have no need to share tools and forcing shared tools may be counterproductive

We therefore need to be mindful of **the effect of shared tools** on the way in which teams interact. If we want teams to collaborate, then shared tools make sense, but **if we need a clear responsibility boundary between teams, separate tools may be best.**



Key takeaways

- Value collaboration aspects
- Insist on APIs and scripted deployments
- Avoid a learning mountain: evolve tooling
- Avoid Production-only tools
- Consider Conway's Law



The authors

Matthew Skelton

Head of Consulting, Conflux

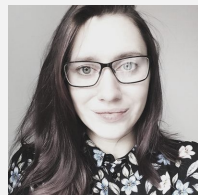
@matthewpskelton



Jovile Bartkeviciute

Research Lead at Conflux

@jovilebart



Further info

- *Continuous Delivery:*
confluxdigital.net/continuous-delivery
- *DevOps:*
<https://confluxdigital.net/devops>
- *Operability:*
<https://confluxdigital.net/operability>
- *SRE:*
<https://confluxdigital.net/sre>

Team Guide series

pocket-sized insights for software teams

#Operability #BusinessMetrics #Testability #Releasability

teamguidesforsoftware.com



Next Steps

Modern software delivery needs engaged teams and smart choices. We have significant experience in helping organisations to adopt and accelerate their DevOps and Continuous Delivery efforts. We publish **our high-value insights** into software delivery and IT operations to help leaders like you succeed.

Visit confluxdigital.net to learn more about **consulting** and **training** for DevOps and Continuous Delivery, together with our collection of **publications** and **guides** to help you on your DevOps and Continuous Delivery journey.

Follow us on Twitter: [@ConfluxHQ](https://twitter.com/ConfluxHQ) | Follow us on LinkedIn: [confluxhq](https://www.linkedin.com/company/confluxhq)

Copyright © 2017-2021 Conflux Digital Ltd. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

