Fakultät für Informatik

Studiengang Informatik

DevSecOps mit Azure DevOps Pipelines und GitHub Actions

Projektarbeit im Fachwissenschaftlichen Wahlpflichtmodul DevOps

von

Florian Weidner

Datum der Abgabe: tt.mm.jjjj
Prüfer: Daniel Kerschagl

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

IT-Sicherheit wird in der Software Entwicklung ein immer wichtigeres Thema. In Abbildung 1.1 ist der Anstieg der Ausgaben für IT-Sicherheit zu sehen. Dieser ist in den vergangen Jahren kontinuierlich gestiegen und auch in der Zukunft sollen die Ausgaben im Jahr 2025 im Vergleich zu 2020 um 57.5% steigen. Das zeigt das vielen Unternehmen die IT-Sicherheit immer wichtiger wird. Die dafür anfallenden Kosten sollen aber natürlich trotzdem immer möglichst gering bleiben. DevOps ist mittlerweile eine verbreitete und viel genutzte Entwicklungsmethode (siehe Abb. B.2) um den Entwicklungsprozess zu beschleunigen und somit auch Kosten zu sparen. Um den It-Sicherheitsaspekt einfach in die Softwareentwicklung zu integrieren wurde so DevSecOps erschaffen. Der Prozess vereint DevOps mit Sicherheitsverfahren. [Ibr22] Zwei der größten Plattformen für DevOps sind GitHub und Azure DevOps. GitHub liefert GitHub Actions und Azure DevOps bietet Azure Pipelines um Operationen aus dem DevOps Prozess zu automatisieren. Diese Frage ist welches Tool den DevSecOps Prozess am besten integriert um sichere Software zu erstellen.

