Article

DesOps is "DevOps 2.0"

Samir Dash Principal Software Engineer (UXD)

DesOps = DevOps (2.0)

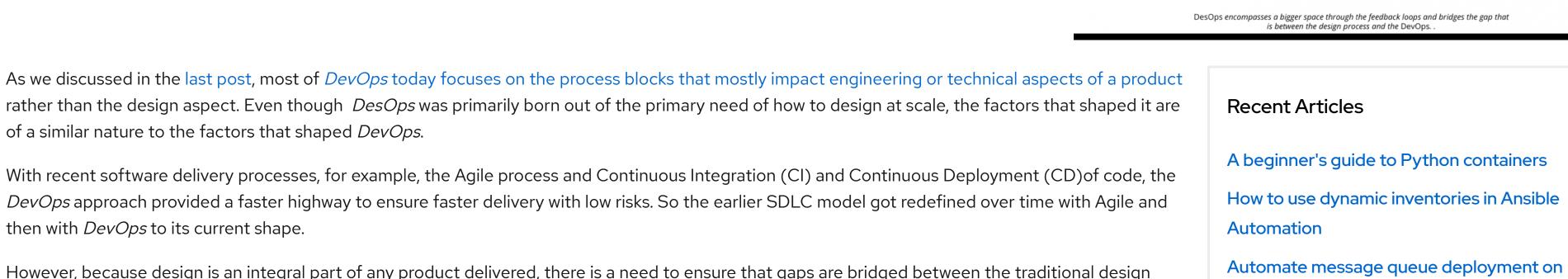
As we discussed in the last post, most of *DevOps* today focuses on the process blocks that mostly impact engineering or technical aspects of a product rather than the design aspect. Even though DesOps was primarily born out of the primary need of how to design at scale, the factors that shaped it are of a similar nature to the factors that shaped *DevOps*. With recent software delivery processes, for example, the Agile process and Continuous Integration (CI) and Continuous Deployment (CD)of code, the

then with *DevOps* to its current shape. However, because design is an integral part of any product delivered, there is a need to ensure that gaps are bridged between the traditional design lifecycle and the fast track of the *DevOps* development lifecycle. *DesOps* and *DevOps* both are complementary to each other. The design delivery process improvements try to optimize the overall delivery process and thereby contribute to *DevOps*, for example, in aspects such as testing of the

product that involves design aspects, usability, accessibility, etc. The need for tighter integration between the design team and the engineering team became a necessity to ensure to design at scale. During the past

two to three years, the top five big companies have made heavy investments in this area that have paved the way for other organizations and design communities to be more explorative in this area.

DesOps = DevOps (2.0)



