Die AppSecure.nrw Software Security Studie

Eine Umfrage unter Entwickler*innen, Product Ownern und Führungskräften

Inhaltsverzeichnis

VORWORT

> „Es tut sich was, es gewinnt an Wichtigkeit, wird aber sicherlich nicht von allen mit Kusshand angenommen, weil Security immer noch bremsend wirkt. [...] Ich glaube es würde keiner sagen Security ist egal, ganz sicher nicht! Aber es ist anstrengend, sich mit dem Thema zu beschäftigen.“ – Interviewteilnehmer*in

Security gewinnt in allen Branchen immer mehr an Bedeutung, da die entwickelten Systeme zunehmend schützenswerte Daten verarbeiten und kritische Dienste bereitstellen. Daher ist es nicht verwunderlich, dass das World Economic Forum in seinem Global Risk Report 2020 das Thema Security als das größte technologische Risiko für die Weltbevölkerung einstuft. Dies wirft die Frage auf, inwieweit deutsche Unternehmen das Thema Security bei der Entwicklung und dem Betrieb ihrer Softwareprodukte adressieren und was aktuelle Herausforderungen sind. Zur Beantwortung dieser Fragen haben wir in unserem Forschungsprojekt AppSecure.nrw (www.appsecure.nrw) im Jahr 2019 eine umfangreiche Studie durchgeführt. Dabei haben wir nicht nur die Entwickler*innen der Software, sondern auch deren Führungskräfte (FK) und Product Owner (PO) befragt, um ein ganzheitliches Bild zu erhalten.

Das Ergebnis unserer Studie ist, dass für Unternehmen die Gewährleistung von Security in ihren Produkten eine vielschichtige Herausforderung ist und Handlungsbedarf besteht. Primär fehlt es den Entwickler*innen, FK und PO an der Sensibilisierung für das Thema Angriffssicherheit, an Security-Kompetenz sowie an Methoden zur sicheren Softwareentwicklung und hierfür passenden Werkzeugen. Darüber hinaus haben wir erkannt, dass die PO zu wenige oder keine Security-Anforderungen an das zu entwickelnde Produkt stellen und FK nur selten Maßnahmen zum Ausbau der Security-Kompetenz durchführen. Somit besteht für viele Unternehmen die Gefahr, dass ihre Produkte nicht vor böswilligen Angriffen geschützt sind. Erfreulicherweise sind sich viele Entwickler*innen, FK und PO ihren aktuellen Problemen und Herausforderungen bewusst und bereit, die heutige Situation zu verbessern. Aufgrund der Ergebnisse unterstützen wir im weiteren Projektverlauf von AppSecure.nrw dieses Vorhaben durch die Definition eines Reifegradmodells für agile Teams, die Erstellung von Weiterbildungen für Entwickler*innen, FK und PO sowie die Weiterentwicklung von bestehenden freien Werkzeugen.

Im Folgenden stellen wir Ihnen unsere Studienergebnisse im Detail vor. Nach der Erläuterung unserer Methodik und dem beruflichen Hintergrund der Studienteilnehmer*innen werden die zentralen Ergebnisse unserer Studie anhand von fünf Themen dargestellt: Zunächst fokussieren wir den Entwicklungsprozess, welchen wir in die Bereiche Anforderungsmanagement, Entwurf, Implementierung & Tests sowie den Betrieb unterteilt haben. Anschließend berichten wir über unsere Erkenntnisse zum Werkzeugeinsatz. Danach thematisieren wir die Security-Kompetenz aller Beteiligten sowie die heutigen Maßnahmen zum Kompetenzausbau. Im letzten Thema analysieren wir die Sensibilisierung für Security. Am Ende des Dokuments finden Sie das Fazit und eine Skizzierung von dessen Auswirkungen sowie unsere sich hieraus ableitenden Empfehlungen.

Wir wünschen Ihnen eine anregende Lektüre. Sollten Sie Fragen zur Studie oder unseren Empfehlungen haben, sprechen Sie uns gerne an.


AppSecure.nrw

1

METHODIK

Im Rahmen der Studienerstellung wurden sowohl qualitative als auch quantitative Forschungsmethoden eingesetzt. Aus insgesamt 21 Forschungsfragen wurden eine Online-Umfrage und zwei Interview-Leitfäden entwickelt. Diese wurden intern und anschließend mit den AppSecure.nrw Projektpartnern AXA Konzern AG, Connext Communication GmbH und adesso mobile solutions GmbH ausführlich bzgl. Verständlichkeit und Eindeutigkeit verbessert. Verantwortlich für die Entwicklung, Durchführung und Auswertung der Befragungen ist das Fraunhofer IEM.

Die 40 Fragen umfassende Online-Umfrage wurde anonym durchgeführt und durch Fraunhofer IEM und dessen Geschäftspartner, den Projektpartnern von AppSecure.nrw, den Technologie-Netzwerken it's OWL und innozent OWL, Heise Medien und der Bitkom beworben. Insgesamt haben 350 Personen an der Umfrage teilgenommen. Die Antworten von Teilnehmenden aus der Schweiz und Österreich mussten wir leider herausnehmen, da diese in zu geringer Anzahl teilgenommen hatten, um eine aussagekräftige Auswertung dieser Länder vorzunehmen. Darüber hinaus haben wir alle Personen herausgefiltert, die die Umfrage nicht vollständig ausgefüllt haben bzw. deren Bearbeitungszeit kein Lesen der Fragen zulässt. Insgesamt verblieben somit die Antworten von 256 Personen aus Deutschland. Diese Auswahl benötigte eine durchschnittliche Bearbeitungszeit von 25 Minuten zur Beantwortung aller Fragen. Bezüglich unserer Ergebnispräsentation sei noch ergänzt, dass wir alle Prozentzahlen auf ganze Zahlen gerundet haben. Daher sind alle Abweichungen der Summe von 100 Prozent rundungsbedingt.

Vier der insgesamt 17 interviewten Personen sind Teil des Projektteams von AppSecure.nrw. Die restlichen 13 Personen hatten keinen Bezug zum Projekt. Die Interviewleitfäden für Führungskräfte (FK) und Product Owner (PO) bestanden aus jeweils 17 Fragen. Hatte ein Interviewpartner beide Rollen inne, wurden die Fragen aus beiden Leitfäden gestellt. Basierend auf den Leitfäden wurden in den Interviews offene Fragen gestellt, um möglichst viele Zwischentöne zu erfassen. Auch Nachfragen und Ausschweifungen wurden bewusst zugelassen. Ein Interview dauerte im Schnitt etwa 45 Minuten. Bzgl. der Auswertung sei noch erwähnt, dass sich in den Interviews große Überschneidungen zwischen den Rollen FK und PO gezeigt haben. Daher haben wir in unserer Auswertung oftmals beide Rollen gemeinsam analysiert – falls es Unterschiede gab, haben wir diese jedoch stets kenntlich gemacht. Alle Zitate, die wir in diesem Dokument darstellen, stammen aus diesen Interviews.

2

BERUFLICHER HINTERGRUND DER TEILNEHMER*INNEN

2.1 Entwickler*innen

Die Online-Umfrage wurde von insgesamt 256 Entwickler*innen aus Deutschland vollständig ausgefüllt. 61% der Befragten geben an, über mehr als zehn Jahre Berufserfahrung in der Softwareentwicklung (vgl. Abbildung 1.a) zu verfügen. 18% der Befragten haben zwischen sechs und zehn Jahre Berufserfahrung und 15% der Befragten haben zwischen zwei und fünf Jahre Berufserfahrung. 6% der Befragten sind Berufseinsteiger*innen und verfügen über weniger als zwei Jahre Berufserfahrung. Insgesamt haben wir mit unserer Online-Umfrage überwiegend sehr erfahrene Entwickler*innen erreicht.

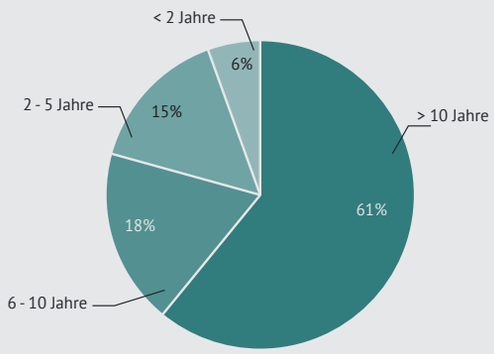
Gegliedert nach Unternehmensgröße stammen 14% der Befragten aus Unternehmen mit maximal 50 Beschäftigten und 26% aus Unternehmen mit mindestens 51 und maximal 250 Beschäftigten. Insgesamt sind damit 40% der Befragten bei sogenannten kleinen und mittleren Unternehmen (KMU) beschäftigt (vgl. Abbildung 1.b). 61% der Befragten arbeiten in großen Unternehmen (mehr als 250 Beschäftigte).

Die Geschäftsmodelle der Unternehmen sind sehr unterschiedlich (vgl. Abbildung 1.c). In den meisten Fällen wird die Software (55%) im eigenen Unternehmen eingesetzt. Zudem wird die Software an Kunden lizenziert (36%) oder im direkten Kundenauftrag entwickelt (28%). 15% der Entwickler*innen werden zudem zu anderen Unternehmen entsendet, um dort in Projekten mitzuwirken.

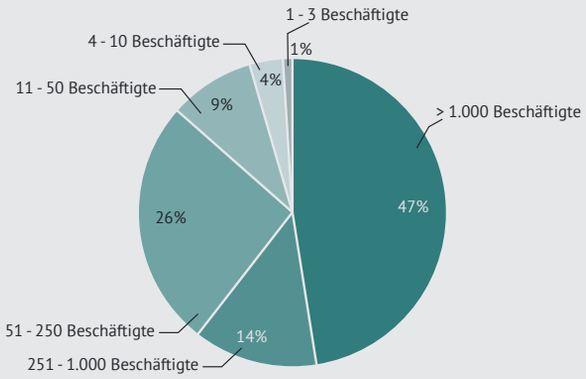
Die Entwickler*innen sind an verschiedensten Applikationen beteiligt (vgl. Abbildung 1.d). Besonders häufig werden Webanwendungen (66%) und Backend-Applikationen (59%) entwickelt, gefolgt von Desktop-Applikationen (37%) und mobilen Applikationen (25%). Eher selten werden Applikationen für das Embedded-Umfeld entwickelt (14%).

Die Größe der Teams ist bei den befragten Entwickler*innen meist 15 oder weniger Personen (vgl. Abbildung 1.e). Innerhalb des jeweiligen Projekts arbeiten die befragten Entwickler*innen üblicherweise in Teams mit 6-15 Personen (56%) oder 1-5 Personen (32%). Größere Teams mit mehr als 16 Personen sind eher selten (13%).

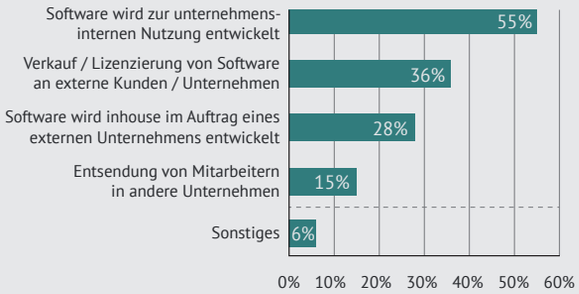
Die Entwickler*innen arbeiten typischerweise in mehreren Disziplinen: 64% sind im Anforderungsmanagement aktiv, 81% im Entwurf, 86% in Implementierung und/oder Softwaretest und 52% im Betrieb der Software tätig (vgl. Abbildung 1.f).



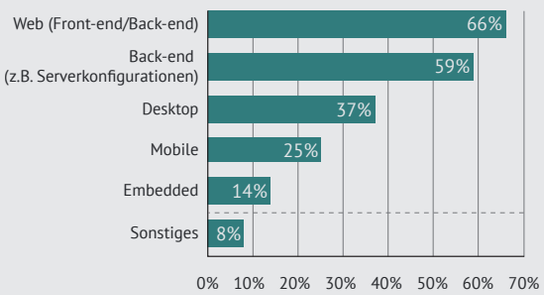
a) Frage: „Seit wie vielen Jahren sind Sie in der Softwareentwicklung tätig?“
N=256.



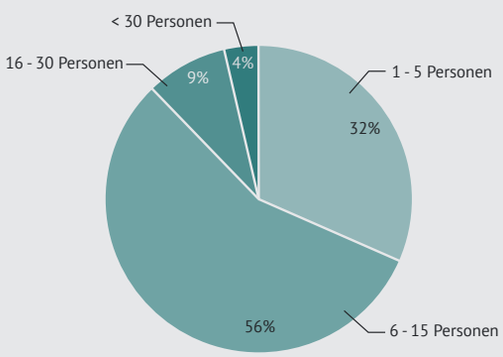
b) Frage: „Wie viele Beschäftigte hat das Unternehmen?“
N=256.



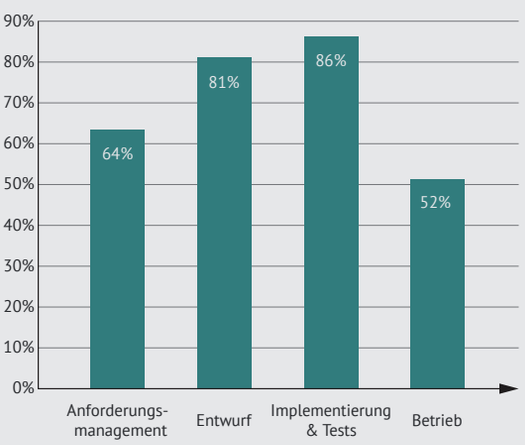
c) Frage: „Was ist das Geschäftsmodell des Unternehmens bezüglich der Softwareentwicklung?“ Mehrfachantwort möglich.
N=256.



d) Frage: „Bei welchen Applikationen sind Sie an der Entwicklung beteiligt?“ Mehrfachantwort möglich.
N=256.



e) Frage: „Wie groß ist das Team, in dem Sie (typischerweise) tätig sind?“
N=256.



f) Die Entwickler*innen wurden pro Disziplin gefragt: „Sind Sie in dieser Disziplin tätig?“ (Ja oder Nein). Im Diagramm ist jeweils der Anteil der Ja-Stimmen abgebildet.
N=256.

*Abbildung 1: Überblick über den beruflichen Hintergrund der befragten Entwickler*innen*

Die Entwickler*innen unserer Studie nutzen ein breites Spektrum an Entwicklungsumgebungen und Programmiersprachen (vgl. Abbildung 2). Besonders häufig werden IntelliJ und Eclipse eingesetzt. Darüber hinaus sind die Entwicklungsumgebungen Visual Studio und Visual Studio Code ebenfalls weit verbreitet. Bei den Programmiersprachen liegen die Sprachen Java, JavaScript/TypeScript und C# auf den vorderen Plätzen (vgl. Abbildung 3). Durchschnittlich nutzen die Entwickler*innen zwei bis drei Programmiersprachen und zwei unterschiedliche Entwicklungsumgebungen.

