

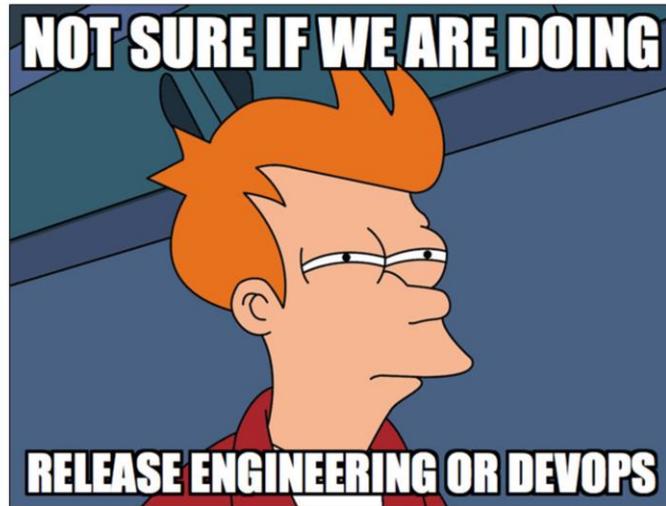
Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Towards Definitions for RelEng and DevOps

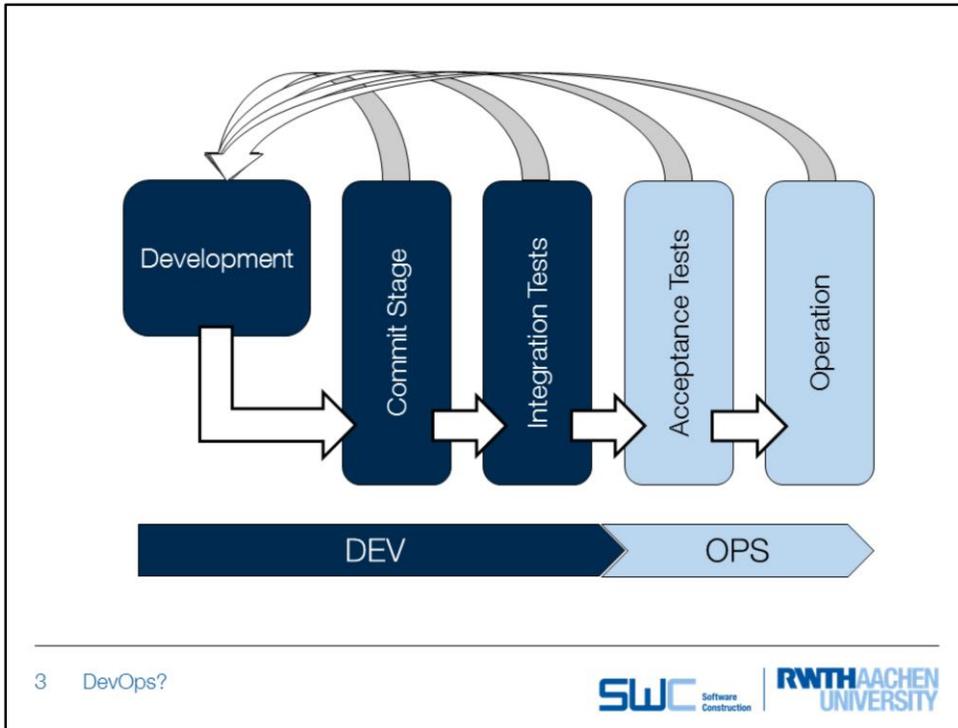
Andrej Dyck, Ralf Penners, Horst Lichter



Two years ago, I watched John O'Duinn's keynote talk "RelEng as a force multiplier" and I was fascinated by RelEng. However, I found it quite difficult to explain people what it is about. Last year, Bram Adams started this workshop pointing out "look, a lot of the topics at DevOpsDays are actually RelEng tasks". So I looked into DevOps and got confused. Where was the difference? I am Andrej Dyck, a research assistant at RWTH Aachen University's chair for Software Construction. Today, I would like to propose two definitions -- one for RelEng, one for DevOps -- in order to differentiate both terms.



