ESEC/FSE

DECAL Lab

One Size Does Not Fit All:

An Empirical Study of Containerized Continuous Deployment Workflows

Yang Zhang, Bogdan Vasilescu, Huaimin Wang, Vladimir Filkov

November 7, 2018

国防科技大学 National university of defense technology

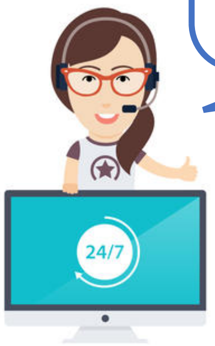
STRUDEL Carnegie Mellon University

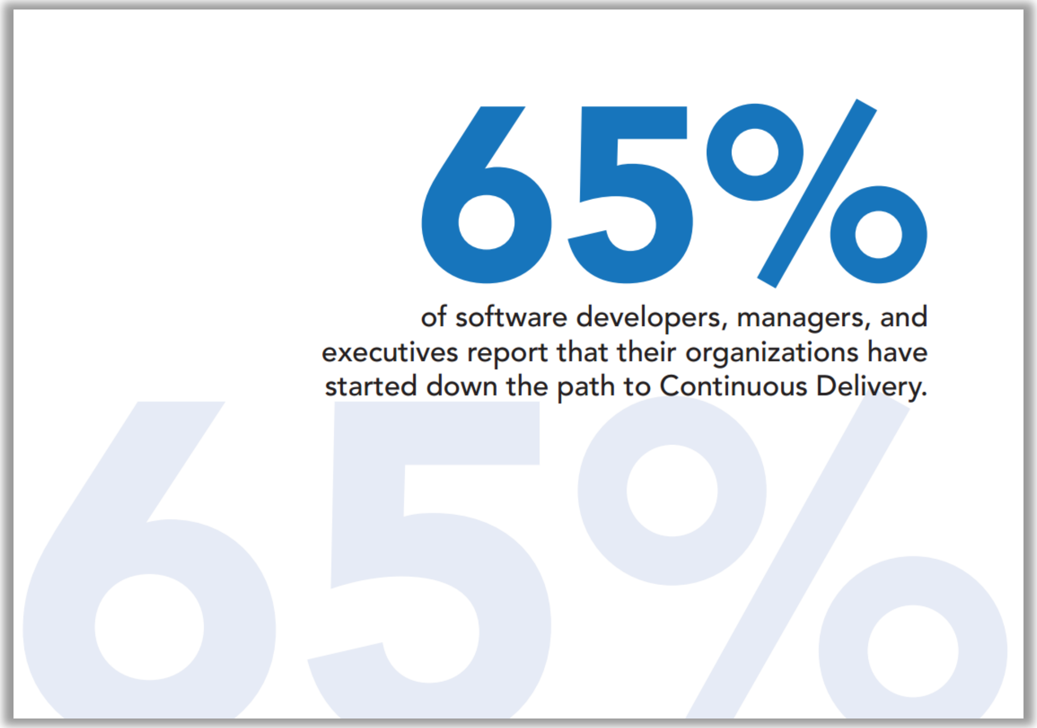
UC DAVIS UNIVERSITY OF CALIFORNIA

Continuous Delivery/Deployment

A software engineering approach in which software functionalities are delivered frequently through automated deployments.

It is the ability to release software whenever we want…it could mean every check-in goes straight to production…it is the ability to deploy at will.



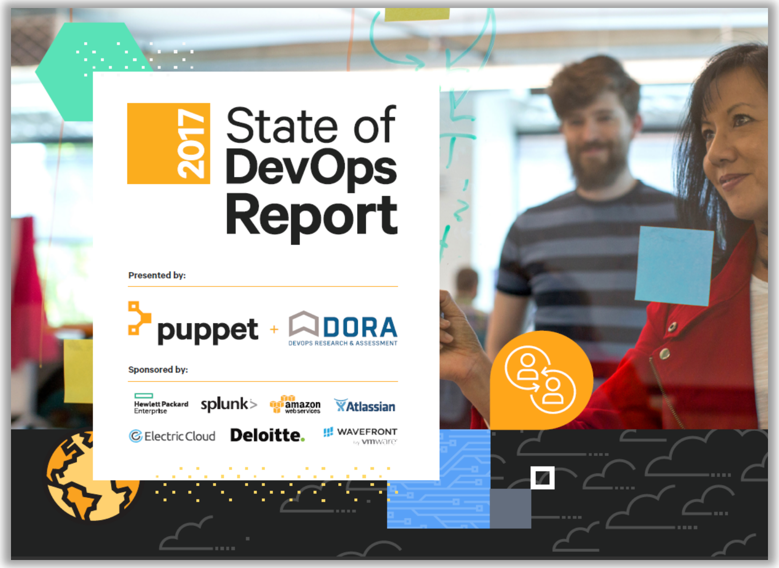


65%

of software developers, managers, and executives report that their organizations have started down the path to Continuous Delivery.

"On the journey to continuous deployment: Technical and social challenges along the way". Information and Software Technology. 2015.
"Continuous delivery? easy! just change everything (well, maybe it is not that easy)". AGILE. 2013.
https://www.perforce.com/sites/default/files/files/2017-09/continuous-delivery-report.pdf

Notable Benefits



2017 State of DevOps Report

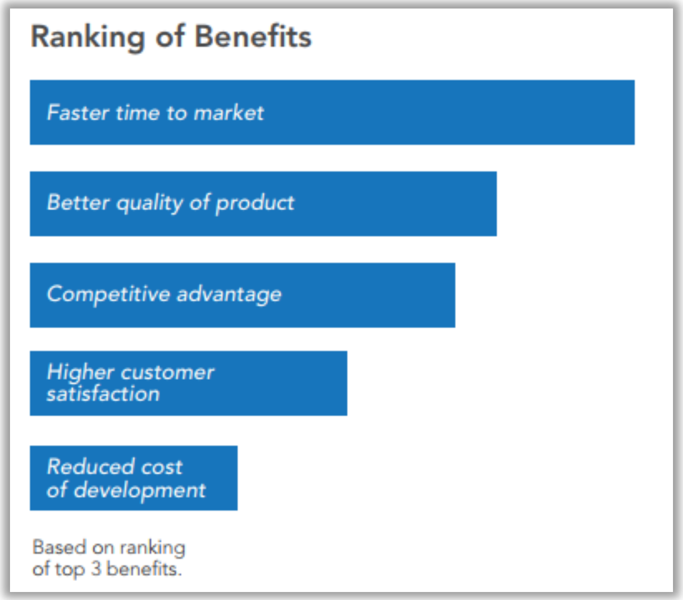
Presented by:

puppet + DORA DEVOPS RESEARCH & ASSESSMENT

Sponsored by:

Hewlett Packard Enterprise splunk> amazon web services Atlassian

ElectricCloud Deloitte. WAVEFRONT by vmware



Ranking of Benefits

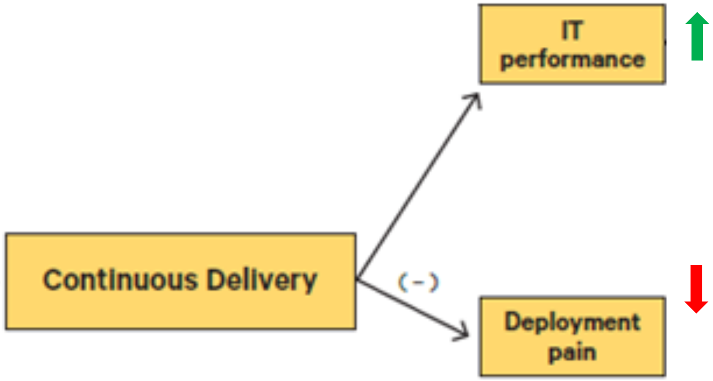
Based on ranking of top 3 benefits.

PERFORCE

Continuous Delivery: The New Normal for Software Development

Findings from Evans Research Survey of Software Development Professionals

Commissioned by Perforce Software



IT performance

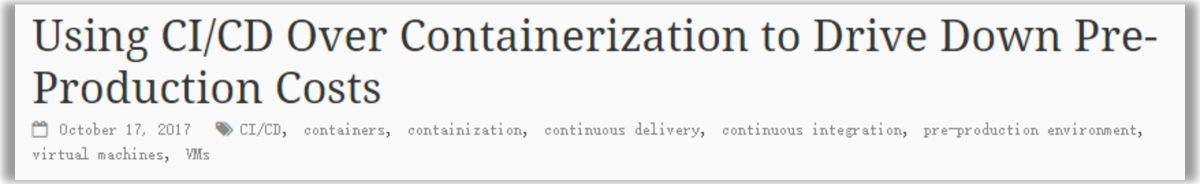
Continuous Delivery

(-)

Deployment pain

https://puppet.com/resources/whitepaper/2017-state-of-devops-report
https://www.perforce.com/sites/default/files/files/2017-09/continuous-delivery-report.pdf

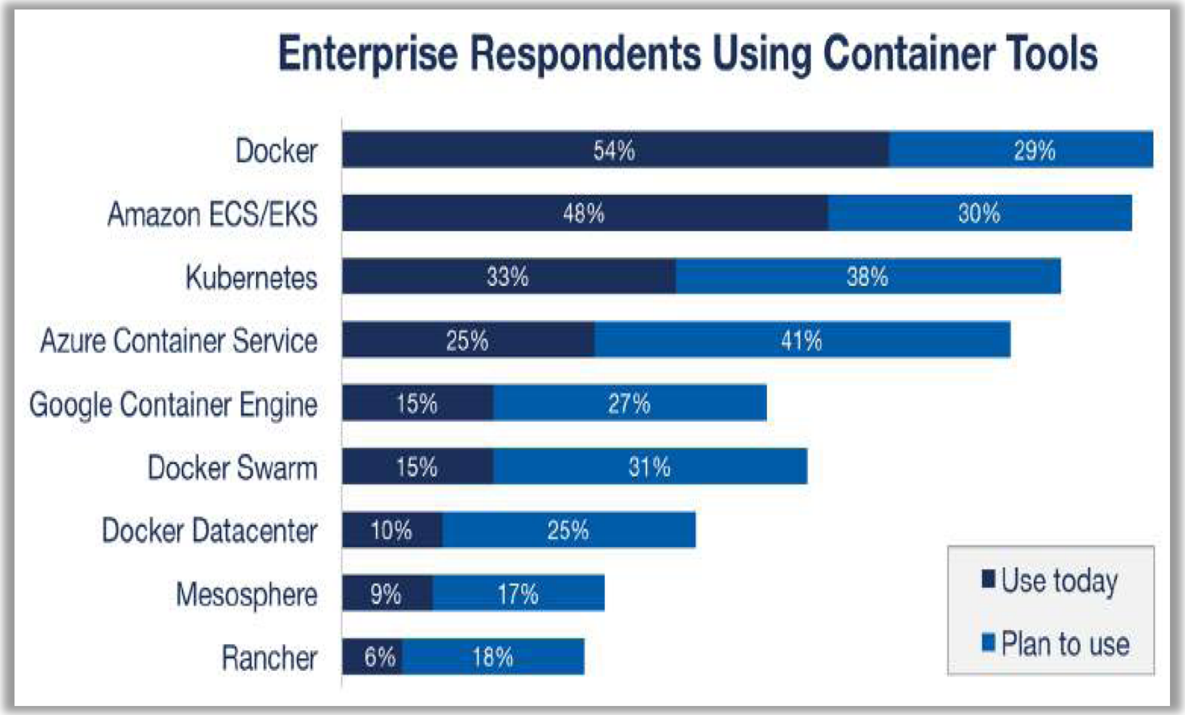
Containerization & Docker



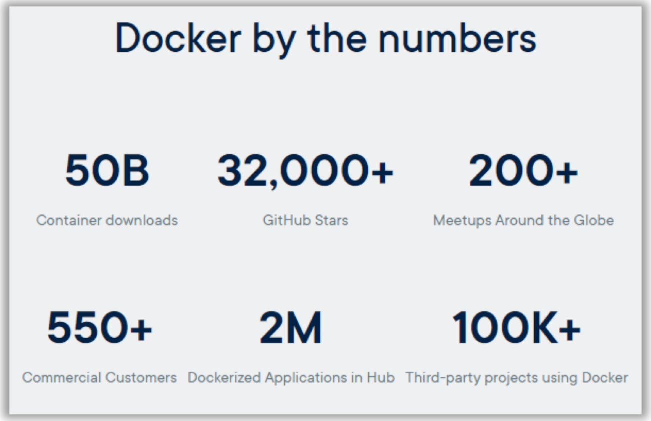
Using CI/CD Over Containerization to Drive Down Pre-Production Costs