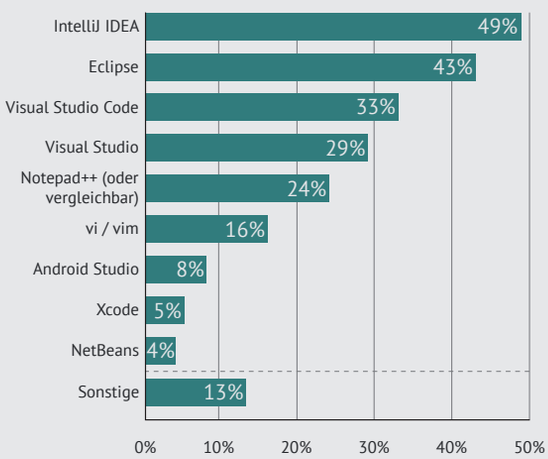


Abbildung 2: Überblick über die verwendeten Entwicklungsumgebungen

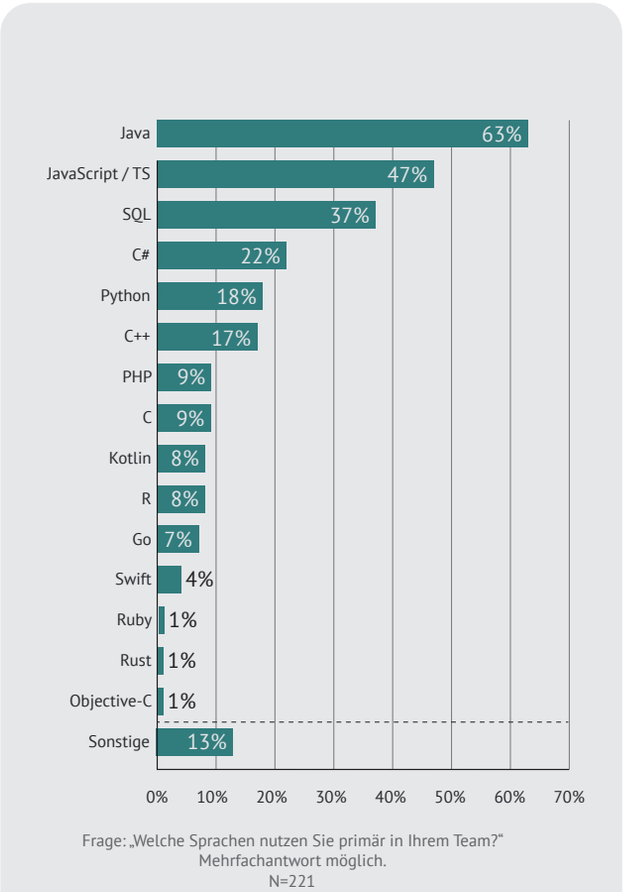


Abbildung 3: Überblick über die verwendeten Programmiersprachen

2.2 Führungskräfte und Product Owner

Im Rahmen der Interviews haben wir 17 Personen anhand eines Leitfadens interviewt. Davon sind sieben ausschließlich als Product Owner (PO) und sechs ausschließlich als Führungskräfte (FK) tätig. Die restlichen vier Personen haben beide Rollen inne. Alle interviewten Personen verfügen über mehr- oder langjährige Berufserfahrung und haben stets direkten Kontakt zur Softwareentwicklung.

Ähnlich zu den Befragten der Online-Umfrage stammen die befragten FK und PO sowohl aus kleinen (2 Personen) und mittleren Unternehmen (10 Personen), sowie aus Unternehmen mit mehr als 250 Mitarbeitenden (5 Personen). Die Unternehmen, für die die befragten FK und PO tätig sind, entwickeln für verschiedene Branchen Software (z.B. Automobil, Gesundheit und Versicherung). Darüber hinaus gibt fast die Hälfte der befragten FK und PO an, für mehr als eine Branche Software zu entwickeln.

Auch bei den Unternehmen der interviewten Personen unterscheiden sich die Geschäftsmodelle. Die Software wird für den internen Gebrauch, im direkten Kundenauftrag oder für den lizenzierten Verkauf entwickelt. Zwei interviewte Personen arbeiten zudem für Unternehmen, die ihre Mitarbeitenden zu anderen Unternehmen entsenden.

3

ENTWICKLUNGSPROZESS UND BETRIEB

> „Es ist ja nicht so, dass wir uns keinerlei Gedanken über Sicherheit machen. Aber es ist natürlich auch immer ein Aufwand, den der Mitarbeiter dann mit sich trägt. Den man immer im Vergleich sehen muss."
> – Interviewteilnehmer*in

Für ein sicheres Endprodukt ist der Entwicklungsprozess, neben geeigneten Werkzeugen und kompetenten Entwickler*innen (vgl. Kapitel 4 und 5), von entscheidender Bedeutung. Daher ist die Idee des so genannten Security-by-Design-Ansatzes der Folgende: Im gesamten Entwicklungsprozess wird Security von vornherein und durchgängig mitbetrachtet. In Abbildung 4 ist ein hierzu passender Entwicklungsprozess[1] abgebildet, bei dem alle Disziplinen um beispielhafte security-stärkende Maßnahmen ergänzt wurden. Damit soll sichergestellt werden, dass möglichst früh in der Entwicklung gravierende Schwachstellen gefunden werden. Dies senkt den Aufwand, Fehler zu beheben, signifikant. Doch wie präsent ist das Thema Security bei den Entwickler*innen in den vier Disziplinen Anforderungsmanagement, Entwurf, Implementierung & Tests sowie Betrieb? Und wie gut sehen sich die Entwickler*innen, Product Owner und Führungskräfte in ihren aktuellen Prozessen hinsichtlich Security aufgestellt? Mittels unserer Studie haben wir diese drei Fragen beleuchtet. Dabei haben wir alle Teilnehmenden der Studie nach ihrer allgemeinen Einschätzung und zusätzlich die Entwickler*innen nach einer Einschätzung bezüglich der vier genannten Disziplinen befragt.

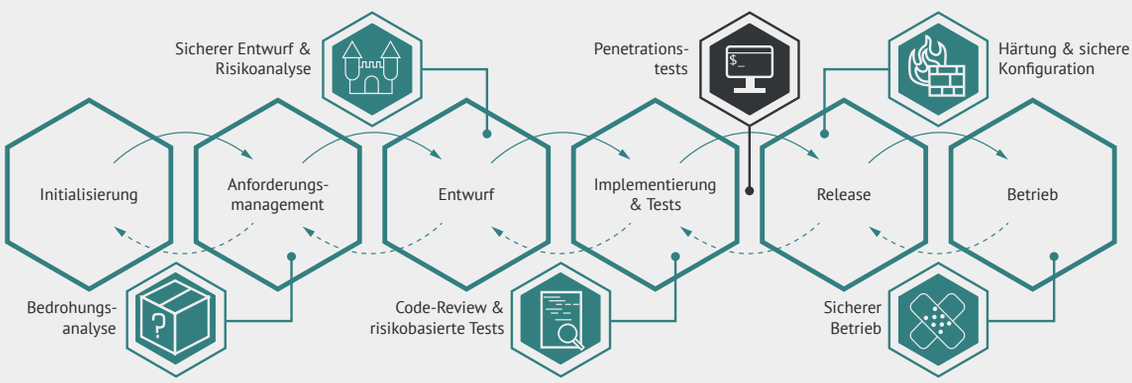


Abbildung 4: Security-by-Design-Entwicklungsprozess

[1] Für den Security-by-Design-Entwicklungsprozess ist es nicht relevant, ob das Produkt nach Wasserfall, V-Modell oder agil entwickelt wird. Bei der agilen Entwicklung erfolgen die Phasen Anforderungsmanagement bis Inbetriebnahme innerhalb eines oder mehrerer Sprints.

3.1 Wird Security im Entwicklungsprozess systematisch berücksichtigt?

Wir haben die Entwickler*innen gefragt, ob und wie sie Security in ihrem Entwicklungsprozess adressieren (vgl. Abbildung 5): 59% der befragten Entwickler*innen geben an, dass sie keine klaren Richtlinien und Prozesse zur sicheren Softwareentwicklung haben und 41% , dass ihre Security-Anforderungen (z.B. zur Verarbeitung sensibler Daten) nicht klar definiert bzw. bekannt sind.

Darüber hinaus geben 56% der Entwickler*innen an, dass ihre Sammlung an Werkzeugen unpassend ist. Dies kann daran liegen, dass keine passenden Werkzeuge am Markt verfügbar sind, aber auch daran, dass bessere Werkzeuge nicht genutzt werden können bzw. dürfen oder dass bisher nicht nach besseren Werkzeugen gesucht wurde.

Zudem geben 62% der Entwickler*innen an, dass es in ihrem Team keine feste Person gibt, die für Security zuständig ist. Dies kann entweder daran liegen, dass alle Personen des Teams Security berücksichtigen sollen und daher niemand explizit zuständig ist oder aber auch, dass Security nicht explizit betrachtet wird. Unsere generelle Empfehlung ist, diese Verantwortlichkeit einer konkreten Person zuzuweisen. Ein klares Aufgabenverständnis ist wichtig, so dass die Person dafür sorgen kann, dass auch alle im Team entsprechende Achtsamkeit walten lassen.

Basierend auf der Einschätzung der Entwickler*innen wird Security von vielen Teams nicht systematisch berücksichtigt. Somit ergibt sich ein hohes Risiko, dass diese eine dauerhaft hohe Qualität ihrer Softwareprodukte nicht gewährleisten können.

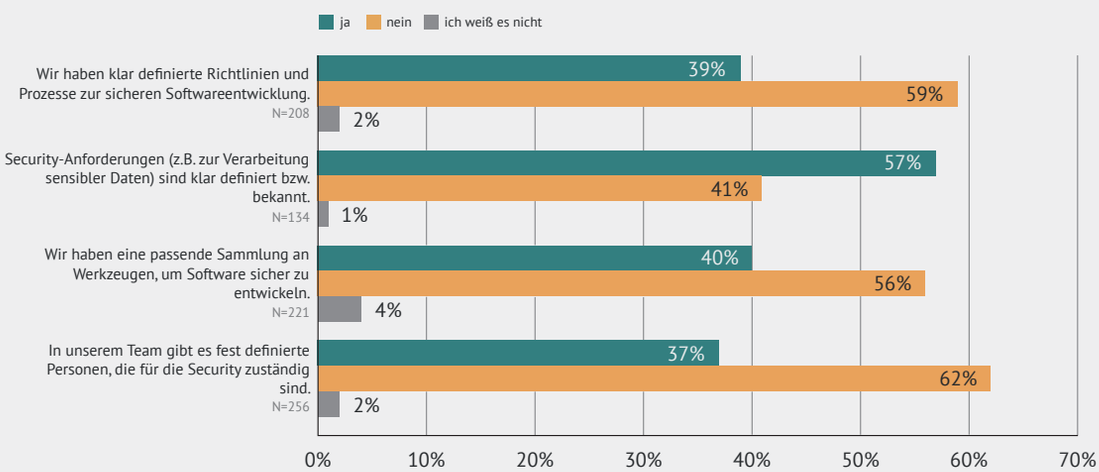


Abbildung 5: Berücksichtigung von Security im Entwicklungsprozess

In den Interviews mit den Führungskräften (FK) und Product Ownern (PO) hat sich gezeigt, dass Security im Softwareentwicklungsprozess keine hohe, sondern eher eine niedrige bis mittlere Priorität besitzt. Mehrheitlich wird auf den Pen-Test[2] verwiesen, der jedoch erst nach der Entwicklung stattfindet. Eine systematische Berücksichtigung der Security während der Entwicklung, wie beispielsweise durch die Verwendung von Checklisten, mittels einer festen Zuordnung von Verantwortung oder durch explizite Security-Richtlinien, ist nur sehr selten anzutreffen. Zudem haben die meisten PO angegeben, dass in den agilen Meetings (Planning, Retro, Review) Security nur eine geringe oder gar keine Rolle spielt. Insgesamt wird Security im Softwareentwicklungsprozess somit typischerweise unsystematisch, also ohne Festschreibung von konkreten Maßnahmen, behandelt. Hierdurch ergibt sich ein erhöhtes Risiko für die Qualität der entstehenden Produkte.

> „Security fängt meistens bei den ganzen Prozessen erst am Ende an. Wobei ich immer das Gefühl habe, es müsste schon sehr viel früher einsetzen." – Interviewteilnehmer*in

[2] Ein Pen-Test (Penetration Test) wird typischerweise vor dem Release einer Software durchgeführt – oftmals durch Dritte. Hierbei wird das fertige Produkt ausgeführt und auf Schwachstellen überprüft (bspw. falsche Konfigurationen, schlechte Passwörter, Angriffssicherheit gegen Standardangriffe).

3.2 Wird Security in den einzelnen Disziplinen systematisch berücksichtigt?

Von den in der Disziplin Anforderungsmanagement involvierten Entwickler*innen geben zwei Drittel (67%) an, dass auf Security geachtet wird (vgl. Abbildung 6). Allerdings haben nur 24% Vorlagen bzw. Standards für die Aufnahme von Security- Anforderungen. Somit scheinen die Anforderungen, falls sie überhaupt betrachtet werden, bei einem Großteil der Entwickler*innen unsystematisch betrachtet zu werden. Eine systematische Vorgehensweise würde das Endprodukt verbessern, da es die Gefahr senkt, dass Anforderungen falsch formuliert oder vergessen werden.

Einen Security-Experten, der die aufgenommen Anforderungen hinsichtlich Security überprüft, haben nur 35% der Entwickler*innen im Team. Dabei würde gerade eine klar benannte und hierfür ausgebildete Person dabei helfen, dass die Security-Anforderungen vollständig und sinnvoll sind, um so frühzeitig die Basis für ein sicheres Produkt zu legen.

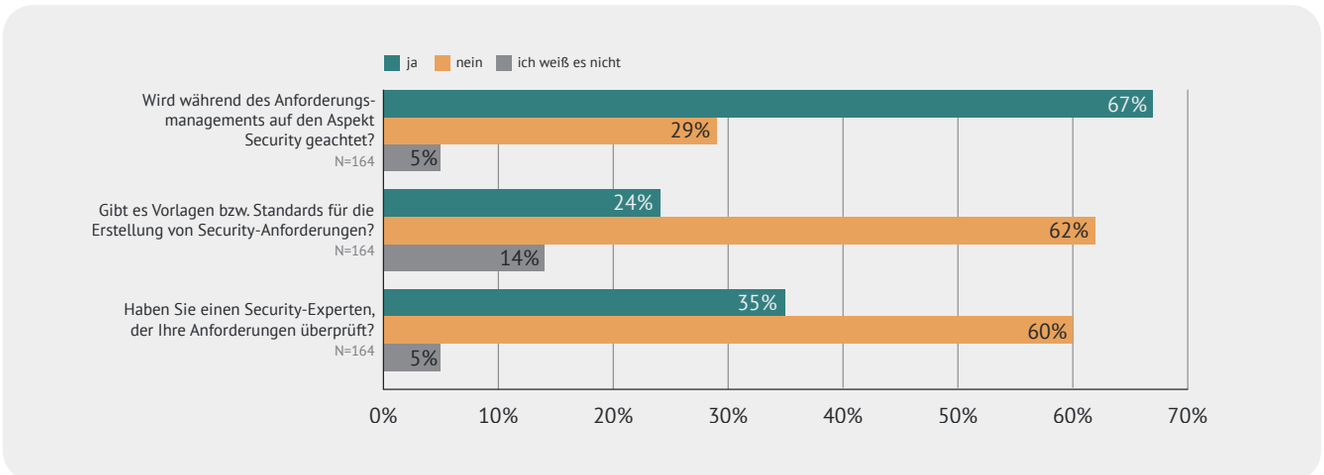


Abbildung 6: Berücksichtigung der Security in der Disziplin Anforderungen

Bei dem Entwurf zeichnet sich ein ähnliches Bild wie beim Anforderungsmanagement ab (vgl. Abbildung 7). So geben 77% der Entwickler*innen an, während des Entwurfs des Softwaresystems auf Security zu achten, jedoch haben nur 24% Vorlagen bzw. Standards. Zur Überprüfung der Sicherheit des Entwurfs haben erneut nur 32% der Entwickler*innen einen Experten.