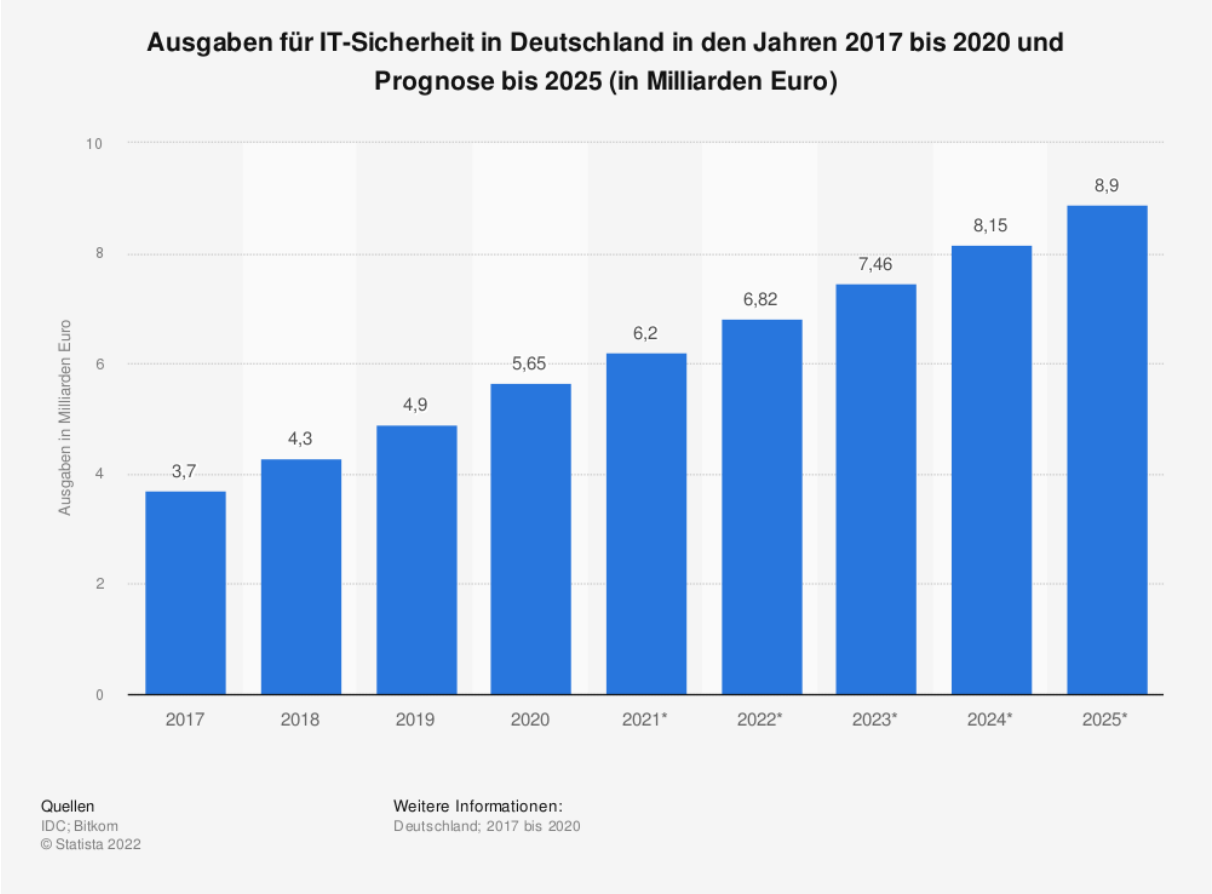

Abbildung 1.1 Ausgaben für IT-Sicheheit in Deutschland in den Jahren 2017 bis 2020 und Prognose bis 2025 (in Milliarden Euro)

2 Was bedeutet DevSecOps

2.1 Zusammenfassung DevOps

DevOps setzt sich aus den Begriffen „Development “(Entwicklung) und „Operations “(Vorgänge) zusammen. Die traditionelle Trennung von Entwicklung und Softwarebetrieb führt oft zu Interessenskonflikten. Entwickler wollen stetig die Software verbessern, der Betrieb hingegen will Änderungen vermeiden um die Stabilität des System zu gewährleisten. [Bee21] Beim Einsetzten von DevOps wird das Entwickeln und den Betrieb von Software näher zusammengeführt. Es entsteht ein Softwareentwicklungs-Prozess, in dem man durch Automatisierungen das Bauen, Testen und Bereitstellen von Software beschleunigen will. Dies erreicht man durch Praktiken wie Continous Integration, Continous Delivery, Continous Deployment, automatisiertes Teste, Infrastructure-as-Code und automatische Veröffentlichungen. Durch die automatische Bereitstellung der Anwendung und der Infrastruktur entsteht eine schnellere Bereitstellung, bessere Softwarequalität und auch schon mehr IT-Sicherheit. [Mac20] Außerdem steht DevOps auch für offene Zusammenarbeit, Kommunikation, Transparenz, Eingestehen von Fehlern und das gemeinsame lösen von Problemen, um Konflikte im Team zu vermeiden. Man will schnelles Feedback ermöglichen und somit das Risiko der Softwareentwicklung minimieren. DevOps beschränkt sich also nicht nur auf technische Hilfsmittel sondern bietet eine Kultur um den Entwicklungsprozess immer weiter zu verbessern.[Bee21]

2.2 Sicherheitsrisiken und Chancen mit DevOps

DevOps alleine bringt einige Herausforderungen zum Thema Softwaresicherheit mit sich. Oft wird Sicherheit für Entwicklungsgeschwindigkeit aufgegeben. Gründe dafür sind, das bei der Umstellung bestehende Sicherheitsmethodiken nicht mir DevOps integrierbar waren oder haben nicht in den Agilen Prozess gepasst. Typischerweise werden dann Sicherheitstests erst am Ende mit der fertig entwickelten Software durchgeführt. Dadurch ist das Sicherheitsteam komplett vom DevOps Prozess ausgeschlossen.

Je schneller die Releaseitegrationen werden, desto weniger Aufwand wird in die Sicherheit gesteckt. Da mit DevOps geänderter Code auch schnell produktiv eingesetzt wird, hat das Entwicklungsteam meistens zu wenig Zeit um die Änderungen sicherheitsrelevant zu Prüfen. Dies bietet dann eventuell neue Angriffsmöglichkeiten oder Risiken in der Software. Ein weiteres Risiko ist, dass das neue Einsetzten von Cloud Computing neue Sicherheitsmaßnahmen fordert, welche zusätzlich berücksichtigt werden müssen.