October 17, 2017 CI/CD, containers, containization, continuous delivery, continuous integration, pre-production environment, virtual machines, VMs

- •Faster time to market
- •Optimum use of infrastructure
- •One-click infrastructure provisioning and decommissioning



docker

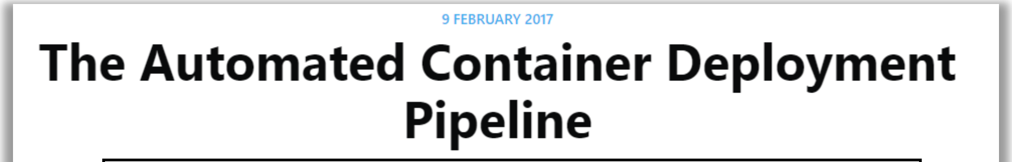


Docker by the numbers

50B	32,000+	200+
Container downloads	GitHub Stars	Meetups Around the Globe
550+	2M	100K+
Commercial Customers	Dockerized Applications in Hub	Third-party projects using Docker

https://containerjournal.com/2017/10/17/using-cicd-containerization-drive-pre-production-costs/
https://www.suse.com/media/report/rightscale_2018_state_of_the_cloud_report.pdf

Containerized CD Workflow/Pipeline



9 FEBRUARY 2017

The Automated Container Deployment Pipeline

> *"The truth is, that containers really make some things easier and more manageable but you still have to use them properly. One area where containers can really make a difference is the automated deployment pipeline."*

FOCUS: RELEASE ENGINEERING

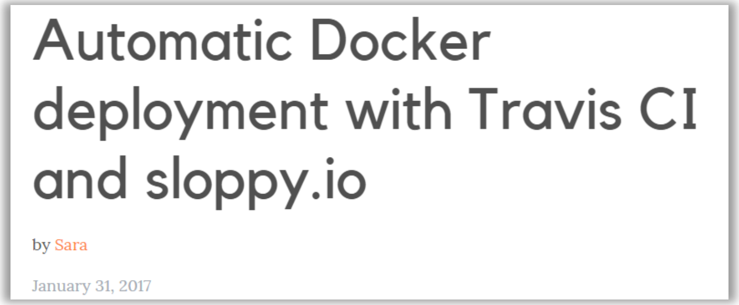
> *"Research is needed to identify these processes (covering areas of business, software development, operations, and so on) and develop and verify alternatives that suit CD."*

Delivery

Huge Benefits, but Challenges Too

CircleCI Blog

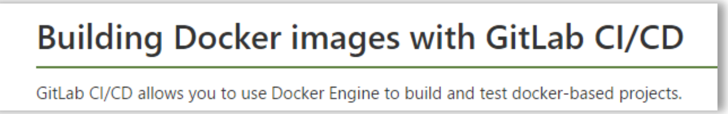
How to build a CI/CD pipeline with Docker



Automatic Docker deployment with Travis CI and sloppy.io

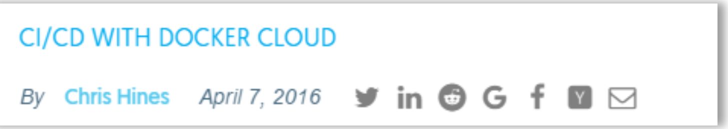
by Sara

January 31, 2017



Building Docker images with GitLab CI/CD

GitLab CI/CD allows you to use Docker Engine to build and test docker-based projects.