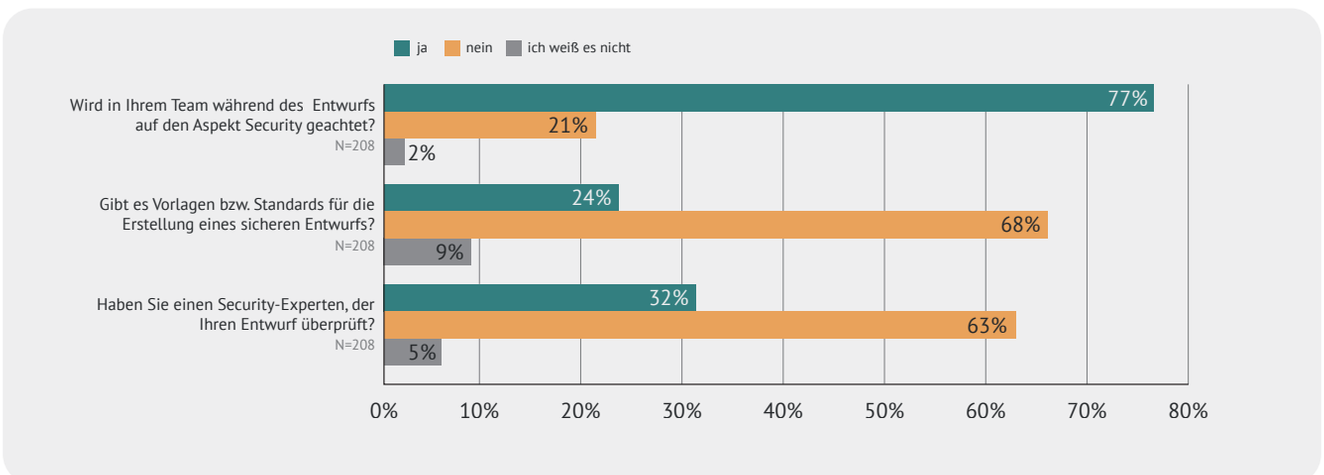


Abbildung 7: Berücksichtigung der Security in der Disziplin Entwurf

In der Disziplin Implementierung und Test haben 75% der Entwickler*innen angegeben auf Security zu achten (vgl. Abbildung 8). Dabei setzen 27% der Entwickler*innen entsprechende Vorlagen und Standards als Hilfestellung ein. Um sicherzustellen, dass alle Security-Eigenschaften von ihrem entwickelten Produkt eingehalten werden, haben 25% der Entwickler*innen einen entsprechenden Prozess. Ein finales Security-Review vor einem Release wird nur von 16% der Entwickler*innen durchgeführt.

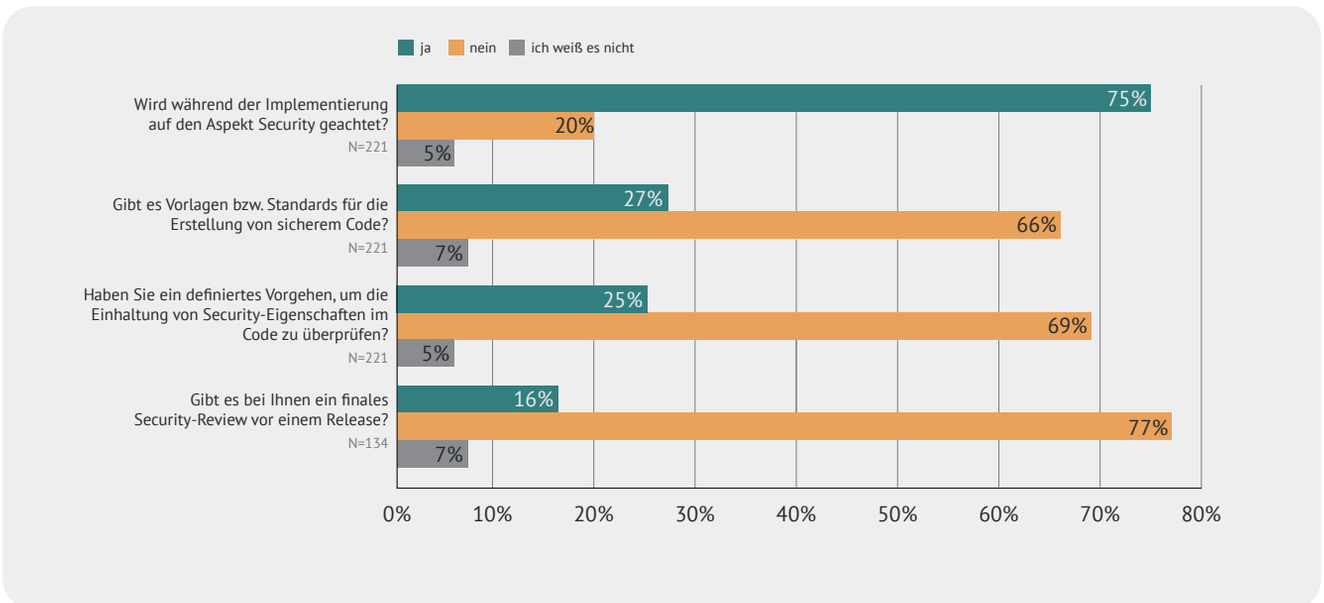


Abbildung 8: Berücksichtigung der Security in der Disziplin Implementierung und Test

Auf unsere Frage nach dem Zeitpunkt, wann die Software bezüglich Security analysiert wird, haben wir mehrere Antwortmöglichkeiten zugelassen, da ein mehrmaliges Analysieren sehr empfehlenswert ist (vgl. Abbildung 9). So gibt knapp die Hälfte (48%) der Entwickler*innen an, dass dies bereits während des Programmierens passiert. Ein weiterer beliebter Zeitpunkt zur Analyse der Security ist vor jedem Release (37%). Weitere Zeitpunkte waren vor einem Check-in (10%), nach einem Check-in (20%), während des Sprints (20%), sowie vereinzelte sonstige Zeitpunkte (7%), wie beispielsweise nach dem Release oder bei durchgeführten Pen-Tests. Somit lässt sich insgesamt feststellen, dass die absolute Mehrheit der Entwickler*innen ihre Software mindestens an einem Punkt im Hinblick auf Security analysiert. Allerdings geben auch 20% der Entwickler*innen an, ihre Software nie hinsichtlich Security zu analysieren.

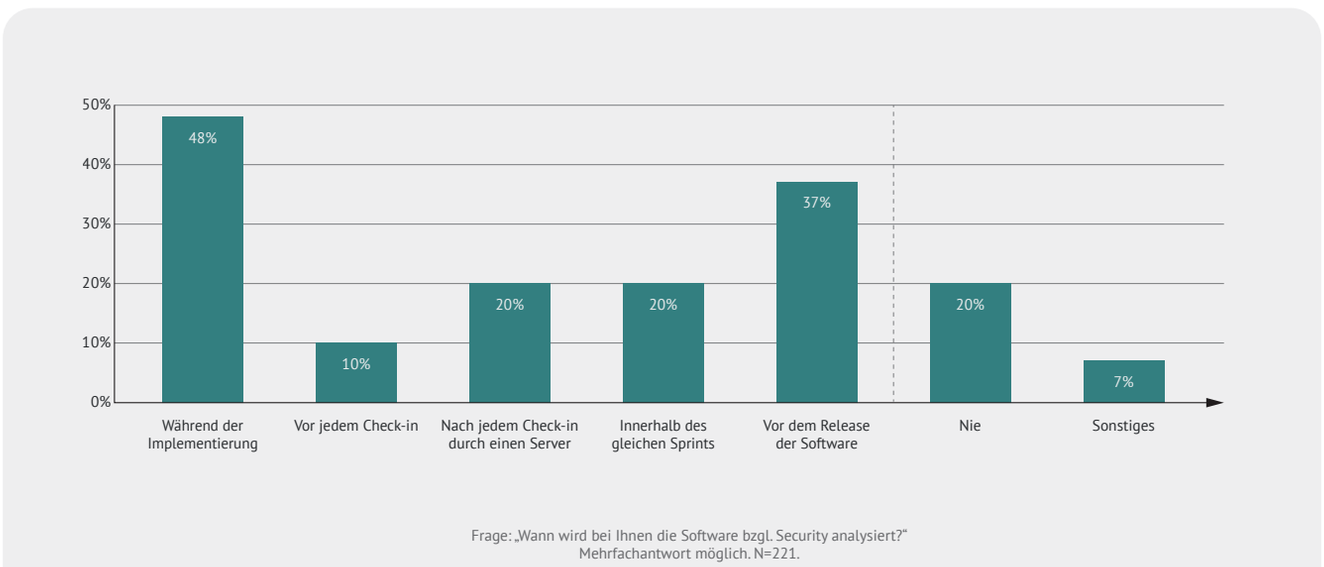


Abbildung 9: Wann wird die Software bzgl. Security analysiert?

Während des laufenden Betriebs der Software gibt es bei 22% der Entwickler*innen automatisierte Security-Checks (vgl. Abbildung 10). 28% führen automatische Security-Checks nach einem Release durch. Für den Fall, dass Sicherheitsprobleme (z.B. bzgl. genutzter Bibliotheken) im Betrieb befindlicher Produkte gefunden werden, geben 40% der Entwickler*innen an, dies mitzubekommen. Zudem sagen ebenfalls 40% der Entwickler*innen, dass sie klare Richtlinien zum Umgang mit (potenziellen) Sicherheitslücken, Angriffen, Datenlecks, etc. im Betrieb haben.

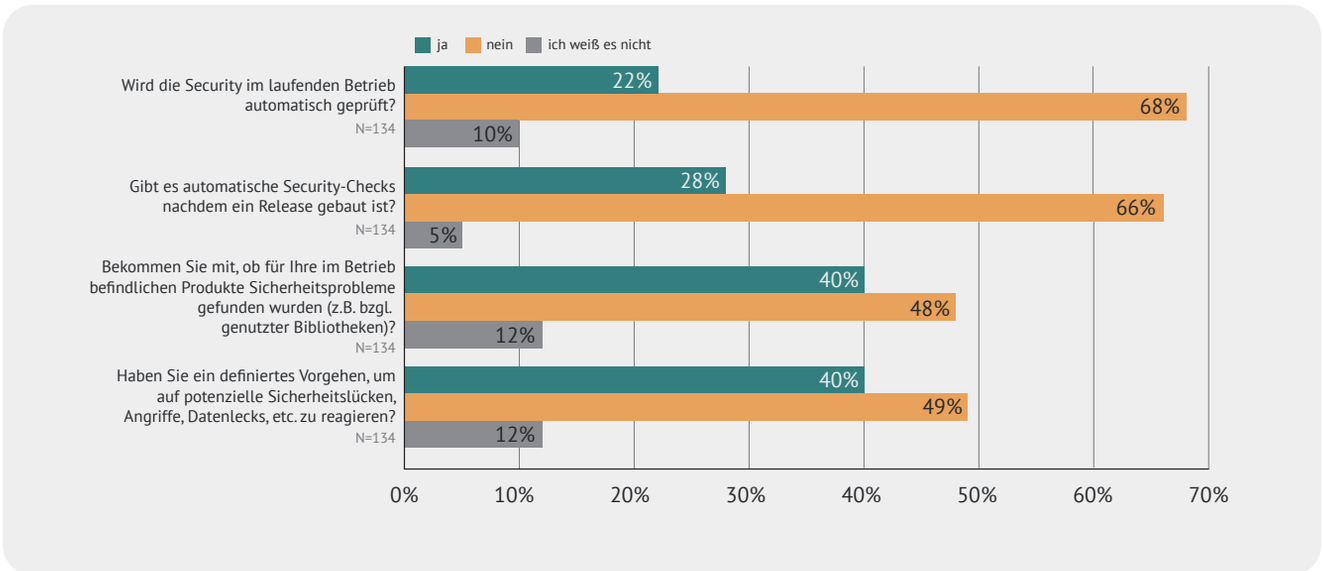


Abbildung 10: Berücksichtigung der Security in der Disziplin Betrieb

Um weitere Informationen zum Vorgehen bei einem Sicherheitsvorfall zu erlangen, haben wir die Führungskräfte (FK) und Product Owner (PO) hiernach befragt. Ein Drittel gibt an, keine Prozesse und Verantwortlichkeiten zu haben bzw. kennt sie nicht. Bei einem Viertel sind die Prozesse und unternehmensweiten Verantwortlichkeiten weitestgehend geregelt, wobei weder die befragten FK und PO noch deren Entwickler*innen konkrete Verantwortlichkeiten innerhalb eines Product Security Incident Response Teams (PSIRT)[3] haben. Bei den restlichen FK und PO sind die Prozesse und Verantwortlichkeiten nur teilweise geregelt bzw. bekannt, wobei meist nur auf einen Datenschutzbeauftragten verwiesen wird. Diese Rolle ist jedoch klassisch nicht für die Angriffssicherheit zuständig oder verantwortlich und somit auch nicht diesbezüglich geschult. Eine persönliche Beteiligung der FK und PO in einem Prozess zur Behebung von Sicherheitsvorfällen ist insgesamt selten.

> „Das kann der Herr [...] ganz sicher beantworten. Das ist unser Datenschutzbeauftragter und der kennt den Prozess auf jeden Fall. Ich selbst bin da nicht verantwortlich und kenne den Prozess auch ehrlich gesagt nicht konkret."
> – Interviewteilnehmer*in

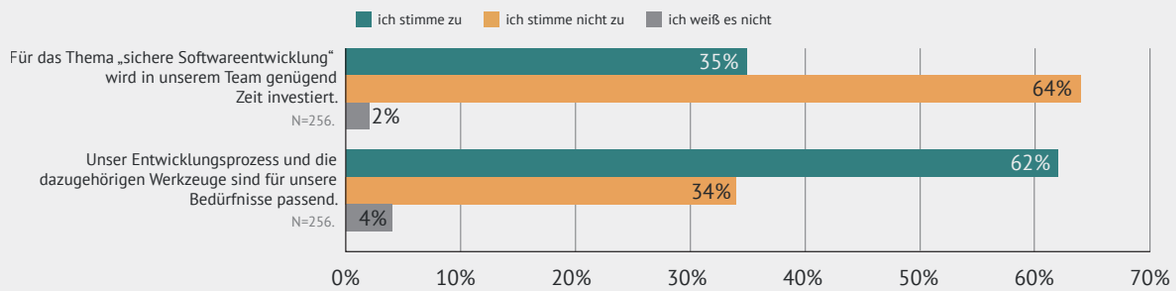
Insgesamt decken sich somit die Angaben der FK und PO mit denen der Entwickler*innen, dass mehrheitlich die Prozesse und Verantwortlichkeiten für einen Sicherheitsvorfall nicht gut genug definiert sind. Dabei ist dies insbesondere für das schnelle Reagieren auf neu entdeckte und gemeldete Schwachstellen wichtig. So kann beispielsweise durch einen professionellen Umgang mit bekanntgewordenen Schwachstellen schlechte Presse vermieden und verlorenes Vertrauen wiederhergestellt werden.

[3] Ein Product Security Incident Response Team ist innerhalb einer Organisation rund um das Auftreten von Sicherheitsvorfällen in Produkten tätig. Dazu gehört sowohl die Prävention durch entsprechende Awareness schaffende Maßnahmen und Fortbildungen/Schulungen, aber insbesondere auch das schnelle Reagieren und Beheben von neu entdeckten Schwachstellen. Zu diesem Zweck koordinieren die Teams Abläufe und Kommunikationswege innerhalb der eigenen Organisation unter Einbindung der meldenden Personen.

3.3 Sind die derzeitigen Prozesse geeignet, um Software sicher zu entwickeln?

Befragt nach dem Gesamtprozess meinen 64% der Entwickler*innen, dass in ihrem Team nicht genügend Zeit für die sichere Softwareentwicklung investiert wird (vgl. Abbildung 11). Dennoch sind 62% der Entwickler*innen der Meinung, dass ihr gesamter Entwicklungsprozess und die dazugehörigen Werkzeuge – unabhängig vom Thema Security – für ihre Bedürfnisse passend sind.