DevOps liefert aber auch Chancen um IT-Sicherheit in den Entwicklungsprozess zu integrieren. Feste, Zentrale und Standardisierte Bereitstellungspipelines helfen dem Security Team einen besseren Blick über die Anwendung und wie diese erstellt wird zu bekommen. Hier können dann verschieden Sicherheitsaspekte in die Pipeline eingebaut werden. DevOps bietet hier also schon einige Möglichkeiten IT-Sicherheit schon in den DevOps Prozess zu integrieren. [Mao20]

2.3 Definition von DevSecOps

Der Begriff DevSecOps baut auf den Prinzipien und Praktiken von DevOps auf und fügt den Sicherheitsaspekt in der Entwicklung noch weiter in den Vordergrund. Es ist also eine Erweiterung des DevOps Prozesses. Durch DevSecOps soll IT-Security schon vom Start eines Projektes mitgeplant und auch in Bereitstellungsprozess integriert werden. Allerdings reicht auch nicht die Sicherheitsaspekte nur in frühere Phasen der Entwicklung zu verschieben, sondern durchgängige Sicherheit entsteht nur durch ständiges Dazulernen und Verbessern im ganzen Projekt und der Bereitstellung. [Mao20]

Eine technologische Rahmenbedingungen für DevSecOps ist das Bedürfnis für die Automatisierung von Sicherheitsaufgaben im Entwicklungsprozess. Das Einbauen von Sicherheitschecks in die Bereitstellungspipelines reduzieren Zeit und Kosten von Fehlern, die sonst manuell gefunden werden müssen. Außerdem wird die IT-Sicherheit in den Lebenszyklus einer Software integriert und somit können auch Entwickler ohne viel Wissen über IT-Security einfach die automatischen Pipelines nutzen. Das selbe gilt auch für die Bereitstellung der Infrastruktur. Infrastrucutre as Code muss auch eine sichere und zuverlässige Bereitstellung liefern, um hier auch Sicherheitslücken zu vermeiden. [Mao20].

2.4 Praktiken von DevSecOps

DevSecOps muss in verschiedenen Prozessen vorkommen zum Spiel um es auf den ganzen Entwicklungsprozess abzubilden. In der Planungsphase müssen in der Risikoanalyse Sicherheitsaspekte berücksichtigt werden und eine Strategie erstellt werden. Während der Entwicklung helfen statische Code Analyzen und Code Reviews um Probleme zu finden bevor der Code eingecheckt wurde, ohne sich auf die Produktivität der Entwickler auszuwirken. Beim automatischen Erstellen der Software nachdem der Code eingecheckt wurde, sollten automatisierte Tests laufen ob es Fehler beim erstellen gibt. Auch Abhänigkeitsanalyzen und Unit Tests sollen kritische Sicherheitsprobleme aufdecken und den verantwortlichen Entwickler benachrichtigen.

[Ibr22] entwickelte ein Security Model um Sicherheitspraktiken in den kompletten Infrastrukturbereitstellungscode zu bringen. Nachdem ein statischer Analyse Security Scanner den Terraform Code analysiert hat, wird der dieser in einen verschlüsselten Objektspeicher in der Cloud ausgeführt. Hier werden alle benötigten SSH Schlüssel automatisch runter geladen. Die von Terraform erstellten IP-Adressen der Infrastruktur werden an Ansible gesendet welches die erstellten Server dann konfiguriert. Am Ende werden alle Geheimnisse von den Servern gelöscht. So wird eine sichere Bereitstellung der Infrastruktur gewährleistet.

2.5 DevSecOps Kultur

Interessant ist bei DevOps sowie DevSecOps, das es sich um sehr offene Begriffe handelt.