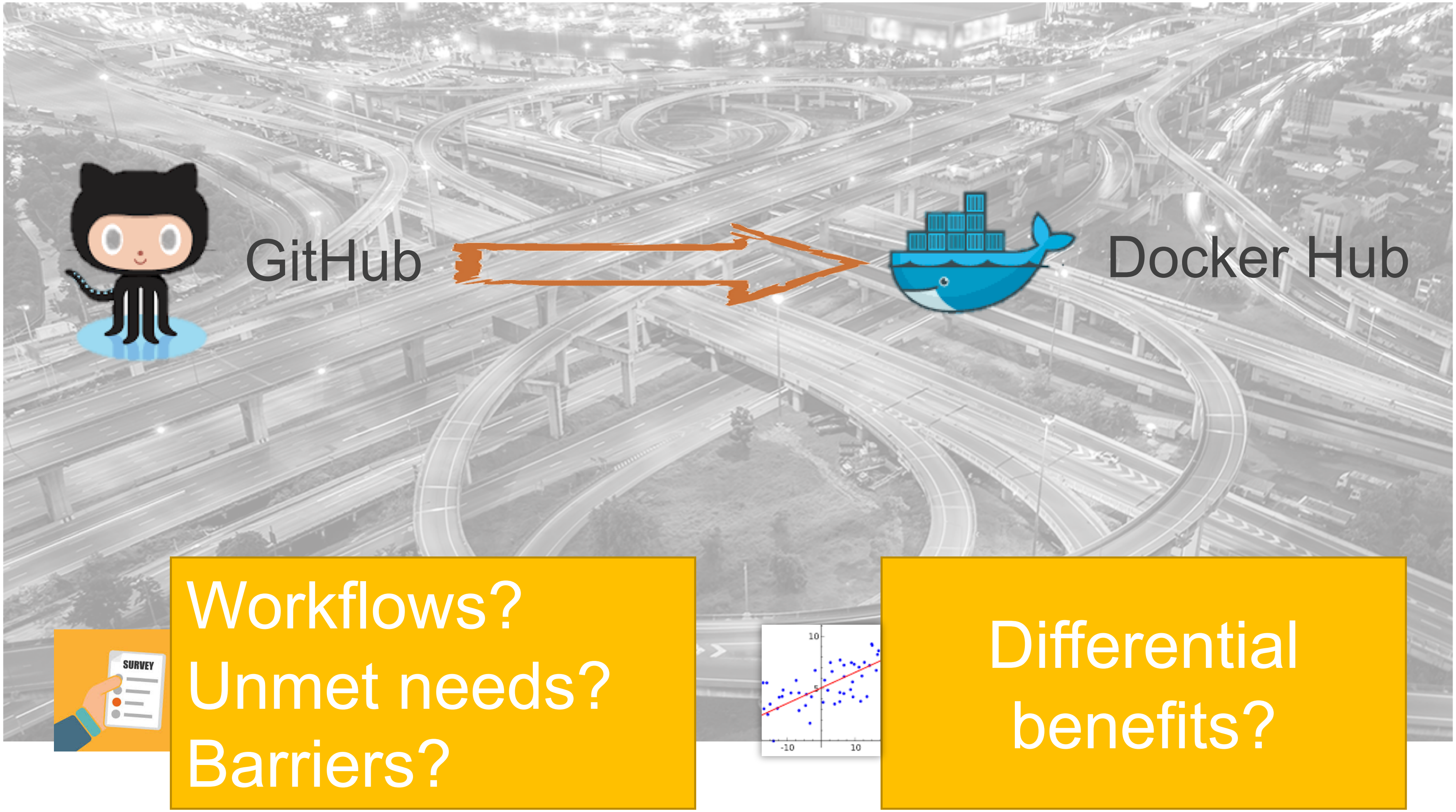
CI/CD WITH DOCKER CLOUD

By Chris Hines April 7, 2016

Continuous Delivery Huge Benefits, but Challenges Too. IEEE Software. 2015.
https://ghost.kontena.io/container-deployment-pipeline/
https://circleci.com/blog/build-cicd-piplines-using-docker/
https://sloppy.io/en/blog/automatic-docker-deployment-with-travis-ci-and-sloppy-io/
https://docs.gitlab.com/ee/ci/docker/using_docker_build.html
https://blog.docker.com/2016/04/cicd-with-docker-cloud/

Our Work: How OSS Projects Use Docker-enabled CD Workflows on GitHub

DECAL Lab



GitHub

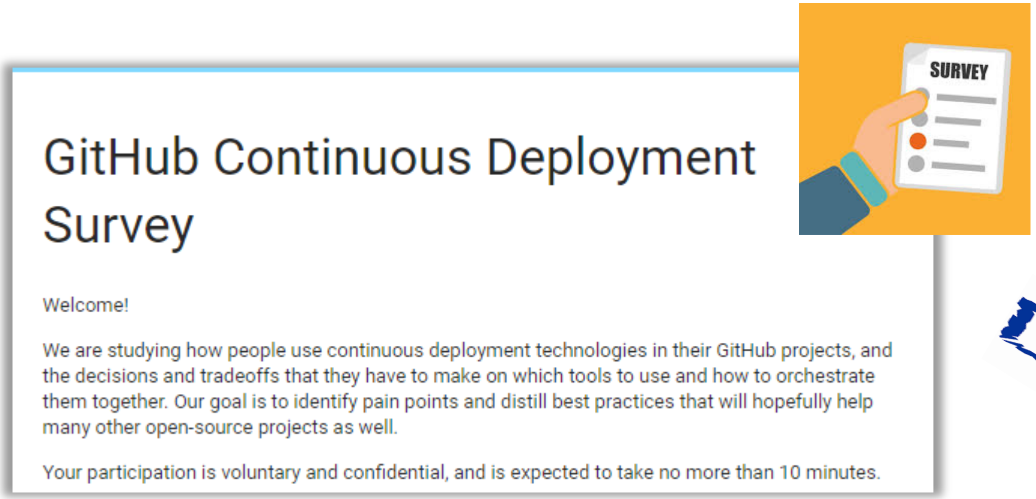
Docker Hub

Workflows?
Unmet needs?
Barriers?

Differential benefits?

https://blog.advaoptical.com/en/many-roads-to-400g-in-the-data-center

Approach: Mixed-methods



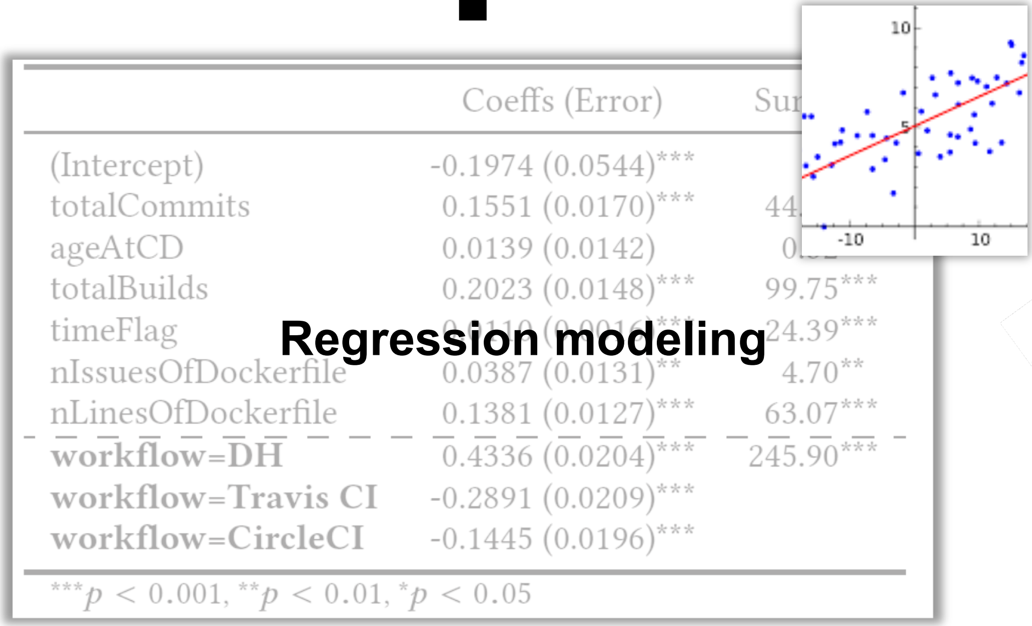
GitHub Continuous Deployment Survey

Welcome!

We are studying how people use continuous deployment technologies in their GitHub projects, and the decisions and tradeoffs that they have to make on which tools to use and how to orchestrate them together. Our goal is to identify pain points and distill best practices that will hopefully help many other open-source projects as well.

Your participation is voluntary and confidential, and is expected to take no more than 10 minutes.