I am sure that you have discussed RelEng vs DevOps and have some definitions in mind. The truth is, many companies do not have a clue. We work with some large software companies for insurance companies or the energy market and they – as many -- struggle to deliver high-quality software fast, reliable, and predictable. Obviously, they want to improve, and thus, trying to do things differently while labeling certain aspects with the first buzz word they hear. And I do not blame them: Many inadequate or even wrong interpretations for RelEng as well as DevOps exists. Consequently, these terms are often confused, misinterpreted, or used as synonyms.



Here is an example of a company's diagram we discussed with that company only a few weeks ago. I am not allowed to show the original, however it boils down to this: they have a development team, which develops and commits features. A test team then responsible for integration tests. Deciding to deploy a certain release the operations team do acceptance tests and deploy the software. Each silo can throw back a feedback package to the development team. They call that DevOps!

What is wrong with existing definitions?

Ambiguous, imprecise, wrong.

So, you might ask yourself: what is wrong with existing definitions and explanations? Well, existing sources often focus strongly on the how, i.e., what practices and tools to use, e.g., “implement continuous delivery and you are doing RelEng/DevOps”, while the meaning or ideas are often not at all explained or explained ambiguously.

Wikipedia’s definition – which btw is referenced by this workshop’s website – describes RelEng as “a sub-discipline in software engineering concerned with the compilation, assembly, and delivery of source code into finished products or other software components.” This definition is missing important aspects of improvement, reliability, and predictability.

Gene Kim’s definition: “The term ‘DevOps’ typically refers to the emerging professional movement that advocates a collaborative working relationship between Development and IT Operations, resulting in the fast flow of planned work (i.e., high deploy rates), while simultaneously increasing the reliability, stability, resilience and security of the production environment.” This definition focuses only on development and IT operations. Moreover, we strongly believe that the term DevOps can define more than just a movement.

A blog post argues that DevOps should be defined by the community and that Wikipedia should hold this definition; “This allows the definition to evolve and change over time as the industry evolves and changes”. We do not argue with that; however, a wrong definition on Wikipedia as discussed is, on one hand, bad for communicating the idea, and, on the other hand, not usable for scientific research. Moreover, there are many similar cases where some people thought a definition is not needed, e.g., the agile movement, which is now defined by ISO/IEC/IEEE 26515:2011.

Wikipedia states that release engineering is “also known as DevOps”. This even contradicts all definitions that are known to us.

Thus, our goal is to formulate definitions – one for DevOps, one for RelEng – as simple, yet precise definitions as possible while considering the important aspects.

This work is not finished ...



Mind that the following definitions are proposals to begin with. We hope that this will ignite researchers and practitioners to come up with definitions that describe DevOps and RelEng well and regardless of the scale of their application.

Release engineering

deploy high-quality software
reliably and predictably.

Discussing release engineering we observed that the deployment of high-quality software is a central goal of release engineering. A deployed faulty software product decreases the customer added value and diminishes its reputation. Moreover, the deployment has to be fast, reliable, and predictable.

Release engineering

concerned with the development,
implementation, and **improvement of
processes to deploy** high-quality software
reliably and predictably.

Based on O'Duinn's "building a pipeline", we see three main tasks of a release engineer: development, implementation, and improvement of the deployment process. Starting with the development of a custom release process for an organization, suitable methods and tools have to be established and introduced. Then, after a reliable and predictable release process is established, it needs to be improved, in order to increase efficiency, and eventually, the product value. Note that, here, improving processes also mean, i.a., improving information flow between the people involved in the project. In order to improve the release process, a release engineer has to establish a good communication between those responsible for various tasks and stress acceptance for other team's goals.

Release engineering

is a software engineering discipline concerned with the development, implementation, and improvement of processes to deploy high-quality software reliably and predictably.

And thus, similar to, for example, requirements engineering, we think that release engineering is a discipline that needs careful planning, execution, and audit.

DevOps

stresses **empathy** and **cross-functional collaboration** within and between **teams**,

In general, there seems to be common ground that DevOps is about improving communication and collaboration between teams in an organization or project. However, the purpose, i.e., why an improved collaboration should be achieved, is often missing or differs strongly.