3 DevSecOps mit Github Actions

4 DevSecOps mit Azure DevOps

5 Vergleich von Github Actions und Azure DevOps

6 Fazit

Github ist am Aufstreben blablabla

A Abkürzungsverzeichnis

MAUI Multiplatform App UI

OWIN Open Web Interface

WPF Windows Presentation Foundation

VB6 Visual Basic 6

HTTP Hypertext Transfer Protocol

GUI Graphical User Interface

AVA Ausschreibung Vergabe Abrechnung

HOAI Verordnung über die Honorare für Architekten- und Ingenieurleistungen

BIM Building Information Modeling

B Abbildungsverzeichnis

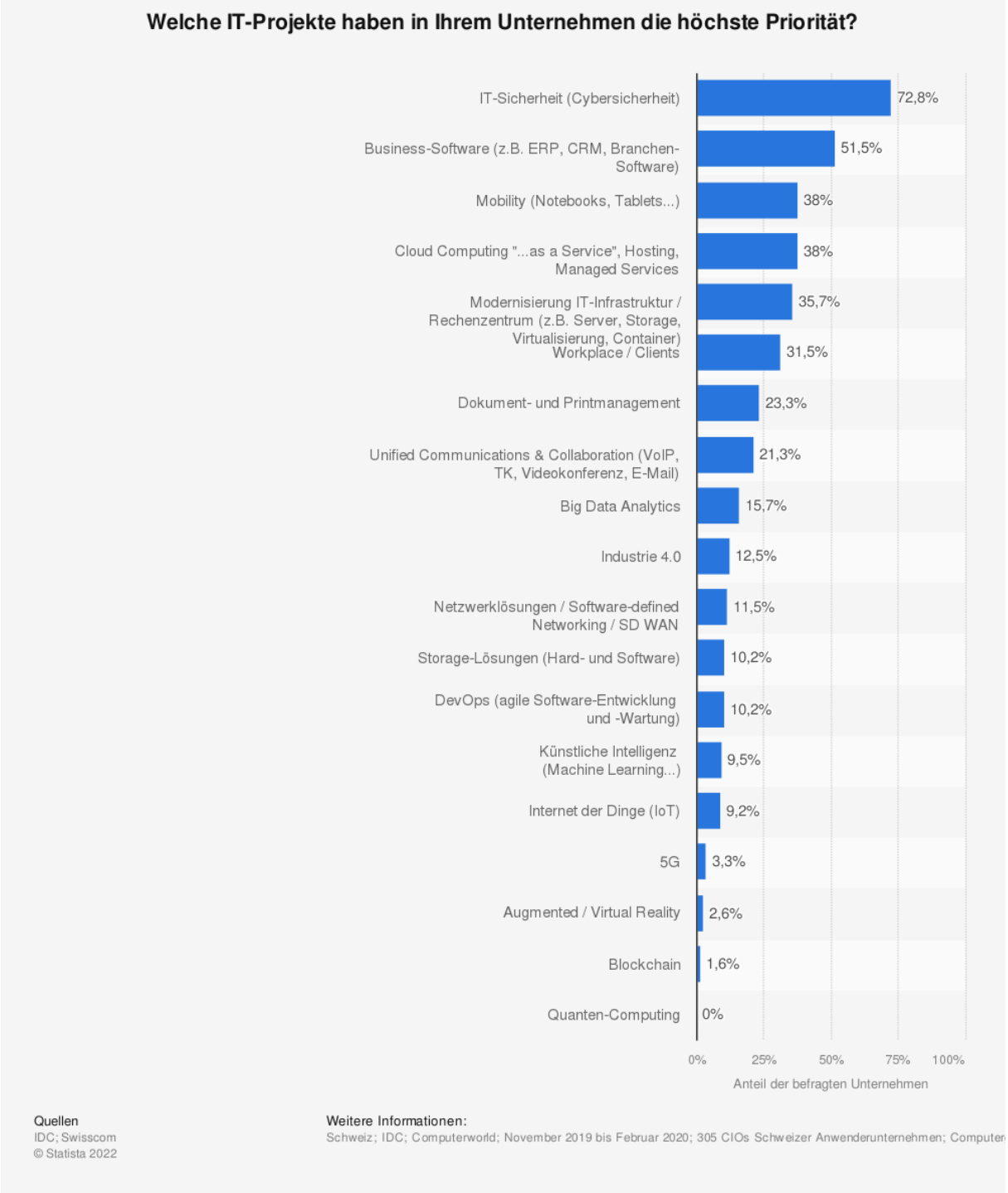

Abbildung B.1 Welche IT-Projekte haben in Ihren Unternehmen die höchste Priorität