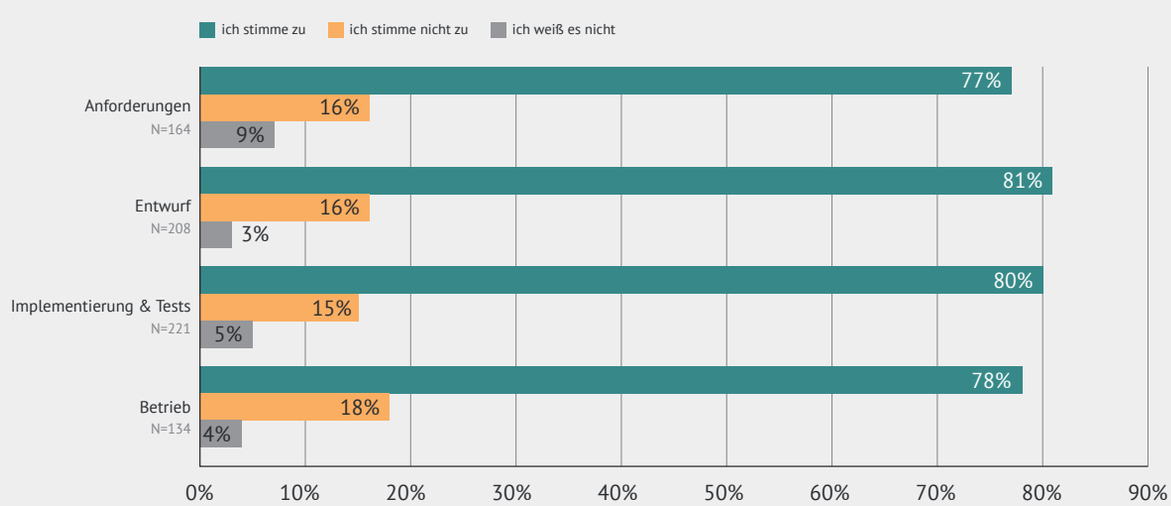
Dieser vermeintliche Widerspruch mag darin begründet liegen, dass eine sichere Softwareentwicklung nicht hoch priorisiert wird. Dadurch empfinden die Entwickler*innen zwar, dass zu wenig Zeit für eine sichere Softwareentwicklung investiert wird, sehen dies aber auch nicht als Problem, da die aktuellen Prozesse gefühlt funktionieren.



Die Anworten „ich stimme zu“ und „ich stimme nicht zu“ fassen sich zusammen aus den Anwortmöglichkeiten „stimme voll zu“ und „stimme eher zu“ (zu „ich stimme zu“) sowie aus „stimme eher nicht zu“ und „stimme gar nicht zu“.

Abbildung 11: Eignung der derzeitigen Prozesse

Im Vergleich der verschiedenen Disziplinen lässt sich ein starker gemeinsamer Trend erkennen (vgl. Abbildung 12): Genauere und verständlichere Prozesse wünschen sich stets deutlich mehr als zwei Drittel der Entwickler*innen (77% in der Disziplin Anforderungsmanagement, 81% im Entwurf, 80% bzgl. Implementierung & Tests sowie 78% im Betrieb). Daraus lässt sich ableiten, dass die Prozesse für einen kleinen Teil der Entwickler*innen funktionieren mögen, aber über alle Disziplinen hinweg genauere und verständlichere Prozesse benötigt werden.



Fragen: „Unsere gegenwärtigen Prozesse sollten genauer und verständlicher sein?“ Die Teilnehmenden wurden je Disziplin hierzu befragt. Die Antworten „ich stimme zu“ und „ich stimme nicht zu“ fassen sich zusammen aus den Antwortmöglichkeiten „stimme voll zu“ und „ stimme eher zu“ sowie aus „stimme eher nicht zu“ und „stimme gar nicht zu“.

Abbildung 12: Beurteilung der Genauigkeit und Verständlichkeit der Prozesse in den jeweiligen Disziplinen

3.4 Zwischenfazit

Die Studienteilnehmer*innen sind sich einig, dass die aktuellen Prozesse für eine sichere Softwareentwicklung und einen sicheren Betrieb stark verbesserungswürdig sind. Dies belegen die Antworten aller Disziplinen nach einem Wunsch für verständlichere und genauere Prozesse (jeweils ~80%), das Empfinden der meisten Entwickler*innen, dass zu wenig Zeit für sichere Softwareentwicklung investiert wird (~60%), aber auch die nur sehr wenig eingesetzten Maßnahmen (Vorlagen, Standards, Experten) für eine sichere Softwareentwicklung, was die Führungskräfte und Product Owner in den Interviews ebenfalls bestätigt haben.

Obwohl die Entwickler*innen angeben, dass auf Security geachtet wird, werden nur selten Maßnahmen umgesetzt, um eine systematische sichere Softwareentwicklung zu gewährleisten. Ohne diese Maßnahmen ist jedoch eine solche Gewährleistung sehr unwahrscheinlich. Somit liegt bei den Entwickler*innen eine unzutreffende Selbsteinschätzung vor, wenn sie sagen, dass auf Security geachtet wird. Eine mögliche Erklärung hierfür könnte darin liegen, dass den meisten Beteiligten nicht bewusst ist, was es für Möglichkeiten zur sicheren Softwareentwicklung gibt.

Besorgniserregend ist darüber hinaus, dass 20% der Entwickler*innen einräumen, während der Implementierung und Tests nicht auf Security zu achten.

> „Funktionalität vor Sicherheit. Es ist immer wichtiger, dass der Button von grün auf blau wird, anstatt dass man nochmal ein Penetrationstest über die Anwendung gehen lässt oder dergleichen. Weil das eine bringt Geld, macht die Benutzer zufrieden, und das andere, da hat man erst mal gar nichts von."
> – Interviewteilnehmer*in

4

WERKZEUGE

> „Also ich finde es total gut und wichtig, wenn wir Tools haben, die wir einsetzen können, sowohl in der Entwicklungsumgebung, aber auch noch viel stärker in unserer Build-Pipeline.“
> – Interviewteilnehmer*in

Werkzeuge sind in der heutigen Softwareentwicklung ein entscheidender Bestandteil zur erfolgreichen Umsetzung von Projekten. Sie unterstützen Entwickler*innen, Fehler zu vermeiden bzw. frühzeitig zu finden und zu beheben. Zusätzlich können durch Werkzeuge manuelle und zeitaufwändige Tätigkeiten automatisiert werden, wodurch nicht nur Zeit eingespart, sondern auch Fehler deutlich reduziert werden können. Insgesamt führt ein systematischer Einsatz von Werkzeugen entlang des Entwicklungsprozesses zu einer effizienteren Entwicklung bei gleichzeitiger Steigerung der Qualität.

Einige Werkzeuge tragen insbesondere zur Gewährleistung der Security von Softwareprodukten bei, indem sie Entwickler*innen bei der systematischen Überprüfung von Security-Anforderungen, der Generierung von sicheren Code-Snippets oder dem Aufspüren von Schwachstellen zu unterstützen. Durch diese Unterstützung kann die Anzahl an Security-Fehlern deutlich reduziert und die Sicherheit der Anwendung gesteigert werden.

Neben all den Vorteilen hat der Einsatz von Werkzeugen allerdings auch Grenzen: Verschiedene Studien haben gezeigt, dass zu viele oder unsystematisch eingesetzte Werkzeuge dazu führen, dass die Qualität der Software langfristig sinkt. Dies trifft insbesondere auf den Aspekt Security zu[4].

In der vorliegenden Studie haben wir daher untersucht, inwieweit Werkzeuge heutzutage in der Entwicklung genutzt werden. Zusätzlich haben wir ermittelt, welcher Bedarf bezüglich solcher Werkzeuge besteht.

4.1 Nutzen Entwickler*innen Werkzeuge zur sicheren Softwareentwicklung?

Die befragten Entwickler*innen geben an, dass sie Werkzeuge entlang des gesamten Entwicklungsprozesses nutzen, jedoch in unterschiedlicher Intensität (vgl. Abbildung 13). In den Disziplinen vor der Implementierung (Anforderungsmanagement und Entwurf) werden vergleichsweise selten Werkzeuge eingesetzt. Im Anforderungsmanagement nutzen 18% der Entwickler*innen Werkzeuge zur Erhebung und Dokumentation von Security-Anforderungen. Im Entwurf setzen 14% der Entwickler*innen Werkzeuge zur Dokumentation von Security-Eigenschaften ein. In beiden Disziplinen werden häufig Ticketsysteme (z.B. Atlassian Jira) oder Microsoft Excel zur Dokumentation genutzt. Darüber hinaus werden Ergebnisse von Security-Analysen aus vorherigen Releases betrachtet, um weitere Security-Anforderungen zu erheben und im Entwurf entsprechende Eigenschaften zu berücksichtigen. 50% der befragten Entwickler*innen, die Werkzeuge nutzen, um Security-Anforderungen zu erheben bzw. zu erfassen, nutzen auch gleichzeitig Werkzeuge, um ihre Security-Anforderungen automatisch zu analysieren. Somit können sie frühzeitig Fehler finden und beheben.

[4] https://www.heise.de/news/Studie-Mehr-Tools-weniger-Sicherheit-4799924.html

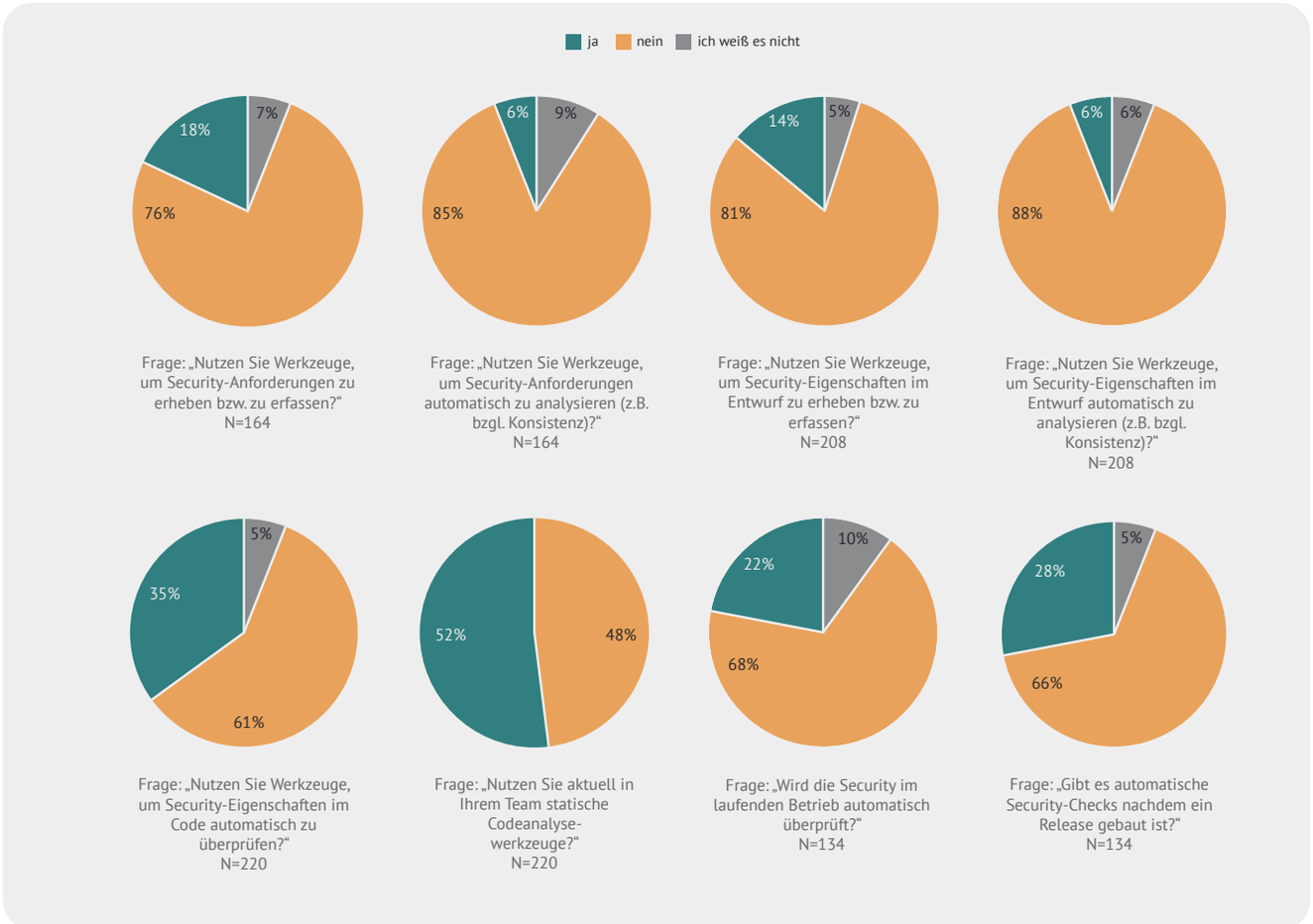


Abbildung 13: Heutige Werkzeugnutzung zur sicheren Softwareentwicklung

Während der Implementierung nutzen 35% der befragten Entwickler*innen Werkzeuge, um Security-Eigenschaften automatisch zu überprüfen. Darüber hinaus gibt ein Großteil der Entwickler*innen an, Werkzeuge zur statischen Codeanalyse einzusetzen (52%). Mit Hilfe von statischen Codeanalysewerkzeugen lassen sich verschiedene Eigenschaften im Code automatisch analysieren, ohne ihn ausführen zu müssen. Beispielsweise können security-relevante Eigenschaften, aber auch der Programmierstil oder doppelter Code erkannt werden. Am häufigsten nutzen die Entwickler*innen die statischen Codeanalysewerkzeuge SonarQube und FindBugs. Beide Werkzeuge definieren standardmäßig u.a. security-relevante Regeln.

Mit den meisten statischen Codeanalysewerkzeugen lassen sich bereits einfache Security-Eigenschaften im Code analysieren. Dazu gehört die Überprüfung, ob sich im Code hartcodierte Passwörter befinden. Darüber hinaus gibt es auch spezielle Werkzeuge zur statischen Analyse, mit denen komplexere Eigenschaften im Code geprüft werden können. Mit diesen Werkzeugen kann zum Beispiel geprüft werden, ob Benutzereingaben vor ihrer Weiterverarbeitung überprüft werden oder ob APIs korrekt genutzt werden. Allerdings setzen nur 50% der Entwickler*innen, die statische Codeanalysewerkzeugen nutzen, diese Werkzeuge zur Analyse von Security-Eigenschaften ein.

Ein ähnliches Bild zeigt sich auch bei der Verwendung von Werkzeugen während des Betriebs der Software. 22% der befragten Entwickler*innen geben an, dass die Security im laufenden Betrieb automatisch geprüft wird. Zudem analysieren 28% der befragten Entwickler*innen automatisch ein Release auf mögliche Schwachstellen.

4.2 Wie ist die Meinung der Führungskräfte und Product Owner zum Thema Werkzeuge zur sicheren Softwareentwicklung?

Im Allgemeinen sehen die Führungskräfte (FK) und Product Owner (PO) einen hohen Bedarf an passenden Werkzeugen zur sicheren Softwareentwicklung bei ihren Entwickler*innen und stehen einer Anschaffung und Einführung offen gegenüber.

Alle befragten FK und PO geben an, dass ihre Entwickler*innen sowohl kommerzielle als auch kostenlose Werkzeuge einsetzen. Des Weiteren sind sie sich einig, dass beim Thema Werkzeuge nicht die Kosten im Vordergrund stehen, sondern der Mehrwert, der durch die Nutzung der Werkzeuge erzielt werden kann. Zudem haben die FK und PO keine negativen Meinungen oder Vorurteile bzgl. der Nutzung von kostenfreien Werkzeugen. Darüber hinaus können die Entwickler*innen bei den meisten Unternehmen ihre präferierten kostenfreien Werkzeuge selbst beschaffen und nutzen.

Befragt zu ihren Entwickler*innen geben die meisten FK und PO an, dass diese nur Werkzeuge nutzen, von denen sie selbst überzeugt sind. Dadurch kann es vorkommen, dass für den gleichen Zweck unterschiedliche Werkzeuge im Einsatz sind – insbesondere wenn es mehrere kostenfreie Alternativen gibt. Hierbei ergibt sich somit das Risiko, dass diese Werkzeuge weniger zielgerichtet eingesetzt werden. Nur ein PO hat angegeben, dass die Nutzung von kostenlosen Werkzeugen regelmäßig diskutiert wird und Richtlinien für die Nutzung erstellt werden.