+



	Coeffs (Error)	Su
(Intercept)	-0.1974 (0.0544)***	
totalCommits	0.1551 (0.0170)***	44
ageAtCD	0.0139 (0.0142)	0.
totalBuilds	0.2023 (0.0148)***	99.75***
timeFlag	0.0110 (0.0016)***	24.39***
nIssuesOfDockerfile	0.0387 (0.0131)**	4.70**
nLinesOfDockerfile	0.1381 (0.0127)***	63.07***
workflow=DH	0.4336 (0.0204)***	245.90***
workflow=Travis CI	-0.2891 (0.0209)***	
workflow=CircleCI	-0.1445 (0.0196)***	

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Regression modeling

Hypotheses

Developer survey

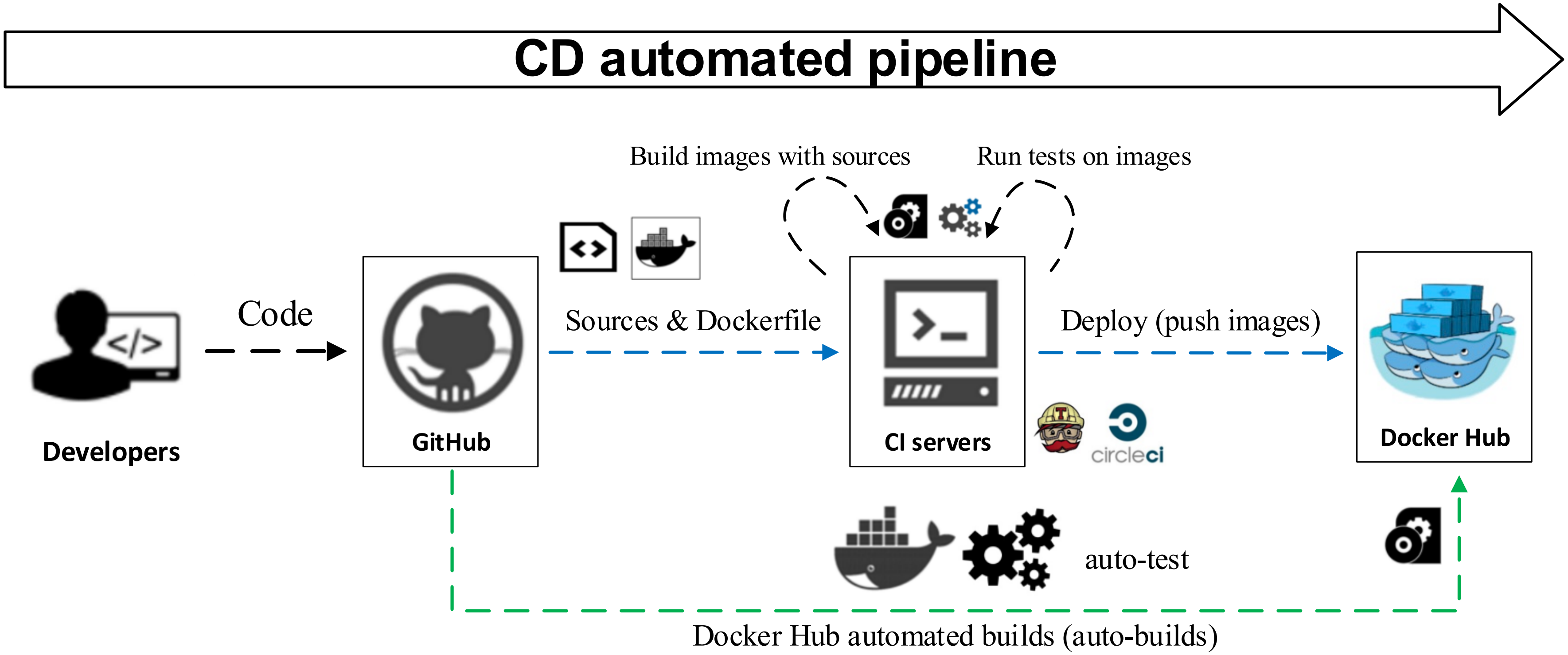
1,000 invitations, 168 responses

Which workflow do developers use in their CD pipeline?

What are the unmet needs in the current CD workflows?

Did developers switch their CD workflows? Why?

Two most prominent CD workflows



34.5%

Travis CI circleci

CI-based Workflow (CIW)

44.1%

Docker Hub auto-builds Workflow (DHW)

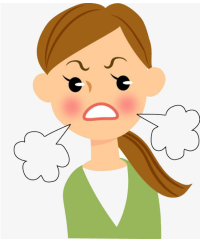
Unmet needs for current workflow

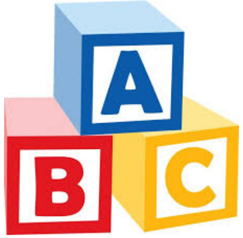
(89.9% of respondents are satisfied)

21.7%

Quicker build speed and higher throughput

“One dockerfile takes more than 2 hours to build and timeouts”



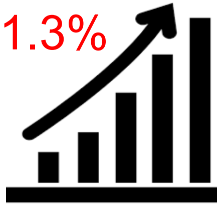


16.9%

Easier to learn and config.

“Sometimes, circle CI config and setup is pain. Docs sometimes doesn’t help”

21.3%



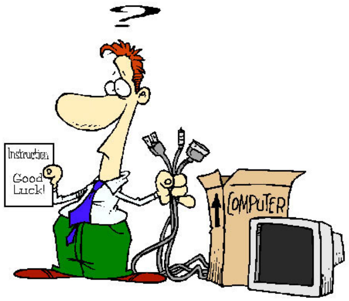
Experienced increasing processing latency over time

Release frequency tends to decrease over time

Image build latency tends to increase over time

Barriers with old CD workflow

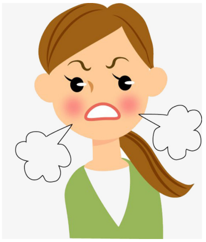
(45.8% of respondents changed)

