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.

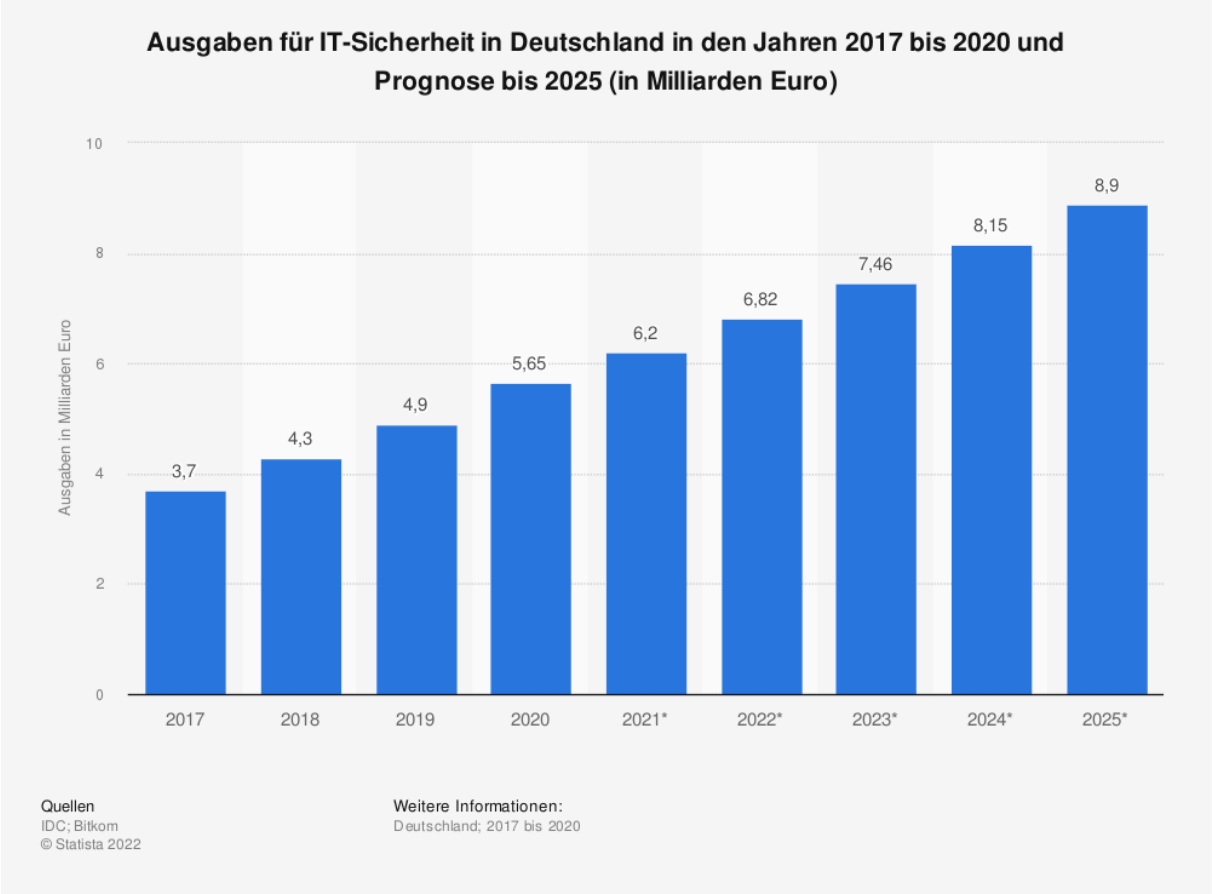

Abbildung 1.1 Ausgaben für IT-Sicheheit in Deutschland in den Jahren 2017 bis 2020 und Prognose bis 2025 (in Milliarden Euro)

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 es nicht, die Sicherheitsaspekte nur in frühere Phasen der Entwicklung zu verschieben. Es soll eine durchgängige Sicherheit durch ständiges Dazulernen und Verbessern des Sicherheitsprozesses im ganzen Projekt und dessen Bereitstellung entstehen. [Mao20] Am Ende ist also das ganze Team für die Applikationssicherheit verantwortlich und Probleme sollen auch im Team angegangen werden. [Ahm19]

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. Zusätzlich wird die IT-Sicherheit in den kompletten 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 auch hier Sicherheitslücken zu vermeiden. [Mao20].