> „Da sind die Entwickler frei, die Werkzeuge zu nutzen, die sie für gut halten. Natürlich gibt es dann immer mal wieder Runden, in denen die Werkzeuge vorgestellt werden und die Vor- und Nachteile der verschiedenen Werkzeuge miteinander verglichen werden." – Interviewteilnehmer*in

Für die Anschaffung von kostenpflichtigen Werkzeugen verfügen die befragten FK und PO in der Regel über ausreichend Budget. Allerdings achten sie hier anders als bei dem Einsatz von kostenfreien Werkzeugen genau auf den Mehrwert, der bei der Nutzung der Werkzeuge entsteht. Erst wenn die Entwickler*innen den Mehrwert ausreichend begründen können, unterstützen die befragten FK und PO die Anschaffung. Zudem wird im Falle einer Anschaffung häufig verlangt, dass die Entwickler*innen diese Werkzeuge auch einsetzen müssen.

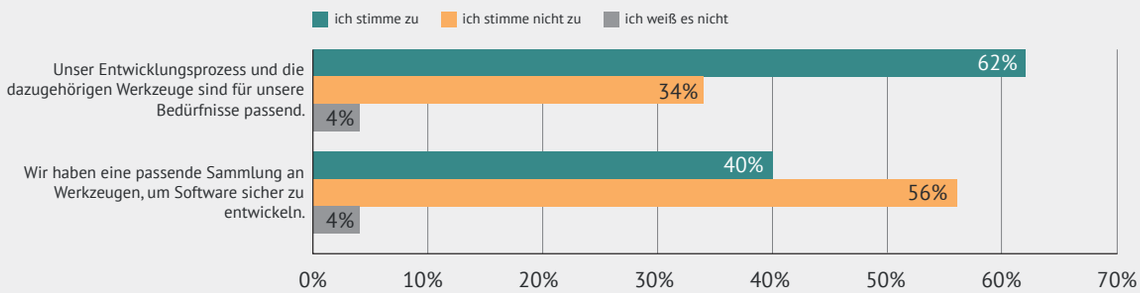
Allerdings berichteten die FK und PO, dass ihre Entwickler*innen nur sehr selten vorschlagen, ein kostenpflichtiges Werkzeug zu beschaffen. Dieser Widerspruch kann verschiedene Gründe haben:

- Die Auswahl an kostenfreien Werkzeugen ist für die Bedürfnisse der Entwickler*innen bereits passend.
- Am Markt befindliche, kostenpflichtige Werkzeuge sind für die befragten Entwickler*innen nicht geeignet. Dies könnte z.B. die Benutzbarkeit, die Performance oder fehlende Funktionen betreffen. Obwohl Bedarf und Budget vorhanden sind, werden daher keine weiteren Werkzeuge angeschafft.
- Die Entwickler*innen haben zwar Bedarf an besseren Werkzeugen, allerdings haben sie bisher noch nicht nach passenden Werkzeugen recherchiert.
- Die Entwickler*innen und FK und PO sprechen nicht oder nur selten über das Thema Werkzeuge. Dadurch kommt es zu keiner Verbesserung der Situation.
- Es fehlt die Übersicht über das Spektrum, welches mit Werkzeugen abgedeckt werden kann, oder das Verständnis für Security-Themen.

> „Auch da könnte ich nur mutmaßen weil diese Frage (Anmerkung der Redaktion: Bedarf für mehr Security-Werkzeuge) von den Entwicklern noch nicht an mich herangetragen wurde. Ich würde aber mutmaßen, dass sie über jede Form der Unterstützung sehr dankbar wären. Ich gehe davon aus, dass sie auch ein Interesse daran haben, möglichst sichere Software zu entwickeln. Aber wenn ich ehrlich bin, müsste ich sagen, ich habe so eine Frage von den Entwicklern noch nicht gestellt bekommen." – Interviewteilnehmer*in

4.3 Welchen Bedarf haben die Entwickler*innen?

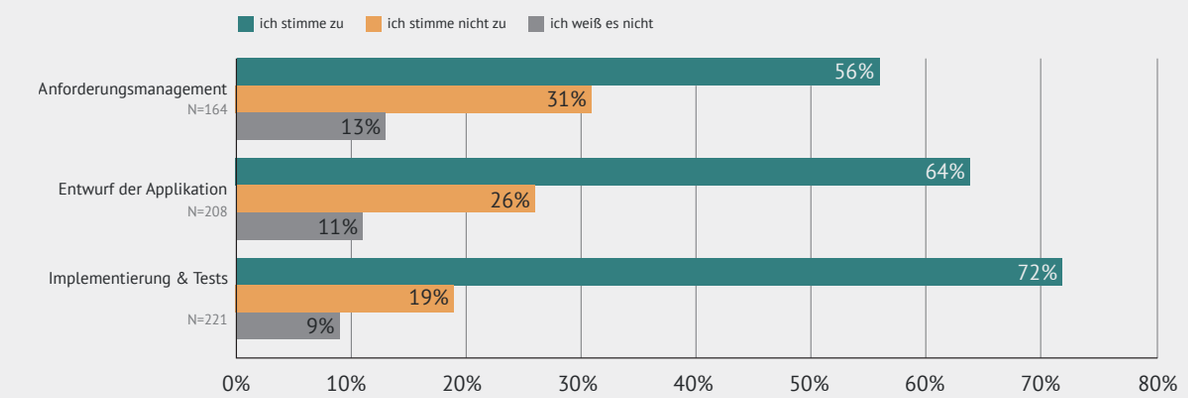
Befragt nach dem Entwicklungsprozess und den dazugehörigen Werkzeugen sind knapp zwei Drittel (62%) der Entwickler*innen der Meinung, dass diese für ihre Bedürfnisse passend sind. Ein Drittel (34%) widerspricht dieser Aussage jedoch (vgl. Abbildung 14). In einer zweiten Frage haben wir daher die Entwickler*innen gefragt, ob sie eine passende Sammlung an Werkzeugen haben, um Software sicher zu entwickeln. Hier geben nur noch 40% an, dass dies der Fall ist während die Mehrheit (56%) dies verneint. Somit besteht laut den Entwickler*innen ein deutlicher Bedarf an mehr bzw. besseren Werkzeugen.



Die Anworten „ich stimme zu" und „ich stimme nicht zu" fassen sich zusammen aus den Anwortmöglichkeiten „stimme voll zu" und „stimme eher zu" sowie aus „stimme eher nicht zu" und „stimme gar nicht zu". N=256.

Abbildung 14: Eignung der heutigen Werkzeuge zur sicheren Softwareentwicklung

Der Mangel an adäquaten Werkzeugen fällt in den einzelnen Entwicklungsdisziplinen jedoch unterschiedlich aus. Während in den Disziplinen Anforderungsmanagement und Entwurf der Bedarf nach mehr bzw. besseren Werkzeugen hoch ist (56% bzw. 64% , vgl. Abbildung 15), sind 72% der Meinung, dass bessere Werkzeuge für die Implementierung dabei helfen würden, die Aufgaben besser zu erledigen.



Frage: „Mehr bzw. bessere Werkzeuge würden uns helfen, unsere Aufgaben besser zu erfüllen." Die Teilnehmenden wurden je Disziplin hierzu befragt. Die Anworten „ich stimme zu" und „ich stimme nicht zu" fassen sich zusammen aus den Anwortmöglichkeiten „stimme voll zu" und „stimme eher zu" sowie aus „stimme eher nicht zu" und „stimme gar nicht zu".

Abbildung 15: Bedarf an Werkzeugen zur sicheren Softwareentwicklung

4.4 Zwischenfazit

Werkzeuge sind ein wichtiger Bestandteil einer erfolgreichen sicheren Softwareentwicklung. Die befragten Entwickler*innen, Führungskräfte (FK) und Product Owner (PO) sehen ebenfalls die Bedeutung dieses Themas. Allerdings hat sich gezeigt, dass die Verbreitung und Nutzung von Werkzeugen vor allem in den frühen Entwicklungsdisziplinen sehr gering ist. Selbst während der Implementierung setzt ein großer Teil keine Werkzeuge zur sicheren Softwareentwicklung ein. Dadurch fallen viele Fehler erst spät in der Entwicklung oder nach dem Release auf, wodurch teure und zeitintensive Reparaturen notwendig sind[5]. Zudem steigt das Risiko, dass im Produktiveinsatz Sicherheitsvorfälle geschehen.

Ein weiteres Ergebnis der Studie im Bereich Werkzeuge ist, dass die Entwickler*innen einen hohen ungedeckten Bedarf an Werkzeugen haben, obwohl es von Seiten der befragten FK und PO genügend Budget gibt, Werkzeuge anzuschaffen.

Bei der Nutzung von kostenfreien Werkzeugen sollte zudem mehr darauf geachtet werden, dass eine Abstimmung innerhalb des Teams geschieht sowie die Werkzeuge nutzenstiftend eingesetzt und gut in die bestehenden Prozesse integriert werden können.

[5] Quelle: NIST Planning Report 02-3, „The Economic Impacts of Inadequate Infrastructure for Software Testing“, 2002, Page 5-4.

5

SECURITY-KOMPETENZ

Die Security-Kompetenz aller Beteiligten (Entwickler*innen, Führungskräfte und Product Owner) ist ein entscheidender Baustein, um das Thema Security während der Entwicklung systematisch und vollständig zu adressieren. Wir haben daher alle Teilnehmenden der Studie um eine Selbsteinschätzung gebeten und sie gefragt, welche Kompetenzen heute vorliegen, wer welche Kompetenzen besitzen sollte und ob die Kompetenzen des Teams heute ausreichen.

> „Kompetenz kann man nie genug haben. Das Lernen hört nie auf.“ – Interviewteilnehmer*in

5.1 Wie schätzen sich die Entwickler*innen jeweils selbst ein?

Um eine Selbsteinschätzung der Entwickler*innen zu ihrer Security-Kompetenz zu erhalten, haben wir entlang des Entwicklungsprozesses (vgl. Abbildung 4) die Maßnahmen zur sicheren Softwareentwicklung in zehn Themengebiete gegliedert[6] und diese den Disziplinen Anforderungsmanagement, Entwurf, Implementierung & Test und Betrieb zugeordnet. Die Aufgabe der Entwickler*innen war es, jeweils einzuschätzen, ob sie selbst die Themengebiete und deren Konzepte kennen und, falls ja, ob sie praktische Erfahrungen hierzu gesammelt haben. Die Umfrage zielt somit auf die Frage ab, ob die Entwickler*innen grundlegende Kompetenzen und Erfahrungen haben und nicht, ob sie Expert*innen in den jeweiligen Themengebieten sind.

Abbildung 16 zeigt die Ergebnisse der Selbsteinschätzung sortiert nach dem Anteil an praktischen Erfahrungen (Spalte 4). Das bekannteste Themengebiet mit praktischen Erfahrungen ist die Eingabeprüfung, die beim Einlesen von Daten stets durchgeführt werden sollte (61%). Da zu diesem Themengebiet auch die bedeutende Angriffsart Injection[7] zählt, ist dieser vergleichsweise hohe Wert erfreulich. In allen anderen Themengebieten haben die Entwickler*innen deutlich weniger praktische Erfahrungen. So sind es im Themengebiet Patch Management nur noch 42%. In den verbleibenden Themengebieten haben zwischen 36% und 21% der Entwickler*innen praktische Erfahrungen.

Bzgl. der praktischen Erfahrungen ist auffällig, dass alle vier Themengebiete der Disziplin Implementierung sowie das Themengebiet Pen-Tests in der oberen Hälfte von Abbildung 16 zu finden sind. Hingegen sind alle anderen Themengebiete (alles was vor oder nach der Implementierung stattfindet) in der unteren Hälfte der Tabelle zu finden. Wir schlussfolgern daraus, dass dem Thema Security während der Implementierung mehr Priorität und Zeit im Vergleich zu den anderen Disziplinen eingeräumt wird. Für eine sichere Softwareentwicklung wäre es jedoch notwendig, dass auch die anderen Disziplinen mehr Priorität und Zeit für das Thema Security erhalten.

[6] Diese Themengebiete sind nicht vollumfänglich, decken aber ein sehr großes Spektrum der sicheren Softwareentwicklung ab.
[7] Unter anderem wird Injection von der OWASP Foundation als höchstes Risiko für Webanwendungen eingestuft: https://owasp.org/www-project-top-ten/

Dass der Pen-Test sich ebenfalls in der oberen Tabellenhälfte befindet, entspricht unserer Erwartungshaltung, da dieser eine weit verbreitete Maßnahme ist, um die Security eines Softwareprodukts vor einem Release zu prüfen. Ein Pen-Test hat jedoch auch viele Beschränkungen: Beispielsweise kann er – wie alle Testverfahren – nur Fehler aufdecken und nicht deren Abwesenheit aufzeigen. Darüber hinaus ist seine Durchführung meist auf wenige Tage begrenzt (Angreifer haben diese Beschränkung nicht) und er untersucht nicht, ob bereits die Anforderungen oder der Entwurf eine Schwachstelle enthalten. Es sind somit alle Themengebiete relevant und nicht nur das Themengebiet Pen-Test.

Themengebiet	Themengebiet nicht bekannt	Themengebiet und Konzepte bekannt, keine praktischen Erfahrungen	Themengebiet und Konzepte bekannt und praktische Erfahrungen	Themengebiet mindestens bekannt	Disziplin im Prozess
Eingabeprüfung (Sanitization)	14%	26%	61%	87%	Implementierung
Patch Management	25%	33%	42%	75%	Implementierung
Penetrations-Tests (Pen-Tests)	18%	46%	36%	82%	Test
Korrekte Verwendung von Kryptographie-Bibliotheken	22%	43%	36%	79%	Implementierung
Defensive Coding (Schreiben von sicherem Code)	27%	38%	36%	74%	Implementierung
Entwurf sicherer Architekturen	25%	45%	31%	76%	Entwurf
Security-Anforderungen und Security-Testfälle	27%	44%	29%	73%	Anforderungs-management
Incident Response (Verhalten bei Sicherheitsvorfällen	29%	45%	26%	71%	Betrieb
security-spezifische Code Reviews	31%	45%	24%	69%	Test
Durchführung einer Bedrohungsanalyse	27%	52%	21%	73%	Anforderungs-management

Die Themengebiete sind absteigend sortiert nach dem Anteil an praktischen Erfahrungen (Spalte 4). Die zusätzliche Spalte „mindestens bekannt" zeigt den prozentualen Anteil der Entwickler*innen, die das Themengebiet und dessen Konzepte kennen – unabhängig von den praktischen Erfahrungen (der jeweilige Wert entspricht der Summe der Werte aus der Spalte 3 und 4).

*Abbildung 16: Wie schätzen die Entwickler*innen ihre Security-Kompetenz je Themengebiet ein?*

Abbildung 16 zeigt zudem in Spalte 5 inwiefern jedes Themengebiet mindestens bekannt ist (diese ergibt sich aus der Summe der Spalten 3 und 4). Somit ergibt sich, dass alle Themengebiete bei mindestens 69% der Entwickler*innen mindestens bekannt sind. Die Eingabeprüfung sowie der Pen-Test erreichen hierbei sehr gute 87% bzw. 82%. Insgesamt lässt sich somit festhalten, dass ein Großteil der Themen einen guten Bekanntheitsgrad hat und kein Thema deutlich nach unten ausschlägt.

Eine zweite Auswertung der Selbsteinschätzung der Entwickler*innen ist in Abbildung 17 zu sehen. Hierbei haben wir je Entwickler*in gezählt, wie viele Themengebiete ihm*ihr unbekannt, bekannt, aber ohne Praxiswissen, oder bekannt und mit Praxiswissen sind. Das Ergebnis ist, dass die Entwickler*innen sehr unterschiedliche Kompetenzen haben. Bei der Betrachtung der Extreme wird ersichtlich, dass 2% gar keine Themengebiete kennen und nur 3% Praxiserfahrung in allen Themengebieten haben. 39% der Befragten kennen alle Themengebiete – im Umkehrschluss bedeutet dies, dass 61% mindestens ein Themengebiet nicht kennen. Darüber hinaus lässt sich die Gesamtheit der Entwickler*innen grob in drei Gruppen unterteilen. Gruppe 1 besteht aus ungefähr einem Viertel der Entwickler*innen (23%). Diese Gruppe kennt fünf oder mehr Themengebiete nicht und hat bei den wenigen Themengebieten, die sie kennen, nur sehr vereinzelt Praxiserfahrung. Gruppe 2 umfasst circa 41%. Sie kennen fünf oder mehr Themengebiete und haben bei einigen auch Praxiserfahrung, kennen aber auch einige Themengebiete nicht.