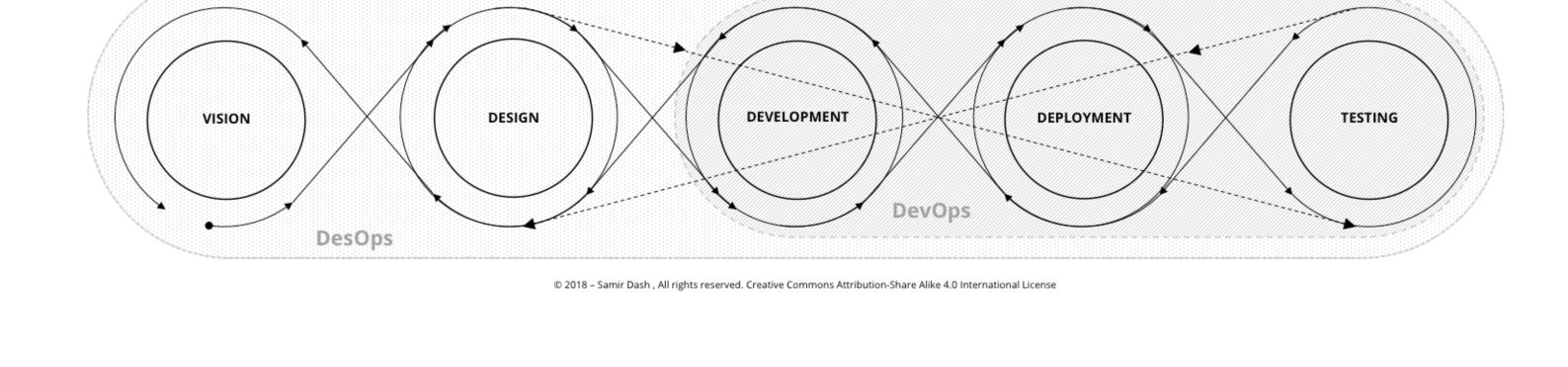
JBoss EAP

OpenShift

How we ensure statically linked

Presenting a new Istio operator on

applications stay that way



The implications of *DesOps* are reflected in the outcome, where the silos among the teams and disciplines get reduced. Along with this, *DesOps* improves the collaboration among cross-functional teams and work practices, which contributes to minimizing waste in the delivery process.

DesOps encompasses a bigger space through the feedback loops and bridges the gap that is between the design process and the DevOps. .

Every product lifecycle has one core goal towards which it strives: reaching customer delight by delivering the value. The design process associated with DesOps helps in understanding, capturing, and delivering that value.

But conventional business processes were more keen on getting the outputs of each process block, which can be fed into the next block, thereby reaching a stage that ultimately delivered the value. Many such practices failed in achieving customer delight because the processes used were not based on the customer shift from outputs to outcomes.