2.4 Praktiken von DevSecOps

DevSecOps muss in verschiedenen Prozessen praktiziert werden, um es auf den ganzen Entwicklungsprozess abzubilden zu können. In der Planungsphase müssen bei der Risikoanalyse Sicherheitsaspekte berücksichtigt werden und eine passende Strategie für das restliche Projekt erstellt zu können. Während der Entwicklung helfen statische Code Analyzen und Code Reviews um Probleme zu finden, bevor der Code eingecheckt wurde. Die Produktivität der Entwickler wird dadurch nur minimal eingeschränkt, es können aber schon erste Fehler erkannt und ausgebessert werden. Nachdem der Code eingecheckt wurde sollten beim Erstellen der Software automatisierte Tests laufen, die feststellen ob es Fehler beim erstellen gibt. Auch Abhänigkeitsanalyzen und Unit Tests sollen kritische Sicherheitsprobleme aufdecken und den verantwortlichen Entwickler benachrichtigen.

Auch die automatisierte Bereitstellung der Infrastrukur ist ein wichtiger Teil, der gesichert sein muss und auch essenzielle Auswirkungen auf das komplette System hat. Hier sind Themen wie Secrets Management, Configuration Management oder Version Control wichtige Aspekte, um die man sich kümmern muss.[Mao20]

[Ibr22] entwickelte ein Security Model um Sicherheitspraktiken in die komplette Infrastrukturbereitstellung 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 Secure Shell (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.

Wenn die Applikation dann auf einer Testumgebung installiert wurde sollen statische und

dynamische Security Tests laufen. Auch automatisierte Attacken sollen Bestandteil des automatischen Testzyklus sein. Im Livebetrieb der Software sollen regelmäßig automatische Sicherheitschecks und Überwachung des Systems stattfinden, um einen Einblick in den Datenverkehr zu bekommen und eventuell bösartige Benutzerverhalten aufdecken zu können. DevSecOps wirkt sich auch auf das Personal aus. Es muss in einem Entwicklungsteam immer „Security Chanpions"[Mao20] geben. Diese sollen besonders Wissen über IT-Sicherheit und dem Entwicklungszyklus haben, um wie auch schon bei DevOps ein funktionsübergreifendes Team zu schaffen. Auch ein regelmäßiges Training der Entwickler und Bereitstellung wichtiger Tools führen dazu das IT-Sicherheit nicht nur nebenbei läuft sondern zu einer Geisteshaltung die das ganze Unternehmen durchdringt. Es gibt also viele verschiedene Praktiken, die man in die Softwareentwicklung integrieren kann und soll.

2.5 DevSecOps Engineer als Beruf

Viele Stoßen beim Einführen von DevSecOps in ihren Entwicklungsprozess auf Herausforderungen. Das sind teilweise hohe Kosten, schon bestehende solide Organisationsstrukturen oder kultureller Widerstand von Entwicklern. Eine Möglichkeit dagegen ist es, einen DevSecOps Spezialisten anzustellen. Dieser kann die Transformation der Arbeitskultur vorantreiben und beschleunigen. Der Experte kann die Einführung verschiedener DevSecOps Tools übernehmen und mit seinem Wissen schneller Sicherheitsintegrationen in zum Beispiel bestehende Pipelines durchführen. [Mao20] Auch DevOps wird oft als Jobbeschreibung verstanden. Als „DevOps Engineer"arbeitet man in verschiedenen DevOps-Teams und -Abteilungen. Diese haben vor allem die Aufgabe genannte Praktiken von DevOps durchzuführen. [Mac20] Ein DevSecOps Engineer setzt auf den Beruf eines DevOps Engineers mit dem Fokus auf IT-Security auf. Er muss ein tiefes Verständnis mitbringen wie sich IT-Security auf die einzelnen Phasen der Entwicklung und das Endprodukt auswirkt. Seine Aufgaben sind es, die automatisierten Sicherheitspraktiken in die Pipeline zu implementieren und immer wieder zu überprüfen und aktuell zu halten. Man sollte also gut in jeden Schritt der Softwareentwicklung integriert sein. Er muss sich natürlich aber auch mit normalen DevOps Prozessen und Prinzipien auskennen. Außerdem sollte er versuchen die restlichen skeptischen Teammitglieder überzeugen, das die Sicherheitspraktiken wichtig sind und die Entwicklungsgeschwindigkeit nicht verlangsamen. Um für einen solchen Job angestellt zu werden helfen verschiedene Zertifikate über DevSecOps Engineering. [Cob19]