We even argue that empathy, i.e., the capacity to understand what another person is experiencing, is not enough and altruism, i.e., the wish that other people may be happy, should be practiced.

DevOps

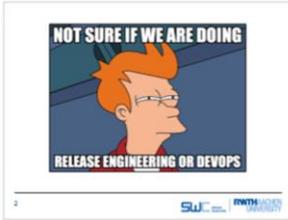
stresses **empathy** and **cross-functional collaboration** within and between teams, in order to **operate resilient systems** and **accelerate delivery of changes**.

On one hand, the goal of DevOps is the acceleration of feedback, higher quality of software, and faster deployment. On the other hand, DevOps is about more than fast deploying high-quality software; “it is about operating software services [and delivering] change to those services”. In other words, both, development and operational goals, are equally important, as developing and maintaining systems should be performed by developers and operators jointly. Here, the term resilient systems stresses the focus on software and its production environment as a whole.

DevOps

is an organizational approach that stresses empathy and cross-functional collaboration within and between teams, in order to operate resilient systems and accelerate delivery of changes.

Considering the goals and manners in more detail, we see DevOps as an organizational approach to achieve those strategic goals. With this, we disagree with some definitions that DevOps is a development method. Nor is it a development process model like RUP, nor a project management framework like SCRUM; it doesn't define any processes, nor does it define any roles, activities of planning, organizing, and controlling. Moreover, we strongly believe that the term DevOps can define more than just a movement or culture.



Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Paper: <https://goo.gl/n1mW1n>
Presentation: <https://goo.gl/0gxWJh>



2



Release engineering

is a software engineering discipline concerned with the development, implementation, and improvement of processes to deploy high-quality software reliably and predictably.

4 Release Engineering



DevOps

is an organizational approach that stresses empathy and cross-functional collaboration within and between teams, in order to operate resilient systems and accelerate delivery of changes.

7 DevOps



Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Paper: <https://goo.gl/n1mW1n>
Presentation: <https://goo.gl/0gxWJh>



Release engineering

is a software engineering discipline concerned with the development, implementation, and improvement of processes to deploy high-quality software reliably and predictably.

DevOps

is an organizational approach that stresses empathy and cross-functional collaboration within and between teams, in order to operate resilient systems and accelerate delivery of changes.



Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Paper: <https://goo.gl/n1mW1n>
Presentation: <https://goo.gl/0gxWJh>



Release engineering

is a software engineering discipline concerned with the development, implementation, and improvement of processes to deploy high-quality software reliably and predictably.

DevOps

is an organizational approach that stresses empathy and cross-functional collaboration within and between teams, in order to operate resilient systems and accelerate delivery of changes.



THANKS!

Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Paper: <https://goo.gl/n1mW1n>
Presentation: <https://goo.gl/0gxWJh>

Do we need DevOps for RelEng and vice-versa?

Andrej Dyck
andrej.dyck@swc.rwth-aachen.de

Paper: <https://goo.gl/n1mW1n>
Presentation: <https://goo.gl/0gxWJh>

A Discussion

SWC Software
Construction

RWTHAACHEN
UNIVERSITY

Comparing software engineering to classical engineering, e.g., construction engineering, the latter rely strongly on both, established processes and good collaboration between teams (e.g., construction workers and architects). Accordingly, a good communication within and between teams influences the proper implementation of processes and the improvement of such requires collaboration. Hence, release engineers can establish processes more easily and effectively in organizations that implement DevOps, since people already communicate and work together. Likewise, a sophisticated release

engineering supports and intensifies the mindset of DevOps; here, release engineering works as a connecting link between teams. In conclusion, implementing both, release engineering and DevOps in an organization increase the effectiveness of productivity, and with that, eventually, the product value.

While release engineering can and should be applied by all software development organizations, the question of what organizations can implement DevOps arises. The operations part of the term suggest that DevOps is only applicable for organizations that operate their own software, e.g., web-sites. However, if, on one hand, widening the term operations to operating infrastructure (e.g., development, build, test, release, etc.) and, on the other hand, not differentiating between internal and external operations, then all software development organizations can implement DevOps. In fact, the heart of DevOps, i.e., the mindset of empathy or promise theory, is not limited to software development organizations.