If we see the *DesOps* processes in terms of a value system, we will see at a high level that *DesOps* touches upon three major areas: 1. Understanding value

3. Capturing and delivering value

As discussed in the last post, in the value system approach, understanding value is about reaching a vision. Creating value is broadly about reaching a

block has

2. Creating value

roadmap with Minimum Viable Product (MVP). Capturing and delivery value is about running the backlog and sprints and ensuring delivery until the end users have access.

Each Process Business Objectives Ideation & Conceptualization Market Strategy Product Design **Product Strategy** Product MVP Market & User Research Product Roadmap

Market & Users Segmentations

Product Vision

overlaps with each other. DesOps is about having a process

part of the workflow.

approach to problem-solving.

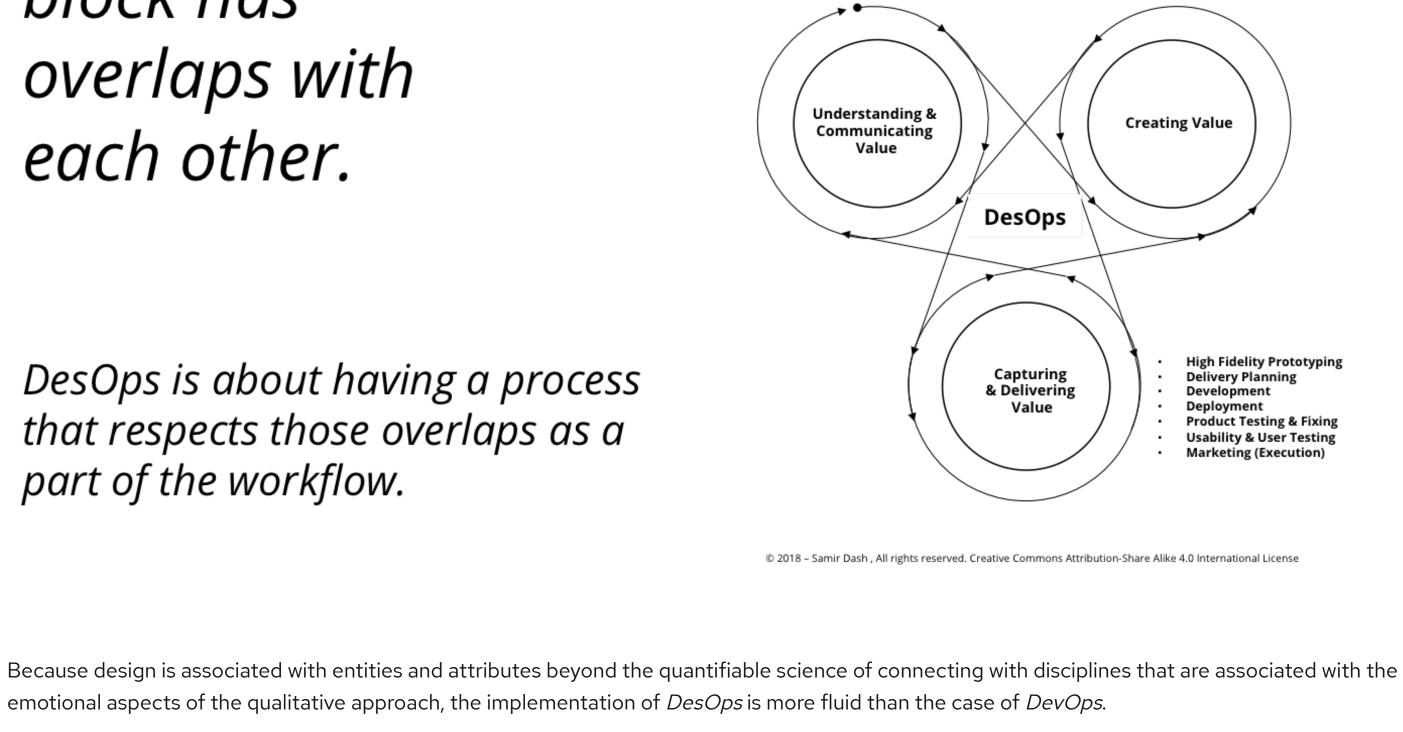
you can easily notice the pattern.