3 DevSecOps mit Azure DevOps

4 DevSecOps mit Github Actions

4.1 Zusammenfassung der Sicherheitsfunktionen

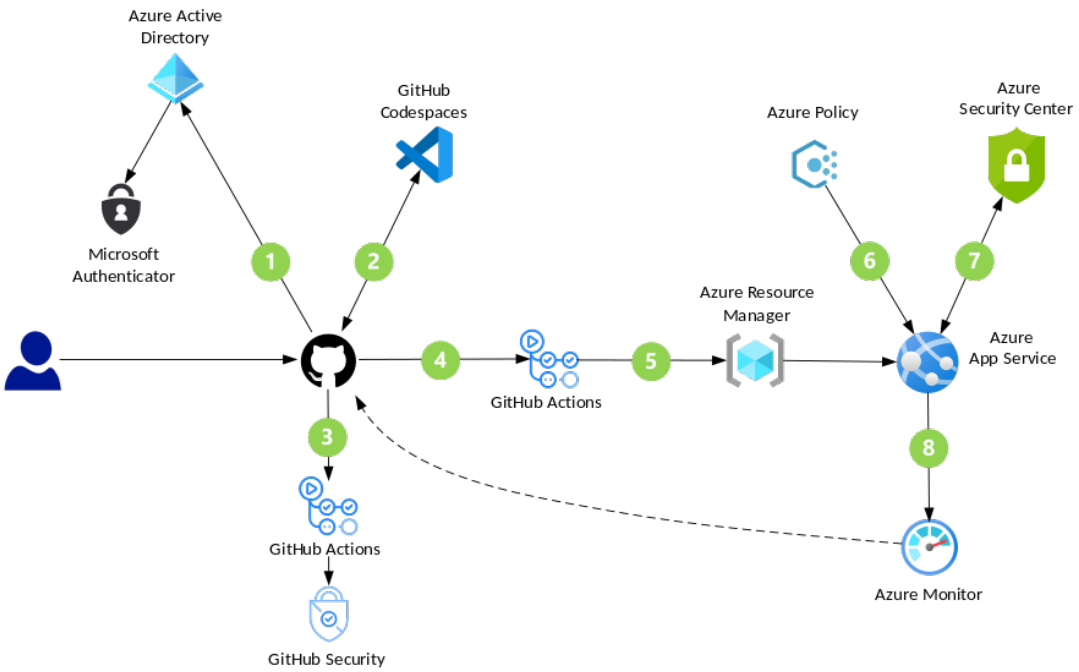

Abbildung 4.1 Architektur von GitHub für DevSecOps Prozess

GitHub bietet mit GitHub Security ein Konzept DevSecOps abzubilden und in den Entwicklungsprozess zu integrieren. Der Aufbau ist in Abbildung 4.1 zu sehen und im folgendem Abschnitt grob beschrieben.