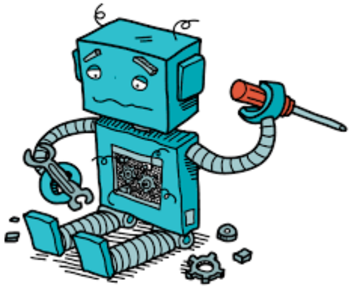


35.2%

Difficult to setup and maintain

"The old CD pipeline is a little harder to setup. It was necessary to write several scripts to get everything working properly. The new CD pipeline is easier to setup and maintain"





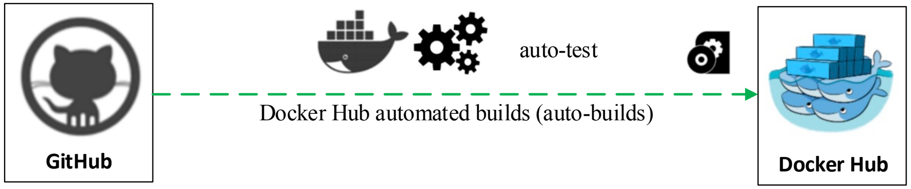
18.3%

Weak support for automation

"Our old workflow contained many manual steps prone to errors, while with the new workflow everything goes smoothly"

Building Hypotheses from Survey

DHW



auto-test

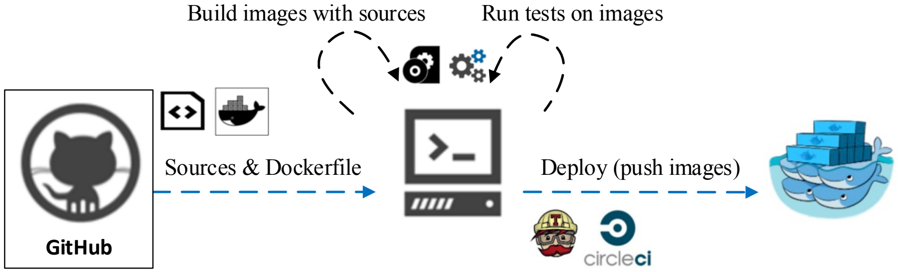
Docker Hub automated builds (auto-builds)

GitHub

Docker Hub

Docker & Docker tools

CIW



Build images with sources

Run tests on images

Sources & Dockerfile

Deploy (push images)

GitHub

circleci

The CI workflow has longer build latency than the DockerHub workflow

The CI workflow has higher release frequency than the DockerHub workflow

Our hypotheses

H1. Release frequency tends to decrease over time.

H2. Build latency tends to increase over time.

H3. Configuration stability tends to increase over time.

H4. CIW tends to have more failed builds than DHW.

H5. CIW tends to have longer build latency than DHW.

H6. CIW tends to have higher release frequency than DHW.

H7. CIW tends to have lower configuration stability than DHW.

H8. Within CIWs, CI tools should not be different.

H1+H6

Release frequency

H4

Build results

H3+H7

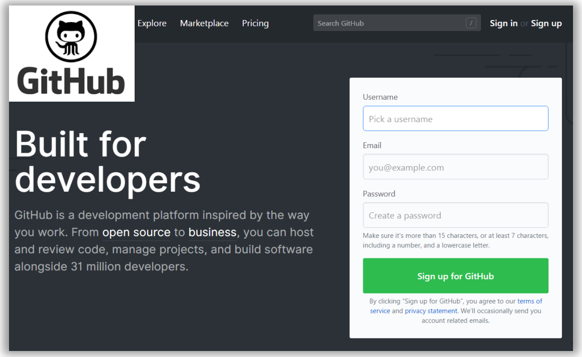
Configuration stability

H2+H5

Build latency

Testing Hypotheses via a Large-scale quantitative study

DECAL Lab



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

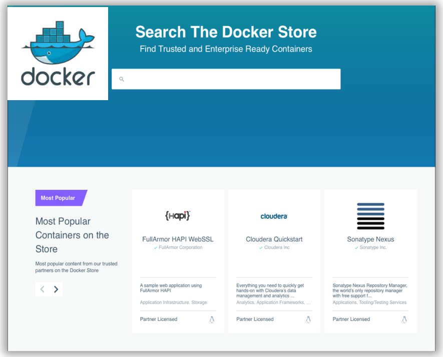
Username: Pick a username

Email: you@example.com

Password: Create a password

Make sure it's more than 15 characters, or at least 7 characters, including a number, and a lowercase letter.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.



Search The Docker Store

Find Trusted and Enterprise Ready Containers

Most Popular Containers on the Store

Most popular content from our trusted partners on the Docker Store

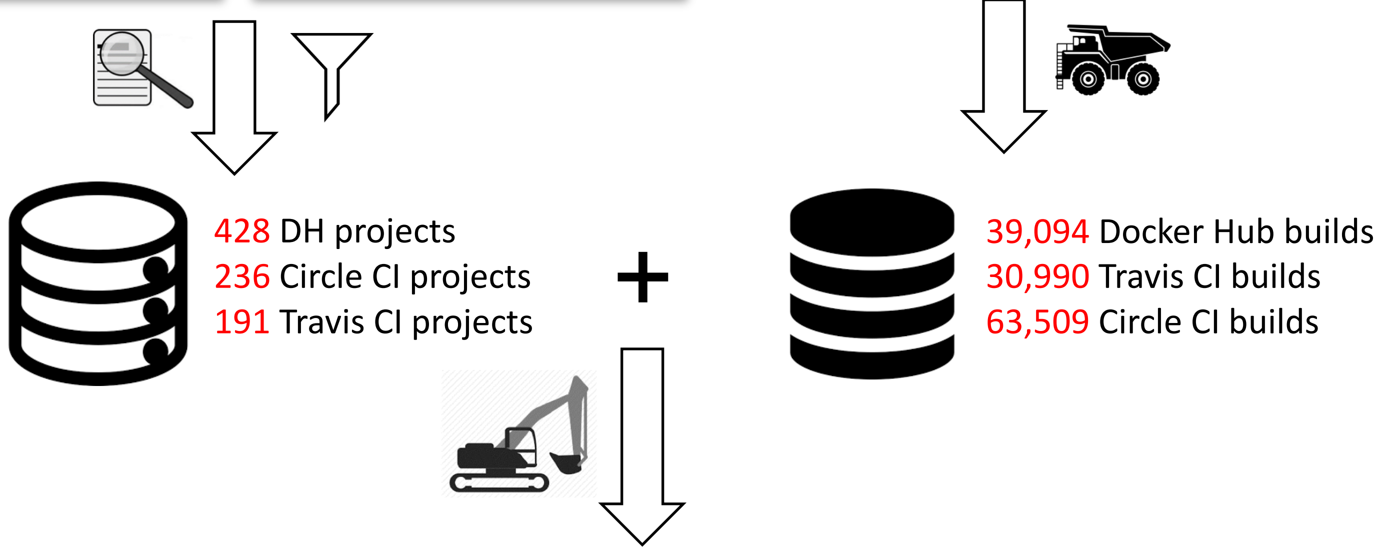
FullArmor HAPI WebSSL
✓ FullArmor Corporation
A sample web application using FullArmor HAPI
Application Infrastructure, Storage
Partner Licensed