business aspects of the value chain.

Code Quality Checks

emotional aspects of the qualitative approach, the implementation of *DesOps* is more fluid than the case of *DevOps*.

that respects those overlaps as a



Low Fidelity Prototyping

As one example, you can see in the following figure that the right side shows the mapping of practices involved with software incident reporting. Here

User Testing

Customer Validations

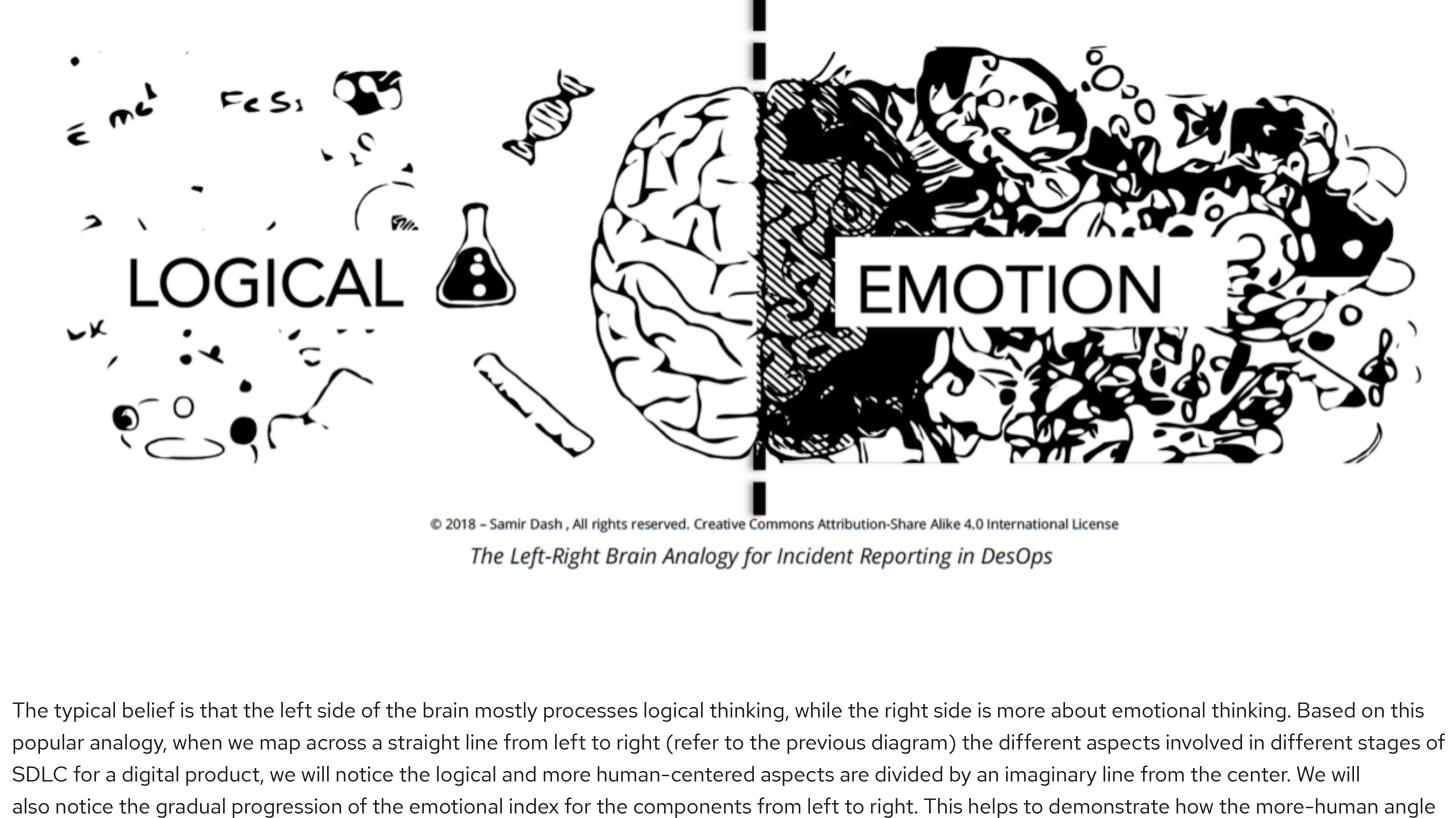
Accessibility Issues

API-UI Functional Bugs

For instance, if we plot the techniques and practices of any domain according to the left brain-right brain analogy, we can see that most of the practices

having soft-attributes or dealing with human emotions, purpose, and behavior will fall into the creative aspect, which involves some kind of design

A/B or Split Testing **Usability Issues Explorative Testing** UI/Front-end Bug **Backend Functional Issues** API Issues Data Testing Code Optimization Checks End-user Feedback **FEEDBACK**



3. Delivering incremental value 4. Investing in quality 5. Empowering team members 6. Setting up clear accountability in teams 7. Learning from experiences

(The ten golden principles to guide the Design Operations in your organization)

is involved as we move from the areas that DevOps touches upon to the areas that DesOps touches, because DesOps touches upon the design and

The Ten Commandments of *DesOps*

In such a context, if we look into *DesOps*, it makes a lot of sense in the following key principles:

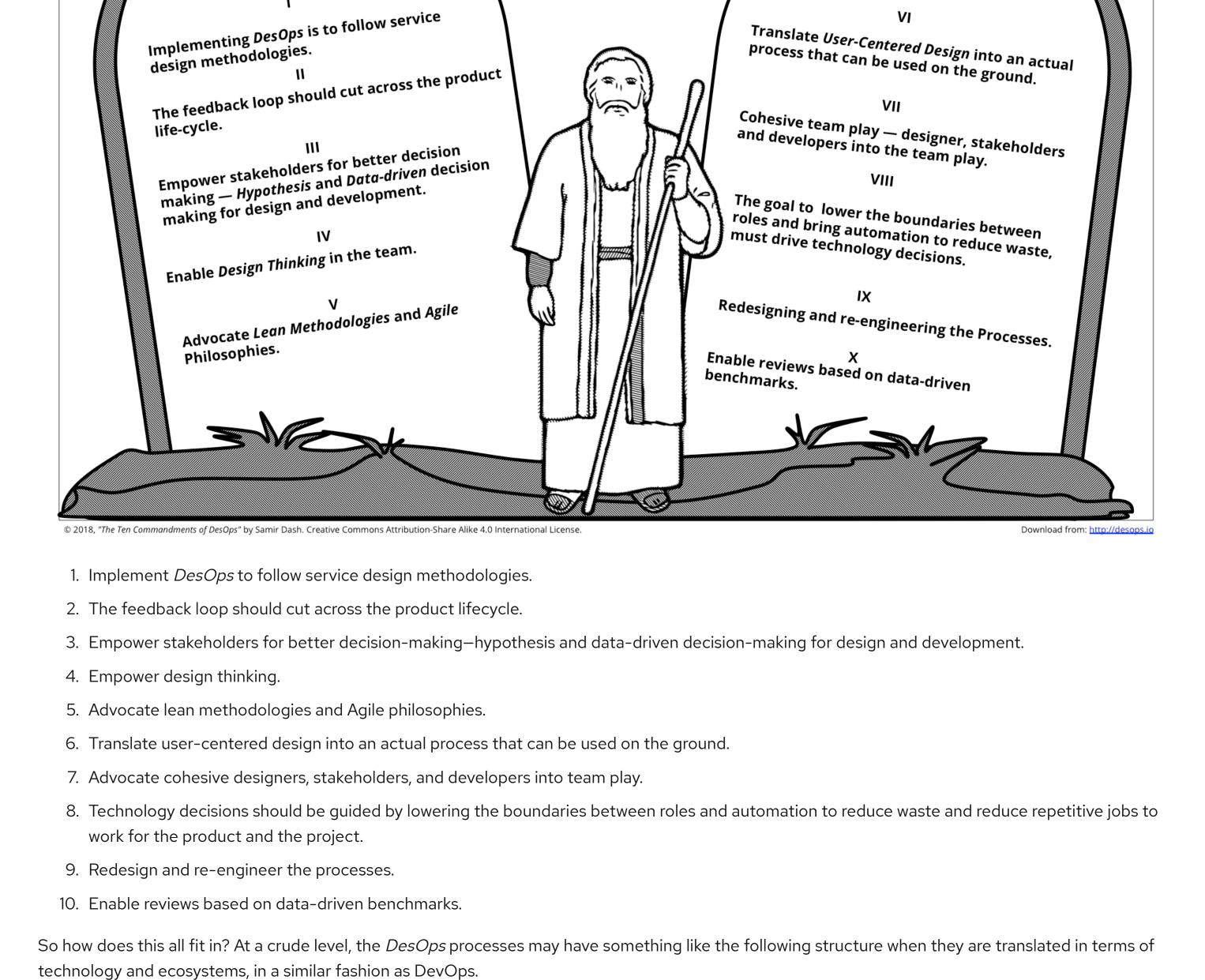
From foundational guidelines, *DevOps* inspires the *DesOps* mindset at a high level:

1. Partnering with customers for improving business value

8. Advocating open communications and transparency

2. Working together towards a shared vision

9. Being agile and adapting to change



1. First, we should create the design benchmarks (that includes qualitative as well as quantitative metrics) from the information at the design stage that can be used in comparing the product features against metrics based on this design benchmark. 2. Then, automate and perform manual tracking of the product during runtime (in real time and in the true context), and then categorize and collate

4. Collect all standards and specifications on different aspects of heuristics to ensure that at least at the basic level the standard principles are

5. On the ground, in the context of the eco-system and technologies, build the critical components that would collect and process all the data

STANDARDS

& SPECIFICATIONS

Collect Standards &

Specifications

3. This involves creating features to support the user feedback cycle and user testing aspects (exploratory, split testing capabilities).

FEEDBACK

& USER TESTING

Data from User

feedback & User

4

MODEL

& MTERICS

Model the data and

Map against the

A.I. & COGNITIVE

Al helps finding the

outliers &

generate Insights / recommendations.

Got it

1 Login ▼

(a)

COMPUTER VISION

Process

Unstructured data

to get inference.

collected in all these stages and generate the desired metrics and inferences and also contribute in bringing continuous integration and continuous delivery blocks to run the process. 6. Build the unit to generate the model to map the data and compare it against the metrics.

7. Build the cognitive unit that can compare the data and apply the correct models and metrics to carry out the filtering of the data and generate

8. And ensure in all these stages that the feedback loop is connected spatially and acting as a meaningful neural network that helps in informed

Keep tuned in to stay with me on this journey into DesOps! (Note: Based on my book *The DesOps Enterprise: Overview & Culture*)

Comments Red Hat Developer Comment Policy

Please keep your comments relevant and polite. Opinions shared in comments are not official Red Hat news or policy.

the insights which can be shared as actionable output to the end-user/customer.

1 Comment

DESIGN

BENCHMARKS

Collect Benchmark

data from design

stage

this data.

followed.

decision-making.

Last updated: September 3, 2019

Please read our Comment Policy before commenting.