Wenn Entwickler auf GitHub Ressourcen zugreifen, müssen sie eine Authentifizierung über eine Azure Actice Directory durchlaufen. Hier wird eine sichere kennwortlose FIDO2-Authentifizierung verwendet, die nach aktuellen Standards entspricht. [Jus] Mit GitHub Codespaces können Entwickler auf vordefinierte Entwicklungsumgebungen in Containern zugreifen, die mit erforderlichen Erweiterungen für Sicherheitsscans ausgestattet sind. Durch GitHub Actions können automatisch Codierungsfehler und Schwachstellen durch automatische Scans beim Einchecken ermittelt werden. Außerdem können die Bildartefakte direkt in einen Azure App Service deployed werden. Durch Pull Requests können weitere automatisierte Tests ausgeführt werden. Außerdem bietet es eine gute Platform um Code Reviews durchzuführen. Microsoft Defender for Cloud identifiziert Attacken in bereitgestellten Projekten Durch den Azrure Monitor können regelmäßig das Verhalten der laufenden Software überwacht evaluiert und Auffälligkeiten automatisch gemeldet werden.[Mig]

4.2 Codespaces

GitHub Codespaces bietet eine Softwareentwicklungsumgebung, die komplett in der Cloud läuft. Visual Studio Code ist so mit Terminal und Debugger im Browser nutzbar. Neben der hohen Skalierbarkeit und Standardisierung der Entwicklungsumgebung bietet das auch Sicherheitsvorteile. Visual Studio Code bietet hier viele Security Scanning Erweiterungen, die den Entwicklungsprozess sicherer machen. Codespaces bieten Security Logs, in denen festgestellt werden kann wer wann von wo mit der Entwicklungsumgebung gearbeitet hat. [codc] Außerdem kann man sensitive Daten (Secrets) verschlüsselt und sicher in den Umgebungsvariablen der Codespaces speichern. [codb]

4.3 Secret Scanning

Token oder private Schlüssel zur Kommunikation mit externen Services sollten immer geheim gehalten werden. Solche Secrets in einem öffentlichen Repository zu speichern ist demnach fatal für die Sicherheit der Anwendung. Dennoch kann es immer wieder passieren, das Secrets aus versehen eingecheckt werden. GitHub bietet hier ein Secret Scanning über GitHub Actions an. Dieses läuft automatisch bei öffentlichen Repositories und kann mit Github Advanced Security noch mit weiteren Secrets-Übereinstimmungsmustern erweitert werden. [sec]

4.4 Code Scanning

Automatisierte Security Code Analysen sind ein wichtiger Schritt im DevSecOps Prozess um Schwachstellen möglichst früh zu erkennen. GitHub bietet mit Code Scanning eine Möglichkeit über GitHub Actions verschiedene Workflows zu definieren, die Analysen druchführen. Es gibt im Moment 52 schon vordefinierte Workflows zum Thema Security, die zur Verfügung stehen. Neben dem eigenen Code Analyse Tool CodeQL können auch viele weitere externe Tools eingebunden werden.[codd] CodeQL kann Sicherheitsschwachstellen und generische Softwarefehler für die Sprachen C/C++, C#, Go, Java, JavaScript, Python, Ruby, Typescript identifizieren. Man kann mit der Query Language Code wie Daten behandeln und Sicherheitslücken werden als Abfragen modelliert, welche dann bei der Analyse ausgeführt werden können. Hier gibt es schon standardisierte Abfragen von der Github Community, man kann aber auch eigene Abfragen für die Analyse erstellen. [coda]

Alle gefundenen Schwachstellen werden dann im GitHub Repository unter dem Tab Security mit jeweiligen Schweregraden aufgelistet. Man wird direkt zur Codestelle weitergeleitet und die Schwachstelle wird beschrieben und direkt eine Empfehlung abgegeben um das Problem zu lösen.Es werden auch dirket passende Common Weakness Enumeration (CWE) Vulnerabilities mit angegeben um weiter über die Schwachstelle Recherchieren zu können. [codd]