Die dritte Gruppe, die circa ein Drittel der Entwickler*innen umfasst, hat Praxiserfahrung in mindestens fünf der Themengebiete und kennt fast alle Themengebiete. Unserer Meinung nach sollten alle Entwickler*innen nahezu alle Themengebiete kennen. Ein Grund hierfür ist, dass alle Entwickler*innen in den agilen Meetings (Planning, Review, Retro) jeweils verstehen sollten, wovon gesprochen wird, bzw. mitreden können, sodass es zu einem Meinungsaustausch und kritischen Nachfragen kommen kann. Darüber hinaus sind wir der Meinung, dass alle Entwickler*innen in einem Großteil der Themengebiete praktische Erfahrungen aufweisen sollten. Ein Argument hierfür ist, dass Entwickler*innen heutzutage typischerweise in mehreren oder gar allen Disziplinen aktiv sind und daher praktische Erfahrungen benötigen, um den Aspekt Security berücksichtigen zu können. Insbesondere die Gruppe 1 benötigt somit einen Kompetenzausbau, um eine sichere Softwareentwicklung gewährleisten zu können.

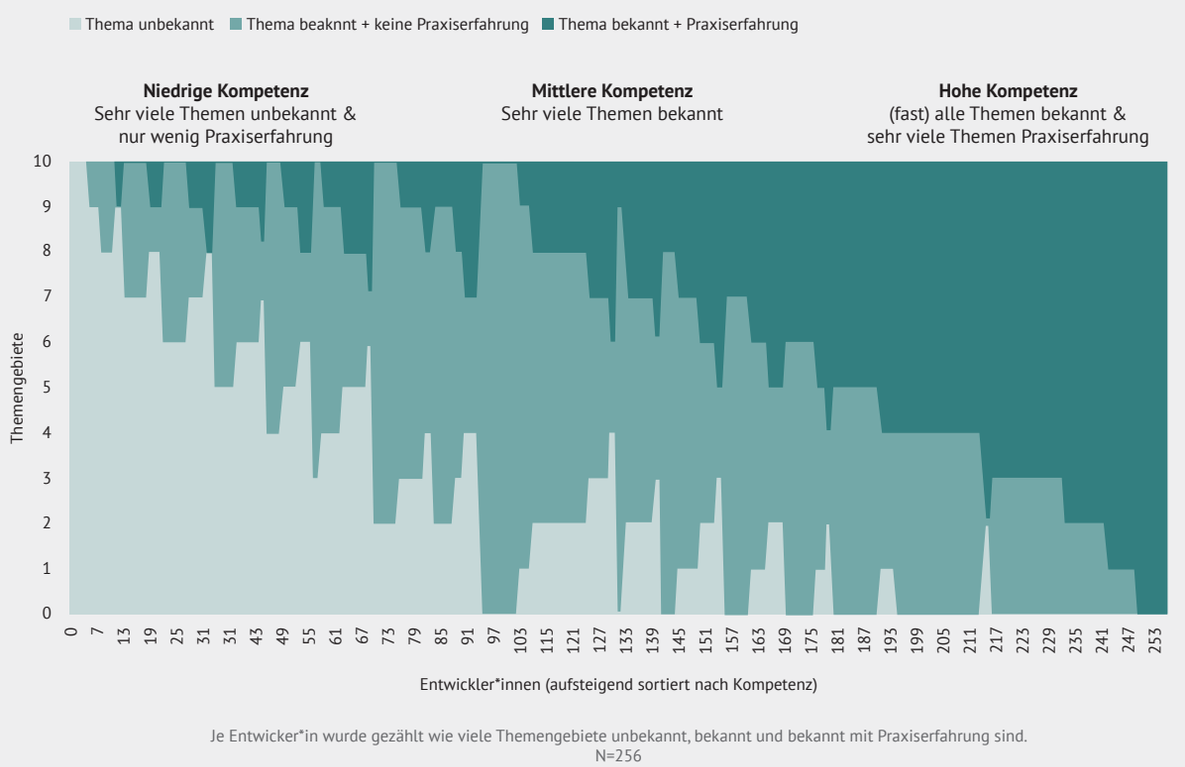


*Abbildung 17: Bekanntheit der Themengebiete je Entwickler*in*

5.2 Welche Kompetenzen haben die Führungskräfte und Product Owner?

Unserer Meinung nach benötigen alle Führungskräfte (FK) und Product Owner (PO) grundlegende Security-Kompetenzen. Beispielsweise sollten beide die für sie relevanten Gesetze und Standards kennen, verstehen was Security alles umfasst (inklusive der Unterscheidung von Datenschutz und Angriffssicherheit) und ihre jeweiligen Aufgaben und Pflichten in Bezug auf Security (u.a. Risikomanagement) kennen. Darüber hinaus sehen wir die FK u.a. in der Pflicht, ihre Mitarbeitenden bei der Weiterentwicklung ihrer Security-Kompetenzen systematisch zu unterstützen. Beim PO sehen wir u.a. die Aufgaben, Security-Anforderungen an das Produkt zu definieren und sich im Review-Meeting erklären zu lassen, wie die Security gewährleistet wird.

In unseren Interviews hat sich gezeigt, dass das Thema Security den FK und PO unzureichend bekannt ist. Nur für die Disziplin Implementierung und Test scheint oftmals Vorwissen vorhanden zu sein – insbesondere für die Themenbereiche Kryptographie bzw. Verschlüsselung und Pen-Tests. Weitere Methoden und Maßnahmen, die eher auf die Disziplinen Anforderungsmanagement, Entwurf sowie Release und Betrieb abzielen, werden jedoch nur sehr vereinzelt genannt.

Die Themen Datenschutz und die DSGVO werden sehr häufig von den FK und PO angesprochen. Einigen FK und PO ist jedoch nicht bekannt, dass es beim Thema Security nicht nur um den Datenschutz, sondern auch um die Angriffssicherheit geht. Des Weiteren sieht weniger als ein Drittel aller befragten FK und PO das Thema Security in ihrem Verantwortungsbereich. Unser Fazit ist daher, dass bei den meisten FK und PO die heutigen Kompetenzen eher gering sind.

> „Ich stelle fest, dass IT-Security ein Thema ist wo ich mich gar nicht so ganz sicher fühle und auf der anderen Seite mir die Relevanz von Sicherheit durchaus sehr bewusst ist und, ja, ich bemerke, dass mir weitere Kompetenzen guttun würden." – Interviewteilnehmer*in

Im Anschluss haben wir die FK und PO gefragt, ob sie sich selbst mehr Security-Kompetenz wünschen. Zwei Drittel wünschen sich dies – einige wenige hiervon geben dabei explizit an, dass sie sich nicht kompetent genug fühlen. 20% der FK und PO sagen, dass sie selbst keinen weiteren Kompetenzausbau benötigen, da sie, ihrer Meinung nach, ausreichend Kompetenzen haben, um ihren Aufgaben nachzugehen. Sowohl eine FK als auch ein PO geben hingegen an, dass sie keine Kompetenz bzgl. der sicheren Softwareentwicklung benötigen, da diese für die Erfüllung ihrer Aufgaben nicht relevant ist.

Da – wie zuvor erwähnt – alle FK und PO eine grundlegende Security-Kompetenz haben sollten und unsere Analyse zum Ergebnis gekommen ist, dass ein Kompetenzausbau notwendig erscheint, ist es erfreulich, dass ein Großteil der FK und PO dies ebenfalls so sieht. Diese Bereitschaft sollte sich daher positiv auf Maßnahmen zum Kompetenzausbau auswirken.

> „Ich müsste mich erst mal so weit einarbeiten, dass ich feststellen könnte was die relevanten Themen sind." – Interviewteilnehmer*in

5.3 Sollte jede einzelne Person des Teams eine hohe Security-Kompetenz haben?

Wünschenswert wäre es, wenn alle Personen eines Teams über eine hohe Kompetenz im Thema sichere Softwareentwicklung verfügen. Aber wie urteilen die Entwickler*innen hierüber? Das Ergebnis unserer Umfrage ist, dass 70% dieser Aussage zustimmen (vgl. Abbildung 18). Dies ist für uns ein überraschend hoher Wert, da dies zum einen bedeutet, dass die deutliche Mehrheit der Entwickler*innen nicht nur für wenige, sondern von allen Personen des Teams hohe Kompetenzen fordert. Und zum anderen, dass die deutliche Mehrheit nicht nur ein Grundwissen, sondern hohe Kompetenzen von ihren Kolleg*innen erwartet. Zusammenfassend ist eine deutliche Zustimmung unter den Entwickler*innen vorhanden.

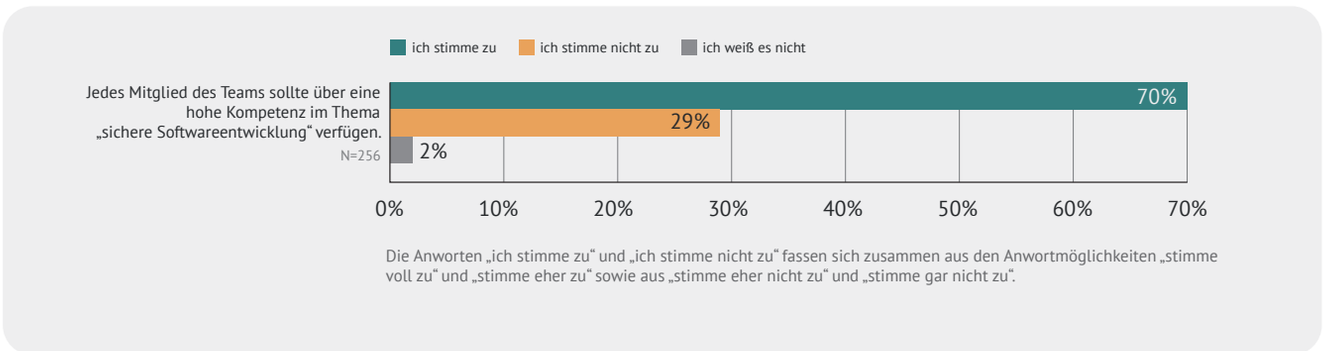


*Abbildung 18: Meinung der Entwickler*innen bzgl. der Kompetenz des Teams im Thema SSE*

5.4 Halten die Entwickler*innen die Kompetenzen ihres Teams für ausreichend?

Jeweils knapp zwei Drittel der Entwickler*innen meinen, dass die heutigen Kompetenzen ihres Teams nicht ausreichen, um ein passendes Security-Anforderungsmanagement zu betreiben, sichere Entwürfe und sichere Implementierungen zu erstellen oder einen sicheren Betrieb zu gewährleisten (vgl. Abbildung 19). Somit lässt sich festhalten, dass die Entwickler*innen ihre Kompetenzen in allen Disziplinen für nicht ausreichend einschätzen und ein insgesamt hoher Handlungsbedarf gesehen wird.

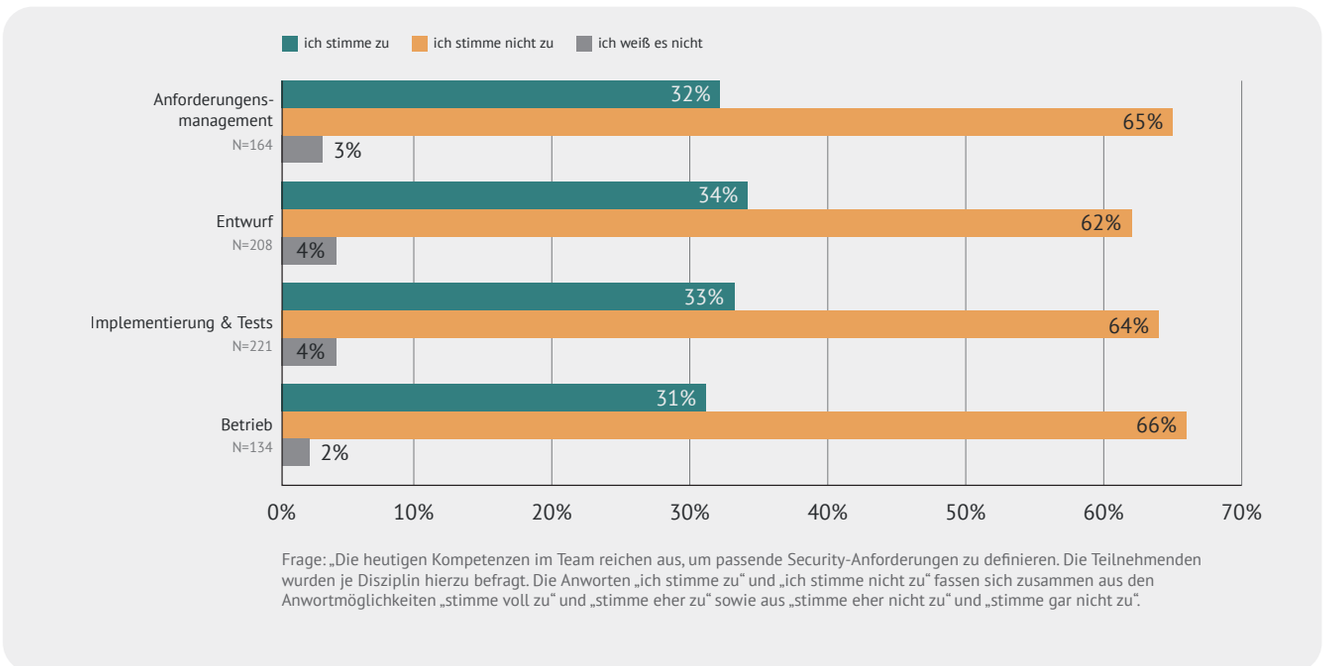


*Abbildung 19: Einschätzung der Entwickler*innen bzgl. der heutigen Security-Kompetenzen im Team*

Darüber hinaus haben wir alle Entwickler*innen gefragt, ob sie jeweils selbst durch Expert*innen im Thema der sicheren Softwareentwicklung weitergebildet werden möchten. Eine deutliche Mehrheit von 89% bestätigt diese Aussage (vgl. Abbildung 20), während die restlichen 11% sich keine Weiterbildung durch Expert*innen wünschen[8]. Unser Fazit ist daher, dass nahezu alle Entwickler*innen nicht nur mehr Kompetenzen bei ihren Teams, sondern auch bei sich selbst als notwendig erachten.