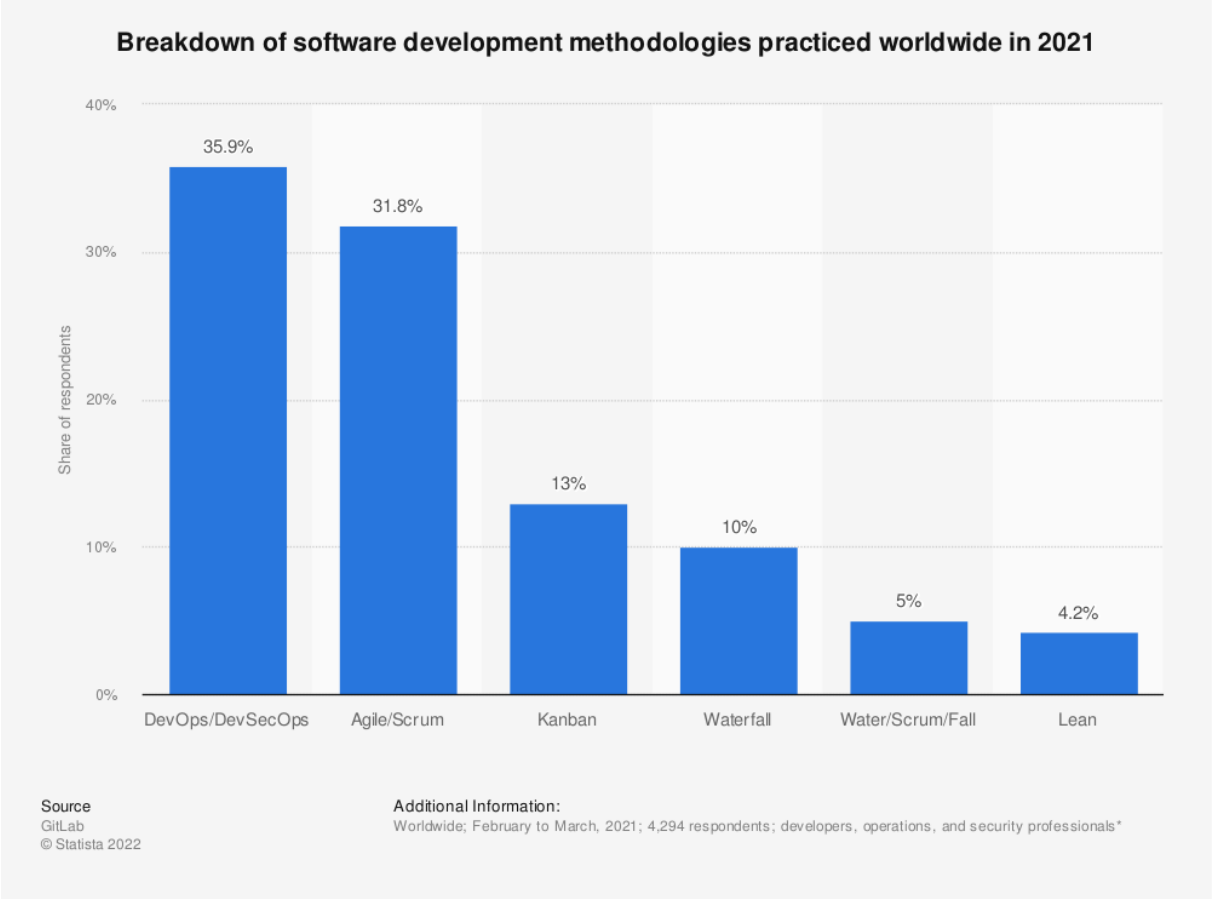

Abbildung B.2 Breakdown of software development methodologies practiced wordlwide in 2021