Eine weiterer Workflow der durch GitHub Actions zur Verfügung steht ist eine Dependency Review. Dieser Workflow analysiert alle Abhängigkeiten und möglich folgende Sicherheitsauswirkungen. Man hat also nicht nur einen Überblick wann sich welche Abhängigkeiten ändern sondern bekommt auch direkt Informationen ob die genutzten Pakete eventuelle Sicherheitsschwachstellen haben. Für alle öffentlichen Repositories wird automatisch bei jedem Pull Request ein Dependecy Graph erstellt, der alle Abhängigkeiten mit kompatiblen Packagemana-

gement Systemen beinhaltet. Zusätzlich gibt es den GitHub Dependabot. Dieser Analyseservice läuft in jedem öffentlichen Repository wenn man die Funktion aktiviert hat. Er erkennt alle Codeabhängigkeiten mit Sicherheitsschwachstellen. Es werden alle hinzugefügten, geänderten oder gelöschten Abhänigkeiten in einem Pull Request analysiert und dann im Security Tab des GitHub Repositories als Alert bereitgestellt.

4.5 Github Security Lab

Mit dem GitHub Security Lab bietet die Plattform ein Portal, auf dem sie versuchen User zu inspirieren und es ihnen ermöglichen leichter sichere Open Source Software zu implementieren. Auf der Plattform können Schwachstellen von großen, wichtigen Open Source Projekten berichtet werden. Das Portal koordiniert dann die Offenlegung der Schwachstellen mit den Sicherheitsteam der betroffenen Projekte. Es werden zusätzlich verschiedene Analysen mit CodeQL, einer semantischen Code-Analyse-Engine durchgeführt um die Sicherheitsforschung für Open Source Projekte zu vergrößern. Zusätzlich werden Common Vulnerabilities und Exposures in einer Datenbank gesammelt. Außerdem werden viele Forschungen, Tutorials, Artikel oder Konferenzen veröffentlicht und der Community bereitgestellt. Es werden auch mit dem so genannten „CodeQL Bug Bounty program“immer wieder Prämien ausgeschrieben, um Sicherheitsforschung in der Community zu verstärken. Die Plattform bietet also über die technischen Funktionen von GitHub hinaus, die Möglichkeit für Entwickler sich über IT-Sicherheit zu vernetzten, zu lernen oder sich an der Sicherheitsforschung für Open Source Projekte zu beteiligen. [git]

5 Vergleich von Github Actions und Azure DevOps

5.1 Kosten

GitHub bietet viel kostenlose für Open Source Repositories an. Um aber den kompletten DevSecOps Prozess abzudecken ist GitHub Enterprise nötig.