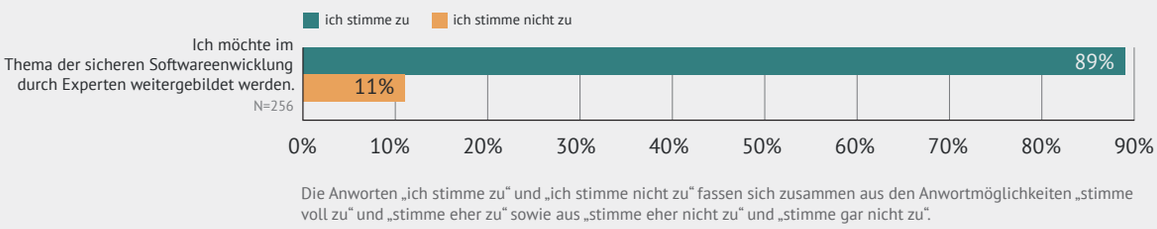


*Abbildung 20: Weiterbildungsbedarf der Entwickler*innen durch Expert*innen*

5.5 Wie ist die Erwartungshaltung der Führungskräfte und Product Owner an ihre Entwickler*innen?

Wir haben die Führungskräfte (FK) und Product Owner (PO) gefragt, welche Kompetenzen bzgl. sicherer Softwareentwicklung sie von ihren Entwickler*innen erwarten. Das Ergebnis: Knapp mehr als die Hälfte der FK und PO nennt ein bis maximal vier Aktivitäten bezogen auf die Disziplin Implementierung. Insgesamt werden bzgl. der Implementierung folgende sechs Aspekte genannt:

- Verschlüsselung
- Sichere Kommunikation
- Login/Passwörter/Hashing
- Überprüfung von 3rd Party Bibliotheken
- Eingabeprüfung
- Beschränkung von Prozessberechtigungen

Die Verschlüsselung und sichere Kommunikation werden hierbei am häufigsten genannt (jeweils von circa einem Drittel aller FK und PO). Unabhängig von der Disziplin erwarten 20% der FK und PO, dass die Entwickler*innen sich selbstständig weiterbilden (u.a. im Selbststudium). Eine FK nennt eine Unternehmensrichtlinie zur sicheren Softwareentwicklung und erwartet, dass die Entwickler*innen die hierfür notwendigen Kompetenzen haben; hingegen hat ein Drittel der FK und PO keine Liste an Kompetenzanforderungen.

> „Ich erwarte von meinen Softwareentwicklern, dass sie ein gewisses Grundverständnis für Security haben.“
> – Interviewteilnehmer*in

[8] 25% der Entwickler*innen, die nicht durch Expert*innen in diesem Thema weitergebildet werden wollen, geben an anderer Stelle im Fragebogen an, dass sie einzig das Selbststudium bevorzugen.

Darüber hinaus werden drei Anforderungen von jeweils 2 FK bzw. PO benannt:

- Datenschutzkompetenz besitzen
- Aktuellen Stand der Technik kennen
- Bereitgestellte Werkzeuge nutzen

Alle weiteren Nennungen sind Einzelmeinungen; diese sind: das richtige Verhalten nach einem Security-Vorfall (Incident Response), die Fähigkeit Security-Vorfälle zu entdecken (Incident Detection), Sicherheitslücken mittels Pen-Tests entdecken, Abstimmung der Security-Konzepte mit anderen Teams, die Fähigkeit das eigene Team zu überzeugen und Verantwortung auf das Team zu übertragen, ein Security Mindset haben, selbst entscheiden können was wichtig ist für eine sichere Software, FK auf notwendige Security-Maßnahmen hinweisen.

Zuletzt seien noch zwei PO hervorzuheben, die sich deutlich von den anderen FK und PO unterscheiden: Einem PO genügt es, wenn die Entwickler*innen Datenschutz-Kompetenzen zum korrekten Umgang mit Kundendaten haben. Ein anderer PO hingegen sagt, dass bewusst keine Anforderungen an die Kompetenzen definiert werden und sich auf das Eigenstudium der Entwickler*innen verlassen wird.

Zusammenfassend lässt sich festhalten, dass die Anforderungen der FK und PO an ihre Entwickler*innen eher von geringem Umfang sind, sich stark unterscheiden und sich nur bzgl. der Disziplin Implementierung signifikante Überschneidungen ergeben. Um ihrer Verantwortung und ihren Aufgaben bzgl. Security nachkommen zu können, sollten FK und PO jedoch in der Lage sein, eine umfangreiche Liste an Anforderungen zu definieren, die alle Disziplinen umfasst. Eine wesentliche Ursache, warum dies derzeit nicht der Fall ist, sehen wir in der unzureichenden Security-Kompetenz der FK und PO, die wir in Kapitel 5.2 ausführlich diskutiert haben.

5.6 Wie schätzen die Führungskräfte und Product Owner die Kompetenz der Entwickler*innen ein?

Mehr als die Hälfte der Führungskräfte (FK) und Product Owner (PO) ist der Meinung, dass ihre Entwickler*innen einen Kompetenzausbau im Bereich sicherer Softwareentwicklung benötigen. Knapp 20% der Befragten meinen, dass zumindest ein geringer Kompetenzausbau bzw. ein Kompetenzausbau bei einigen wenigen Entwickler*innen notwendig ist. Zwei FK und PO sind der Meinung, dass kein weiterer Kompetenzausbau notwendig ist, da sie davon ausgehen, dass die Entwickler*innen bereits genügend Kompetenzen besitzen. Als Fazit ergibt sich somit, dass laut den befragten FK und PO ein Kompetenzausbau im Thema Security bei einem Großteil der Entwickler*innen notwendig ist.

Darüber hinaus haben wir die FK und PO gefragt, ob ihre Entwickler*innen sich einen Kompetenzausbau im Thema Security wünschen. Mehr als die Hälfte der FK und PO schätzt, dass sich nur vereinzelte Entwickler*innen einen Kompetenzausbau wünschen. Die restlichen FK und PO teilen sich in drei etwa gleich große Gruppen auf: die erste Gruppe meint, dass ihre Entwickler*innen keinen weiteren Kompetenzausbau wünschen. Die zweite Gruppe gibt an, dass ihre Entwickler*innen Kompetenzausbau rege nachfragen. Hingegen weiß die dritte Gruppe nicht, ob ihre Entwickler*innen mehr Kompetenz ausbauen wollen. Insgesamt lässt sich somit festhalten, dass die FK und PO der Meinung sind, dass sich einige, jedoch nicht alle Entwickler*innen einen Kompetenzausbau wünschen. Dies steht jedoch im Gegensatz zur Erkenntnis aus Kapitel 5.4, wo 89% der Entwickler*innen angeben, dass sie durch Expert*innen weitergebildet werden möchten.

> „Ich glaube, dass wir Kompetenz bzgl. sicherer Softwareentwicklung ausbauen und aufbauen müssen. “ – Interviewteilnehmer*in

5.7 Zwischenfazit

In diesem Kapitel hat sich gezeigt, dass die Kompetenzen der Entwickler*innen oftmals zu gering und zudem sehr unterschiedlich sind. Beispielsweise belegt die Selbsteinschätzung der Entwickler*innen, dass die Bekanntheit der Themenfelder und das dazugehörige Praxiswissen sehr gering ist. Zwei Drittel der Entwickler*innen denken zudem, dass die heutigen Kompetenzen ihres Teams nicht ausreichen, um Software sicher zu entwickeln oder zu betreiben. Erfreulich ist jedoch, dass mehr als zwei Drittel der Entwickler*innen der Meinung sind, dass alle Personen des Teams hohe Kompetenzen haben sollten.

Viele Führungskräfte (FK) und Product Owner (PO) haben nicht ausreichende Kompetenzen, um ihren Aufgaben nachzukommen. Die meisten FK und PO sehen das ebenfalls so, da sich zwei Drittel selbst mehr Kompetenzen wünschen.

An ihre Entwickler*innen haben die FK und PO nur wenig Anforderungen. Darüber hinaus hat sich gezeigt, dass es keine einheitliche oder mehrheitliche Meinung bzgl. der geforderten Kompetenzen gibt. Zumeist beziehen sich die FK und PO auf die Disziplin der Implementierung. Die anderen Disziplinen scheinen bei den FK und PO weniger im Fokus zu stehen. Mehr als die Hälfte der FK und PO ist jedoch der Meinung, dass ihre Entwickler*innen einen Kompetenzausbau im Bereich sicherer Softwareentwicklung benötigen.

6

KOMPETENZAUSBAU

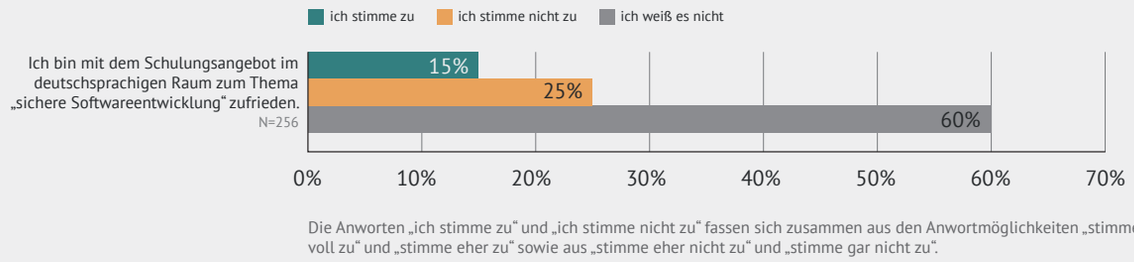
> „Ich glaube, dass dieses Thema so umfänglich ist, und vor allen Dingen auch so dynamisch ist, dass es damit nicht getan ist, irgendwie Entwicklern Schulungen zu geben, und dann zu erwarten ist, dass sie in der Lage sind, sicherheitsrelevante Software zu erstellen. [...] Das ist ja ein fortlaufender Prozess und das muss eigentlich permanent betreut werden." – Interviewteilnehmer*in

In diesem Kapitel beschäftigen wir uns damit, wie der heutige Kompetenzausbau in den Unternehmen aussieht. Wir thematisieren das Schulungsangebot im deutschsprachigen Raum, inwiefern sich Entwickler*innen über neue potenzielle Sicherheitslücken informieren und ob sie Meetups und Konferenzen besuchen. Im Anschluss erklären wir, wie Führungskräfte (FK) und Product Owner (PO) den Kompetenzausbau ihrer Entwickler*innen unterstützen und welche künftigen Weiterbildungsformate sich die Entwickler*innen wünschen.

6.1 Ist das Schulungsangebot im deutschsprachigen Raum zufriedenstellend?

Das Schulungsangebot im deutschsprachigen Raum bzgl. der sicheren Softwareentwicklung ist weniger als der Hälfte der Entwickler*innen (40%) bekannt (vgl. Abbildung 21). Eine mögliche Ursache hierfür ist, dass die Entwickler*innen bisher nicht aktiv nach (deutschsprachigen) Schulungen gesucht haben. Gleichzeitig liegt der Verdacht nahe, dass die existierenden Security-Schulungen im deutschsprachigen Raum bisher zu wenig bzw. zu selten beworben werden oder von den Entwickler*innen nicht wahrgenommen werden.

Von den 40% der Entwickler*innen, die das Angebot kennen, geben knapp zwei Drittel an, dass sie mit dem Angebot nicht zufrieden sind. Dies ist somit ein deutlicher Hinweis, dass die existierenden Schulungen, die den Befragten bekannt sind, den Bedarf der Entwickler*innen nicht vollständig abdecken. Ob das an den Inhalten oder am Format der Schulung liegt, gilt es in zukünftigen Studien zu erörtern.



Die Anworten „ich stimme zu" und „ich stimme nicht zu" fassen sich zusammen aus den Anwortmöglichkeiten „stimme voll zu" und „stimme eher zu" sowie aus „stimme eher nicht zu" und „stimme gar nicht zu".

*Abbildung 21: Zufriedenheit der Entwickler*innen mit dem Schulungsangebot im deutschsprachigen Raum*

Die große Mehrheit der Führungskräfte (FK) und Product Owner (PO) ist nicht mit dem Schulungsangebot im deutschsprachigen Raum bzgl. der sicheren Softwareentwicklung vertraut und kann es daher nicht bewerten. Vereinzelt werden von den FK unternehmensinterne Weiterbildungsmöglichkeiten genannt, jedoch haben diese laut den FK und PO typischerweise keinen oder nur einen geringen Fokus auf Security. Eine FK spricht zudem das Thema an, dass in der Ausbildung und der universitären Lehre zu wenig Kompetenz zur sicheren Softwareentwicklung vermittelt wird. Des Weiteren wird von den FK und PO oftmals angemerkt, dass Security-Themen in der Auswahl entsprechender Weiterbildungen nicht von den Entwickler*innen präferiert werden. Ein neues Framework, ein neues Tool oder eine andere Programmiersprache scheinen beliebter zu sein als Security. Wie in Kapitel 5.2 dargestellt, sehen wir hier insbesondere die FK in der Pflicht, den Kompetenzausbau bzgl. Security bei ihren Entwickler*innen stärker voranzutreiben.

> „Das primäre Verlangen der Entwickler*innen nach Wissen sind eher Zertifizierungen und Schulungen bzgl. neuer Technologien. Das Thema Security steht nicht an erster Stelle.“ – Interviewteilnehmer*in

6.2 Informieren sich Entwickler*innen über neue potenzielle Sicherheitslücken?

Bezüglich des heutigen Kompetenzausbaus haben wir die Entwickler*innen gefragt, ob sie sich regelmäßig über neue potenzielle Sicherheitslücken informieren (beispielsweise in den genutzten Bibliotheken und Frameworks) und woher sie die entsprechenden Informationen erhalten. 55% gaben an, dass sie sich regelmäßig informieren (vgl. Abbildung 22). Unserer Meinung nach sollten nahezu alle Entwickler*innen sich regelmäßig über neue potenzielle Sicherheitslücken informieren – daher stufen wir diesen Wert als eher niedrig ein. Wobei es selbstverständlich legitim wäre, wenn die Entwickler*innen mittels Werkzeugen automatisch prüfen, ob für das Produkt, welches derzeit entwickelt bzw. betrieben wird, potenzielle Sicherheitslücken vorliegen.

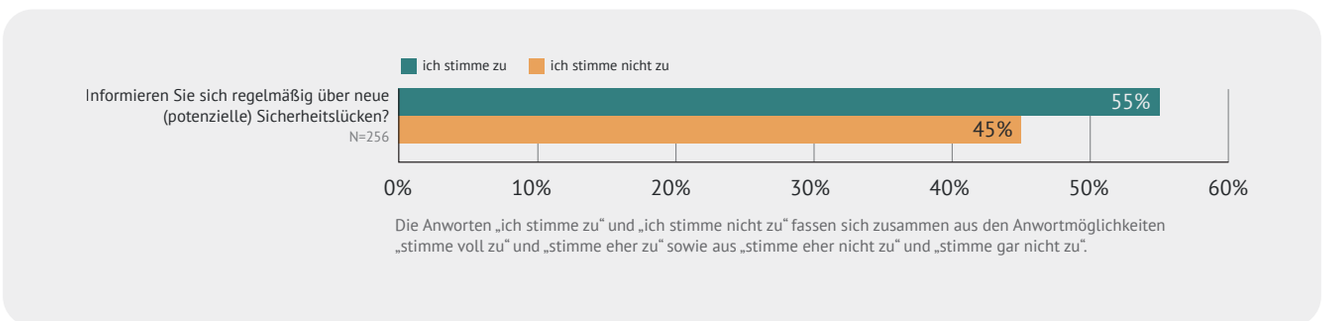


*Abbildung 22: Anteil der Entwickler*innen, die sich regelmäßig über neue potenzielle Sicherheitslücken informieren*

Anschließend gaben wir den Entwickler*innen die Möglichkeit ihre Informationsquellen zu nennen. Webseiten zu IT-Nachrichten sind bei den Befragten die primäre Informationsquelle (am häufigsten wurden Heise und Golem angeführt). Zwei Drittel der Befragten (64%) gaben Quellen aus dieser Kategorie an. Am zweithäufigsten wurden Newsletter bzw. Mailinglisten genannt (13%) und am dritthäufigsten Social Media (Twitter, Github, etc.: 11%). Weitere Nennungen im hohen einstelligen Prozentbereich waren das BSI, Blogs, CVE sowie Hersteller-Webseiten. Insgesamt lässt sich somit festhalten, dass zum heutigen Zeitpunkt den Redakteur*innen von IT-Nachrichten ein hohes Vertrauen entgegengebracht wird und von ihnen erwartet wird stets zeitnah über kritische Lücken zu berichten.

6.3 Besuchen Entwickler*innen Meetups und Konferenzen zu Sicherheitsthemen?

Eine weitere Möglichkeit zum Kompetenzausbau stellen Meetups und Konferenzen zu Sicherheitsthemen dar. 77% der Entwickler*innen gaben an, dass sie solche Veranstaltungen nicht regelmäßig besuchen (vgl. Abbildung 23). Diese Zahl deckt sich mit dem Ergebnis unserer Befragung der Führungskräfte (FK) und Product Owner (PO), da die Mehrheit von ihnen angibt, dass ihre Entwickler*innen nie oder nur sehr selten Meetups und Konferenzen zu Sicherheitsthemen besuchen. Typischerweise sehen die FK und PO die Entwickler*innen in der Pflicht für sich passende Veranstaltungen herauszusuchen. Sobald ein Vorschlag (innerhalb Europas) eingereicht wird, wird diesem in der Regel auch zugestimmt.