Developer Sandbox

Interactive Tutorials

Operators Marketplace

Developer Tools

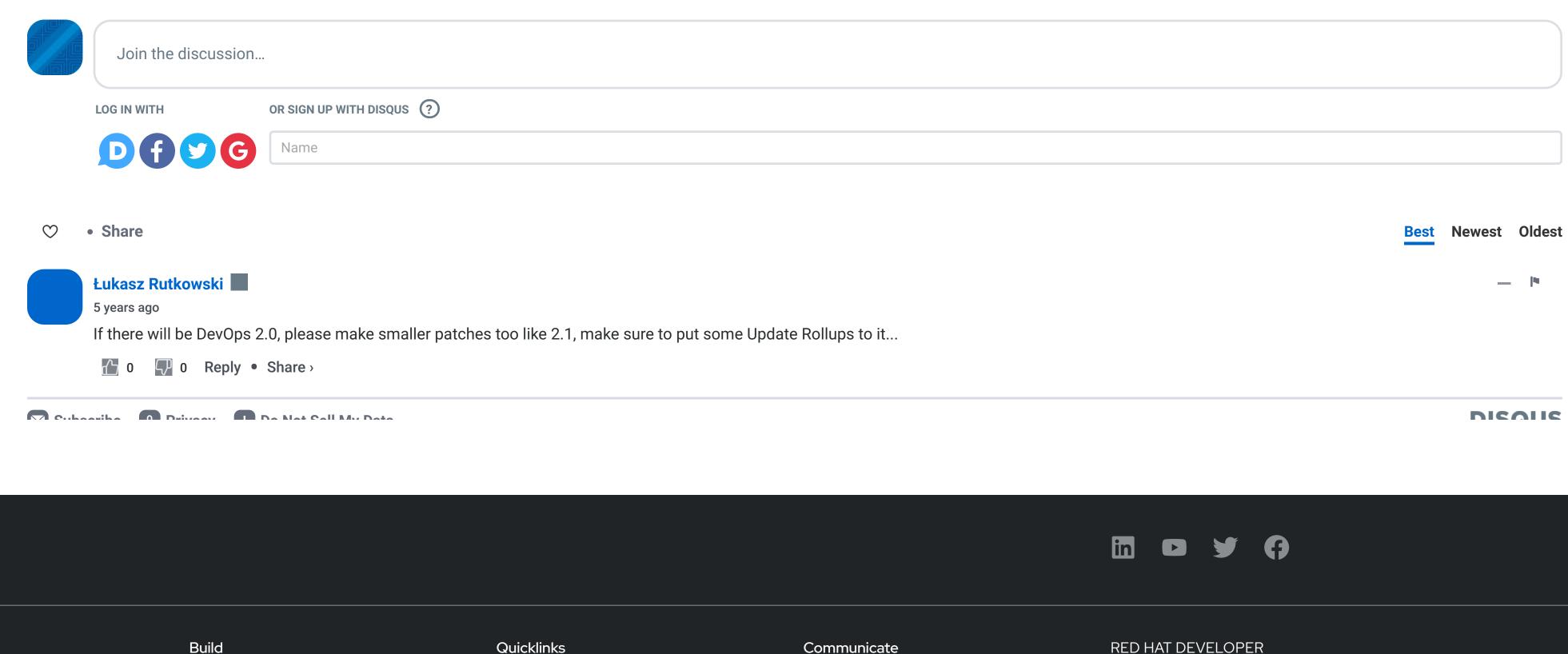
API Catalog

Terms of use

AUTOMATED

Track & Collect data

from product



About us

Contact sales

Find a partner

Site Status Dashboard

Report a security problem

Build here. Go anywhere.

project manager or team lead.

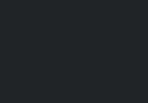
Sign me up 🔿

We serve the builders. The problem solvers who create careers with code.

Join us if you're a developer, software engineer, web designer, front-end

We've updated our Privacy Statement effective July 1, 2023

designer, UX designer, computer scientist, architect, tester, product manager,



Red Hat Developer

Products

Red Hat Enterprise Linux

Red Hat Ansible Automation Platform

Red Hat OpenShift

See all products

See all technologies

Learning Resources

E-books

Blog

Events

Cheat Sheets