6 Fazit

Github ist am Aufstreben blablabla

A Abkürzungsverzeichnis

SSH Secure Shell

CWE Common Weakness Enumeration

B Abbildungsverzeichnis

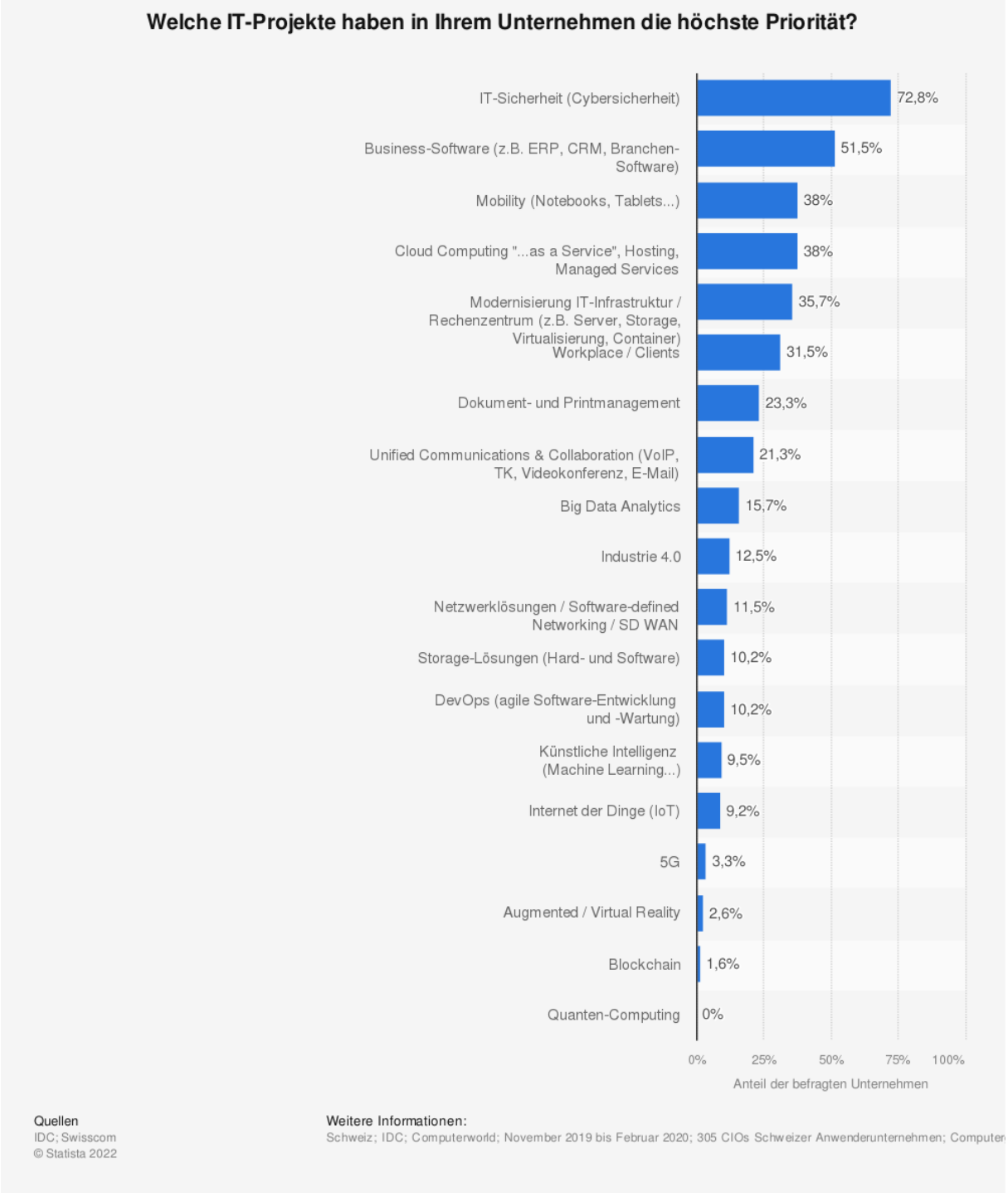

Abbildung B.1 Welche IT-Projekte haben in Ihren Unternehmen die höchste Priorität