Cloudera Quickstart
✓ Cloudera Inc
Everything you need to quickly get hands-on with Cloudera's data management and analytics ...
Analytics, Application Frameworks, ...
Partner Licensed

Sonatype Nexus
✓ Sonatype Inc.
Sonatype Nexus Repository Manager, the world's only repository manager with free support f...
Applications, Tooling/Testing Services
Partner Licensed

circleci

TRAVIS



428 DH projects
236 Circle CI projects
191 Travis CI projects

+

39,094 Docker Hub builds
30,990 Travis CI builds
63,509 Circle CI builds

Release frequency

Build results

Configuration stability

Build latency ~ CD Workflow

Mixed-effects regression models

DECAL Lab

Response

Release frequency: #SuccessBuilds

Build results: #ErrorBuilds

Configuration stability: #DockerfileChanges

Build latency: avg.BuildLatency

Independent

CD Workflow

- DHW
- Travis CIW
- Circle CIW

31 Time

Control

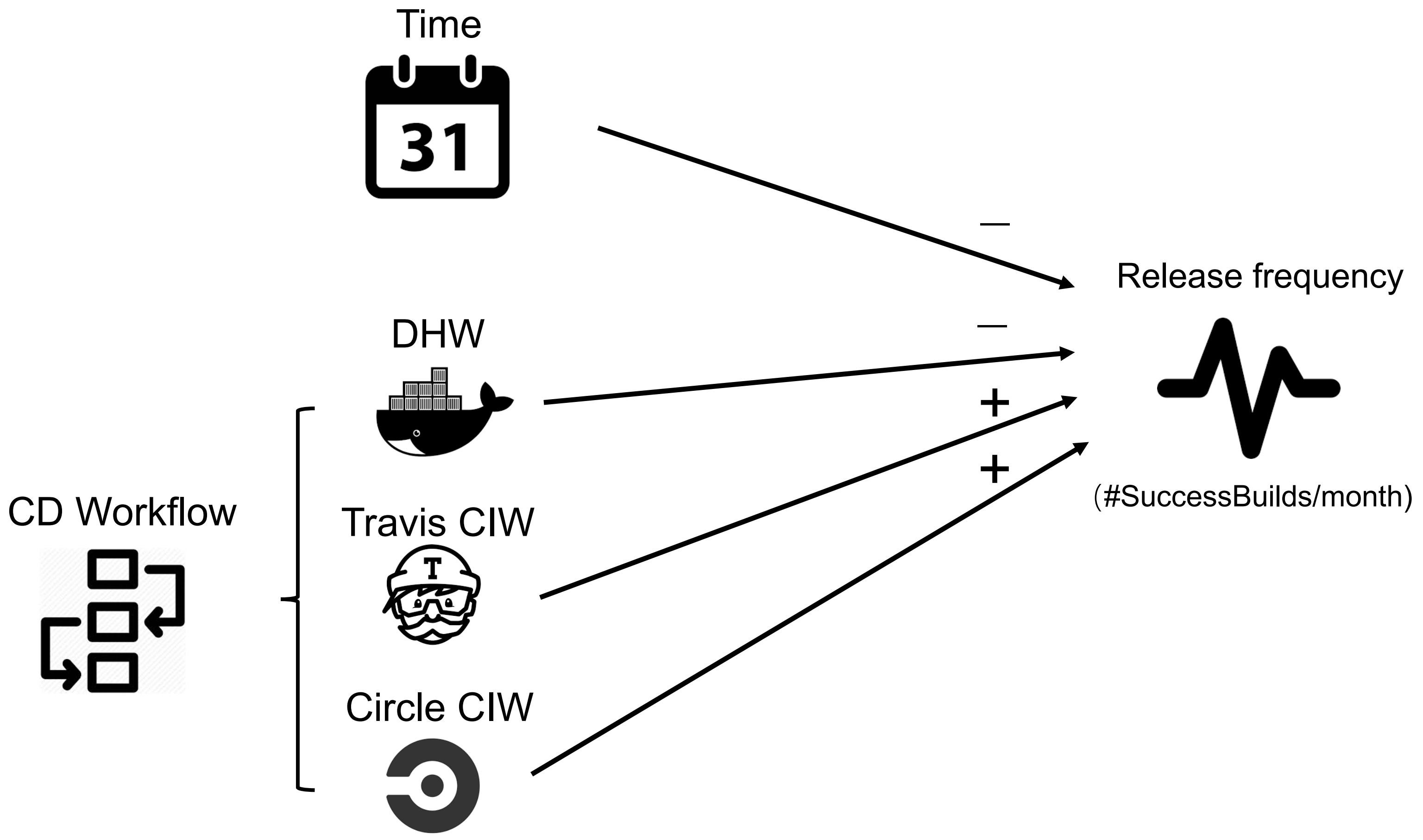
#Builds #Commits

CD Age . . .