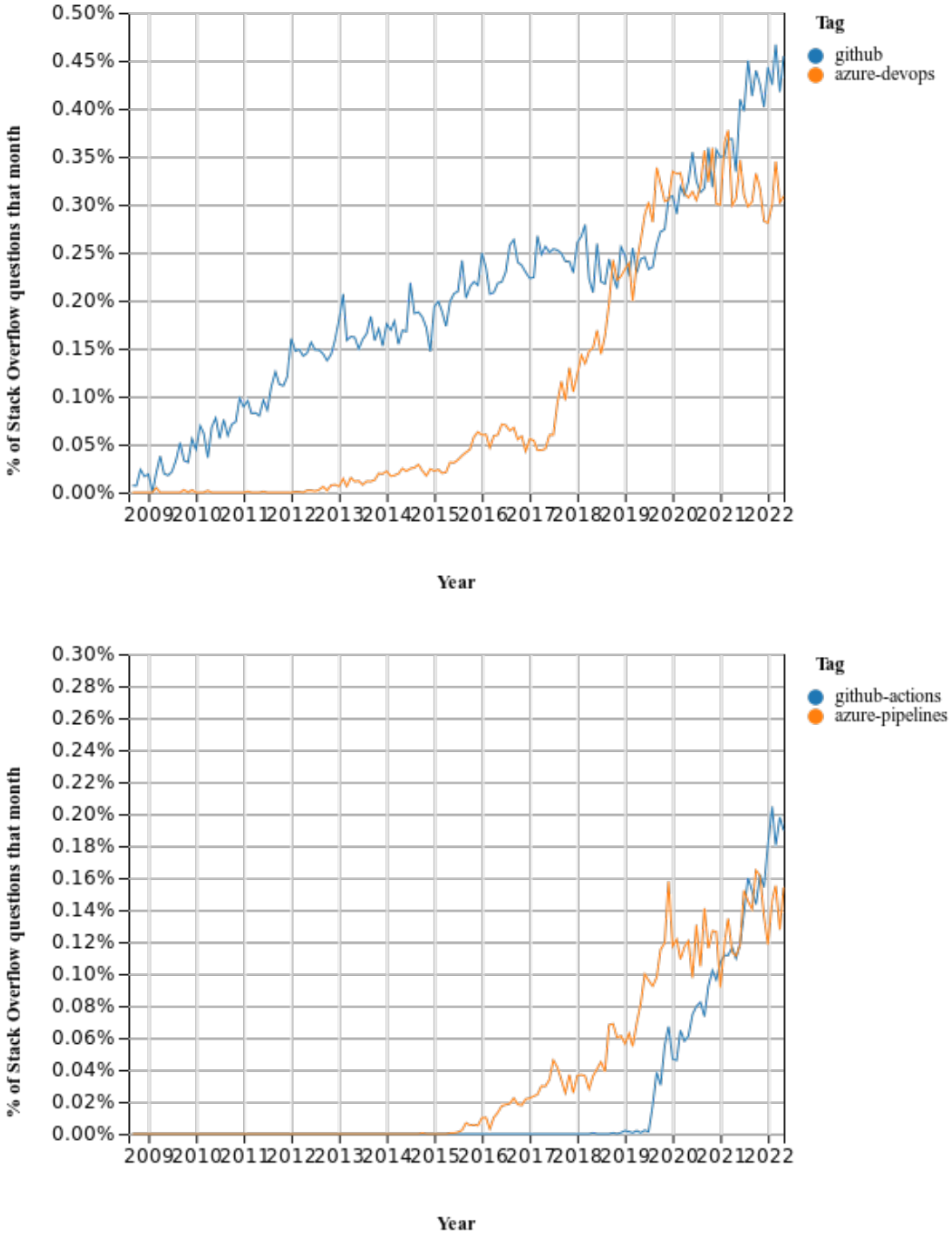

Abbildung B.3 StackOverflow Trends von Github und Azure DevOps vom 18.06.2022

Literaturverzeichnis

[Bee21] F. Beetz und S. Harrer. GitOps: The Evolution of DevOps? *IEEE Software*, S. 0–0, 2021.

[Ibr22] A. Ibrahim, A. H. Yousef und W. Medhat. DevSecOps: A Security Model for Infrastructure as Code Over the Cloud. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, S. 284–288. 2022.

[Mac20] R. W. Macarthy und J. M. Bass. An Empirical Taxonomy of DevOps in Practice. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, S. 221–228. 2020.

[Mao20] R. Mao, H. Zhang, Q. Dai, H. Huang, G. Rong, H. Shen, L. Chen und K. Lu. Preliminary Findings about DevSecOps from Grey Literature. In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, S. 450–457. 2020.

Eigenständigkeitserklärung / Declaration of Originality

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Rosenheim, den tt.mm.jjjj

Vor- und Zuname