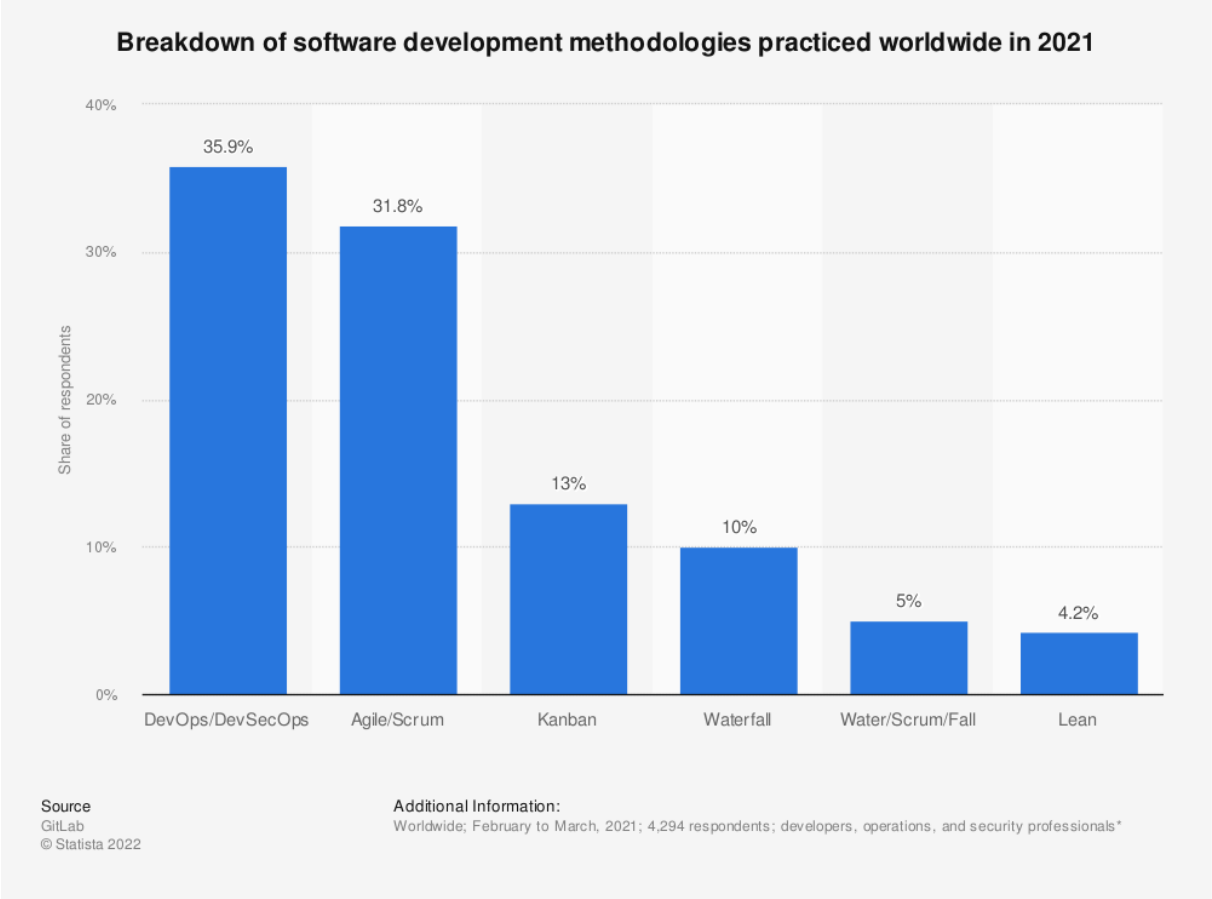

Abbildung B.2 Breakdown of software development methodologies practiced wordlwide in 2021