+ Import Ubuntu

Base Image

Release frequency & CD workflow



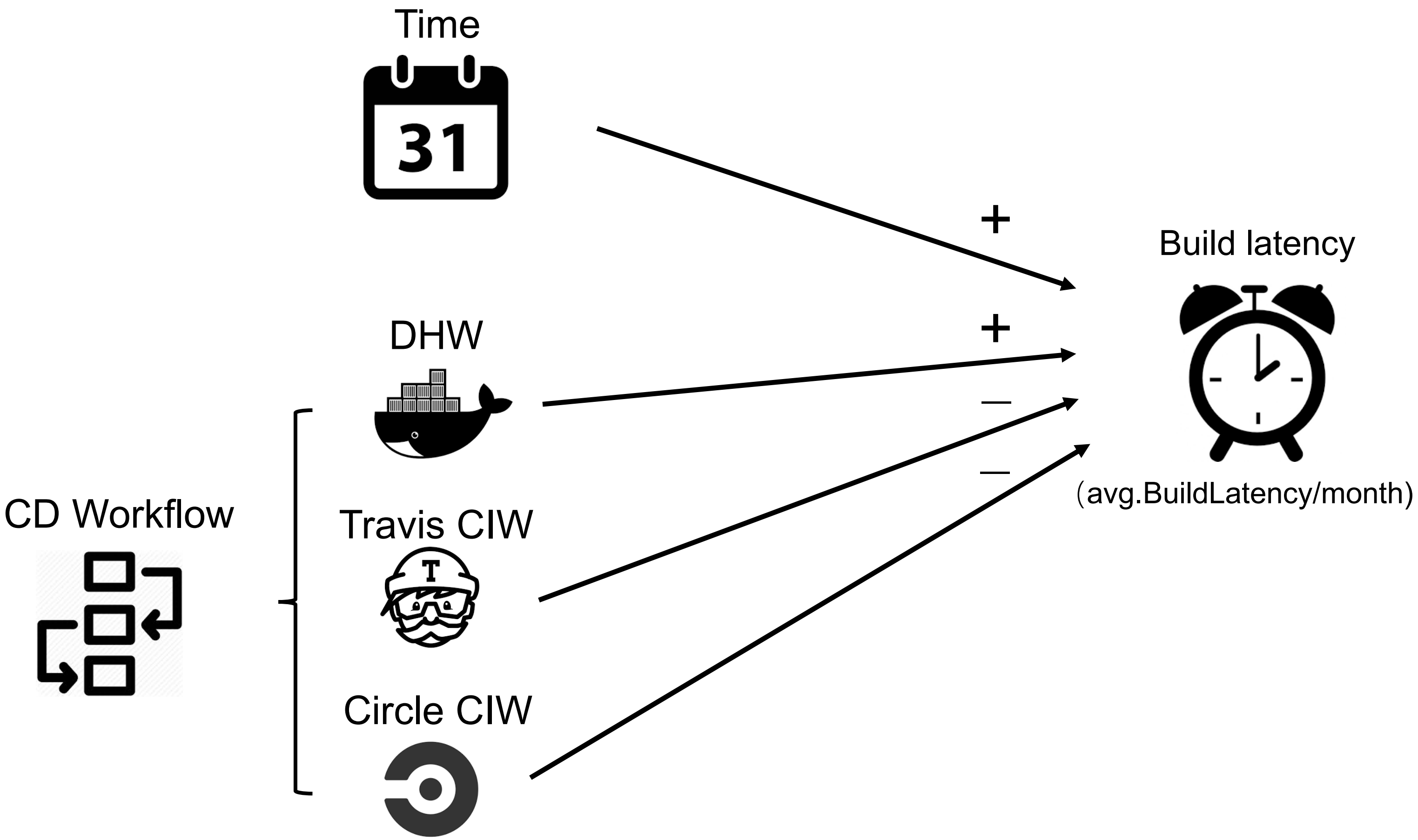
SUPPORTED

H1. Release frequency tends to decrease over time.

SUPPORTED

H6. CIW tends to have higher release frequency than DHW.

Build latency & CD workflows



Time

DHW

Travis CIW

Circle CIW

CD Workflow

+

+

–

–

Build latency

(avg.BuildLatency/month)

SUPPORTED

H2. Build latency tends to increase over time.

NOT SUPPORTED

H5. CIW tends to have longer build latency than DHW.

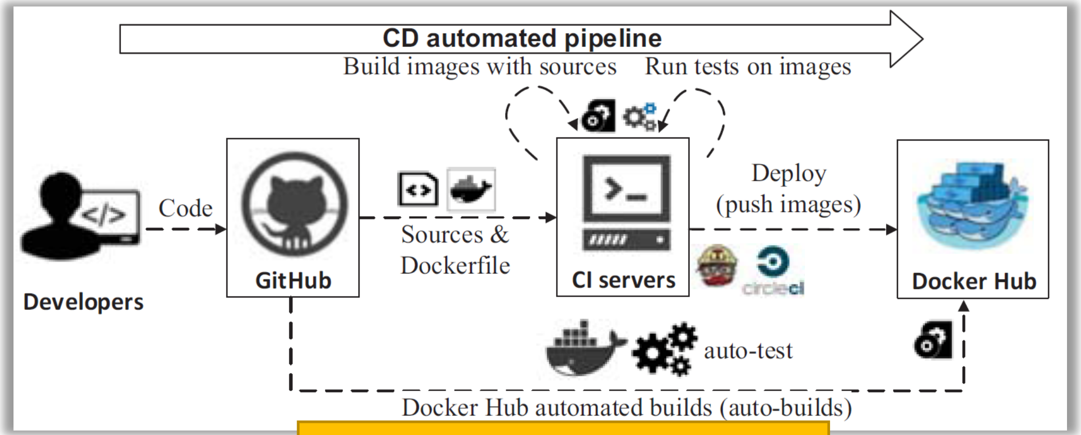
Trade-Offs between CD workflows

DHW	VS	CIW
Higher simplicity		Higher configurability
Diverse needs		Higher performance
Lower maintenance		Higher reliability
Lower experience		Higher scalability

One Size Does Not Fit All:

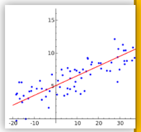
An Empirical Study of Containerized Continuous Deployment Workflows

DECAL Lab



Two CD workflows: DHW and CIW; Unmet needs and barriers;

DHW		CIW
Higher simplicity	Vs	Higher configurability
Diverse needs		Higher performance
Lower maintenance		Higher reliability
Lower experience		Higher scalability



Developers face trade-offs between DHW and CIW

Yang Zhang

Bogdan Vasilescu

Huaimin Wang

Vladimir Filkov

国家自然科学基金委员会
National Natural Science Foundation of China
国防科技大学
National university of defense technology
NSF
STRUDEL
Carnegie Mellon University
UCDAVIS
UNIVERSITY OF CALIFORNIA