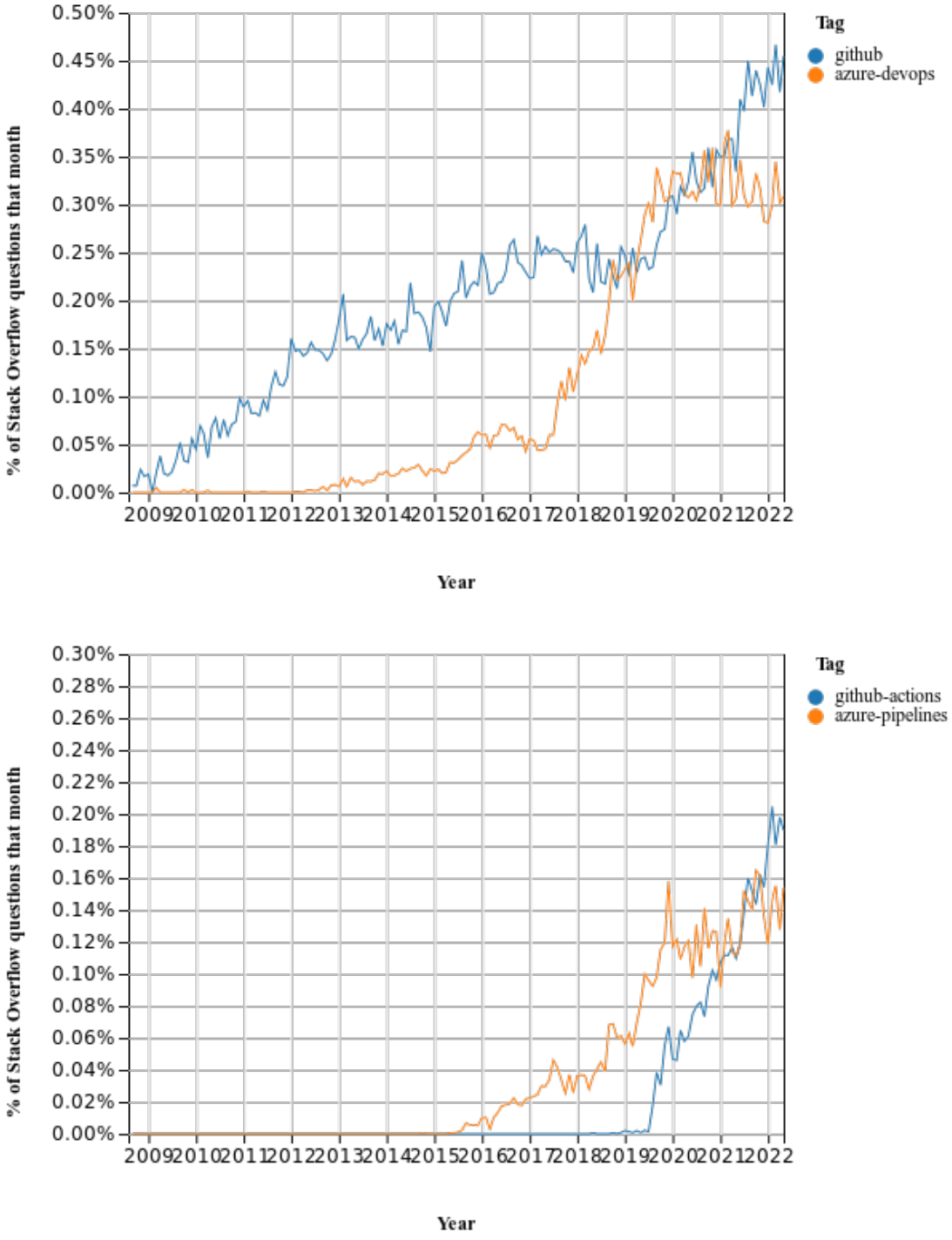

Abbildung B.3 StackOverflow Trends von Github und Azure DevOps vom 18.06.2022

Literaturverzeichnis

[Ahm19] Z. Ahmed und S. C. Francis. Integrating Security with DevSecOps: Techniques and Challenges. In *2019 International Conference on Digitization (ICD)*, S. 178–182. 2019.

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

[Cob19] M. Cobb. What it takes to be a DevSecOps engineer. `https://www.techtarget.com/searchsecurity/tip/What-it-takes-to-be-a-DevSecOps-engineer`, Sept. 2019. Accessed: 2022-6-28.

[coda] About CodeQL. `https://codeql.github.com/docs/codeql-overview/about-codeql/`. Accessed: 2022-6-28.

[codb] Managing encrypted secrets for your codespaces. `https://docs.github.com/en/codespaces/managing-your-codespaces/managing-encrypted-secrets-for-your-codespaces`. Accessed: 2022-6-28.

[codc] Reviewing your security logs for Codespaces. `https://docs.github.com/en/codespaces/managing-your-codespaces/reviewing-your-security-logs-for-codespaces`. Accessed: 2022-6-28.

[codd] Setting up code scanning for a repository. `https://docs.github.com/en/code-security/code-scanning/automatically-scanning-your-code-for-vulnerabilities-and-errors/setting-up-code-scanning-for-a-repository`. Accessed: 2022-6-28.

[git] GitHub security lab. `https://securitylab.github.com/`. Accessed: 2022-6-28.

[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.

[Jus] Justinha. Kennwortlose Azure Active Directory-Anmeldung - Microsoft Entra. `https://docs.microsoft.com/de-DE/azure/active-directory/authentication/concept-authentication-passwordless`. Accessed: 2022-6-28.

[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.

[Mig] F. Migacz. DevSecOps mit GitHub Security. `https://docs.microsoft.com/de-de/azure/architecture/solution-ideas/articles/devsecops-in-github`. Accessed: 2022-6-28.

[sec] About Secret Scanning. `https://docs.github.com/en/code-security/secret-scanning/about-secret-scanning`. Accessed: 2022-6-28.

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