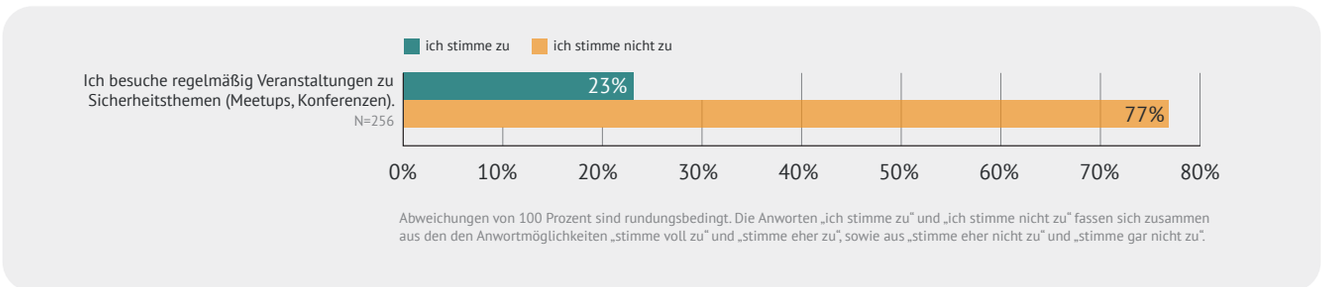


*Abbildung 23: Regelmäßiger Besuch von Veranstaltungen zu Sicherheitsthemen durch die Entwickler*innen*

6.4 Wie unterstützen Führungskräfte und Product Owner den Kompetenzausbau der Entwickler*innen?

Um einen Kompetenzausbau bzgl. der sicheren Softwareentwicklung bei ihren Entwickler*innen zu erreichen, ermöglichen die meisten Führungskräfte (FK) und Product Owner (PO) den Besuch von internen und externen Schulungen sowie den Besuch von Konferenzen. Darüber hinaus werden vereinzelt weitere Maßnahmen genannt: Beispielsweise wird ein Entwicklerfrühstück erwähnt, bei dem sich Entwickler*innen eines Bereichs zum Frühstück treffen und über verschiedenste IT-Themen austauschen, wobei das Thema Security häufig vorkommt. Eine andere Maßnahme, die genannt wurde, ist, dass je Team eine Person als sogenannter Security Champion aufgebaut wird, die das Thema Security bei der Softwareentwicklung umfassend berücksichtigt sowie die Kolleg*innen und den Product Owner hierzu einbindet (u.a. zur Definition und Abnahme von Security-Anforderungen).

„Es gibt natürlich Möglichkeiten für alle Entwickler, sich innerhalb ihrer Arbeitszeiten weiterzubilden im Rahmen von Konferenzen oder auch innerhalb von Projekten hier im Haus. Da ist ja auch schon ganz viel passiert, und da ist Security auch immer eines der Themen." – Interviewteilnehmer*in

Nahezu alle befragten FK und PO geben zudem an, dass ihre Entwickler*innen Zeit erhalten, um sich im Selbststudium u.a. bezüglich des Themas Security fortzubilden und sie es auch begrüßen, wenn ihre Entwickler*innen dieses Angebot annehmen. Viele FK und PO wissen jedoch nicht, ob diese Möglichkeit für Security-Themen genutzt wird. Darüber hinaus empfehlen die FK und PO diese Themen nicht. Nur ein PO gibt an, dass er es für nicht sinnvoll hält, wenn sich seine Entwickler*innen im Selbststudium weiterbilden, da hierfür ein Trainer*in dringend nötig ist.

„Sie bekommen von mir jederzeit die Zeit zum Selbststudium." – Interviewteilnehmer*in

Bis auf einen PO geben alle befragten FK und PO an, ausreichend finanzielle Mittel zur Verfügung zu stellen, um ihren Entwickler*innen interne als auch externe Weiterbildungsmaßnahmen zu ermöglichen. Typischerweise wird das Budget für interne und externe Schulungen genutzt, aber auch, um den Entwickler*innen die Teilnahme an Konferenzen zu ermöglichen. Meist werden alle Anfragen von den FK und PO gewährt solange diese die Veranstaltung als inhaltlich sinnvoll erachten.

In den meisten Fällen gibt es für Weiterbildungsmaßnahmen kein festes Budget, sondern das Geld wird auf Anfrage gewährt. Eine FK gibt jedoch an, dass allen Entwickler*innen ein festes Budget von 1.000 Euro pro Jahr zur Verfügung steht. Solange diese Grenze nicht überschritten wird, dürfen die Entwickler*innen vollkommen selbstständig entscheiden, für welche Weiterbildung sie das Geld ausgeben. Dieses Modell wird laut der FK von den Entwickler*innen sehr gut angenommen, u.a. um sich in Onlinekursen fortzubilden. Eine weitere FK gibt an, dass sie bewusst keine Budget-Grenze definiert hat, da sie ihre Entwickler*innen bzgl. der Weiterbildung nicht beschränken möchte – das Thema Security würde dabei stets bewilligt.

> „Es gibt bei uns die Besonderheit, dass wir keine Budget-Grenzen haben für solche Fortbildungen. Fort- und Weiterbildungen, die sinnvoll sind, können besucht werden. Der Begriff Security würde immer als sinnvoll von uns eingestuft werden." – Interviewteilnehmer*in

Darüber hinaus haben wir die FK gefragt, ob sie Security-Kompetenzen formal im Stellenprofil definiert haben. Nahezu alle befragten Führungskräfte geben an, dass dies nicht der Fall ist. Eine FK weiß nicht, ob dies der Fall ist, und eine weitere FK gibt an, dass dies zwar der Fall ist, die Security-Kompetenz jedoch nur bei Bewerbungen für „extrem security-relevante Rollen" entscheidend ist. Eine FK, die angibt, es nicht im Stellenprofil definiert zu haben, ergänzt jedoch, dass dieses Thema dennoch im Bewerbungsgespräch thematisiert wird.

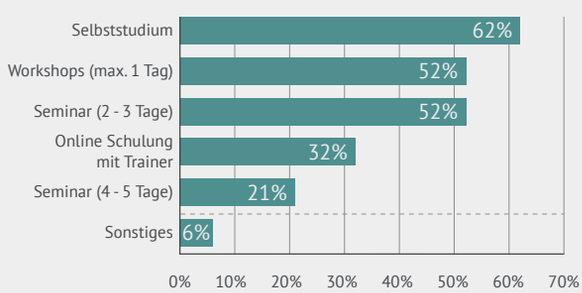
Zusammenfassend lässt sich somit sagen, dass die FK und PO ihren Entwickler*innen typischerweise ermöglichen, an internen und externen Schulungen sowie an Konferenzen teilzunehmen – das Budget ist hierbei kein begrenzender Faktor. Zudem ermöglichen und wünschen sich viele FK und PO ein Selbststudium zum Kompetenzausbau, wissen jedoch nicht, ob dieses Angebot genutzt wird. Die FK und PO überlassen es jeweils ihren Entwickler*innen, ob diese sich für einen Ausbau der Security-Kompetenz entscheiden. Einzig das Definieren eines Security Champions ist eine gezielte Maßnahme von Seiten der FK und PO, um die Security-Kompetenz auszubauen. Da in Kapitel 5.6 deutlich wurde, dass nahezu alle FK und PO einen deutlichen Ausbau der Security-Kompetenz bei ihren Entwickler*innen befürworten, sollten sie diesen systematischer vorantreiben und stärker einfordern.

6.5 Welches Weiterbildungsformat wünschen sich die Entwickler*innen?

Wie bereits in Kapitel 5.4 berichtet, geben 89% der Entwickler*innen an, dass sie im Themenbereich sichere Softwareentwicklung durch Expert*innen weitergebildet werden möchten (vgl. Abbildung 20).

Weiterhin haben wir die Entwickler*innen gefragt, welche Möglichkeiten zur Weiterbildung sie bevorzugen, wobei Mehrfachantworten möglich waren (vgl. Abbildung 24). Am häufigsten wurde hierbei das Selbststudium genannt (62%). Jeweils knapp mehr als die Hälfte (52%) findet einen maximal eintägigen Workshop bzw. ein 2-3 tägiges Seminar sinnvoll. Circa ein Drittel (32%) findet eine Online-Schulung mit einem Trainer/einer Trainerin sinnvoll und immerhin noch jede*r Fünfte ein Seminar über 4-5 Tage.

Als Fazit lässt sich festhalten, dass es sinnvoll wäre, wenn für die Vermittlung gleicher bzw. ähnlicher Inhalte unterschiedliche Formate angeboten würden, um die unterschiedlichen Lernpräferenzen zu adressieren.



Frage: „Welche Möglichkeiten zur Weiterbildung bevorzugen Sie?"
Mehrfachantwort möglich.
N=256.

*Abbildung 24: Welche Weiterbildungsformate wünschen sich die Entwickler*innen?*

6.6 Zwischenfazit

Insgesamt lässt sich zum heutigen Kompetenzausbau sagen, dass dieser an mehreren Punkten stark verbesserungswürdig ist. So ist den meisten Entwickler*innen, Führungskräften (FK) und Product Ownern (PO) das Schulungsangebot im deutschsprachigen Raum nicht bekannt, obwohl die meisten Beteiligten in Kapitel 5 angeben, dass ein Kompetenzausbau im Thema Security notwendig ist. Von den Entwickler*innen, die das Schulungsangebot kennen, sind zudem zwei Drittel nicht damit zufrieden – auch hier besteht offensichtlich Handlungsbedarf.

Nur etwas mehr als die Hälfte der Entwickler*innen informiert sich regelmäßig über potenzielle Sicherheitslücken und 77% der Entwickler*innen geben an, Meetups und Konferenzen zu Security-Themen nicht zu besuchen. Auch hier empfehlen wir eine Steigerung der Aktivitäten. Eine weitere Erkenntnis ist, dass die FK und PO ihren Entwickler*innen ermöglichen sich durch Schulungen, Konferenzen oder Selbststudium weiterzubilden. Jedoch wurde deutlich, dass die FK und PO den Ausbau der Security-Kompetenz bei ihren Entwickler*innen nicht systematisch vorantreiben oder einfordern. Die Ausnahme bildet hierbei die Maßnahme zur Einsetzung eines Security Champions, über die wenige FK und PO berichteten.

Letztlich hat sich gezeigt, dass zur Vermittlung von Kompetenzen unterschiedliche Formate angeboten werden sollten, um die unterschiedlichen Lernpräferenzen der Entwickler*innen zu adressieren.

7

SENSIBILISIERUNG ALLER BETEILIGTEN

> „Security ist aus meiner Sicht ein Riesenthema und man denkt zwischendurch mal drüber nach. Aber das sind Gedanken, die ich verdränge.“ – Interviewteilnehmer*in

In den vorangegangenen Kapiteln hat sich gezeigt, dass die Qualität der Entwicklungsprozesse zur sicheren Softwareentwicklung, die Verbreitung geeigneter Werkzeuge als auch die Security-Kompetenz meist nur niedrig oder mittelmäßig sind. Eine mögliche Ursache hierfür könnte eine unzureichende Sensibilisierung einzelner oder gar aller Beteiligten sein. Eine unzureichende Sensibilisierung hätte zur Folge, dass die Risiken von unsicheren Produkten nicht bewusst sind und aufgrund dessen keine Anstrengungen unternommen werden, die Prozesse zu verbessern, mehr oder geeignetere Werkzeuge zu nutzen oder Security-Kompetenz auszubauen. Im Folgenden werden wir uns daher mit der Frage beschäftigen, inwieweit alle Beteiligten bzgl. Security sensibilisiert sind. Hierbei untersuchen wir zunächst die Sensibilisierung der Führungskräfte (FK) und Product Owner (PO) und anschließend die Sensibilisierung der Entwickler*innen.

Zur Beantwortung der Frage unterteilen wir die Sensibilisierung in drei Kategorien:

- **Keine Sensibilisierung:** Eine Person ist für das Thema sichere Softwareentwicklung nicht sensibilisiert, wenn ihr das Thema unbekannt ist oder sich ihr dessen Nutzen nicht erschließt. Selbstverständlich sehen Personen dieser Gruppe keinen Handlungsbedarf, Kompetenzen auszubauen, Prozesse zu verbessern oder Werkzeuge zu nutzen.
- **Geringe Sensibilisierung:** Eine Person, der das Thema der sicheren Softwareentwicklung bekannt ist und die dessen Nutzen verstanden hat, ist gering sensibilisiert. Damit ein Thema bekannt ist, genügen grundlegende Kompetenzen. Diese Kompetenzen können in Ansätzen bereits durch eine der heute in Unternehmen üblichen DSGVO-Schulungen erlangt werden. In solchen Schulungen werden ausschließlich Datenschutz-Themen behandelt, Angriffssicherheit wird jedoch nicht thematisiert. Für eine umfassende Sensibilisierung ist jedoch das Wissen über beide Themengebiete notwendig.
- **Umfassende Sensibilisierung:** Für eine umfassende Sensibilisierung im Thema sichere Softwareentwicklung ist ein Bewusstsein für Datenschutz, Datensicherheit und Angriffssicherheit unerlässlich. Personen dieser Gruppe wissen daher um die Risiken, dass ein Fall von Datenverlust, -Diebstahl, oder -Manipulation eintreten kann und sieht deutlichen Handlungsbedarf, falls nicht genügend Kompetenzen vorhanden sind, Prozesse unvollständig sind oder Werkzeuge fehlen bzw. falsch genutzt werden. Zudem können umfassend sensibilisierte Personen klar zwischen den Begriffen Datenschutz und Angriffssicherheit unterscheiden und wissen diese auch anzuwenden. Darüber hinaus definieren wir für Personen dieser Kategorie, dass sie ihre Aufgaben und ihre Verantwortung im Kontext Security benennen können.

7.1 Sind die Führungskräfte und Product Owner sensibilisiert?

> „Ich glaube, wenn wir ein Security-Thema hätten, würden wir wirklich einzelnen Personen sagen: ‚Mach dich fit darin.‘ “
> – Interviewteilnehmer*in

In den Kapiteln 4 und 5 hat sich jeweils gezeigt, dass die FK und PO das Thema Security bezogen auf die Entwicklungsprozesse und Werkzeuge eher unsystematisch angehen, obwohl ein Handlungsbedarf jeweils notwendig ist. Dies spricht für eine eher geringe Sensibilisierung. Die einzige Ausnahme ist, dass die FK und PO stets bereit sind, Geld für kostenpflichtige Werkzeuge zur sicheren Softwareentwicklung zu investieren, solange dies von den Entwickler*innen gewünscht wird und es einen sinnvollen Nutzen gibt.

Wie in Kapitel 5.2 bereits berichtet, haben die meisten FK und PO unserer Einschätzung nach eher geringe Security-Kompetenzen. Oftmals bleibt es zudem bei der Aussage, dass Security wichtig ist, ohne dass entsprechende Maßnahmen, um diese sicherzustellen, genannt werden konnten. Insgesamt führt uns dies zum Schluss, dass nur wenige FK und PO umfassend sensibilisiert sind.

> „Ein Team, das rein für Innendienst-Anwendungen tätig ist, geht mit Anwendungs-Security sicherlich etwas anders um als ein Team, das Software baut, die auch auf Webseiten betrieben werden, also sogar aus dem Public Internet zugänglich sind, und auch entsprechend Angriffen täglich ausgesetzt sind. “
> – Interviewteilnehmer*in

Darüber hinaus haben wir in Kapitel 6 gezeigt, dass die FK und PO den Kompetenzausbau ihrer Entwickler*innen derzeit eher unsystematisch angehen, obwohl ein deutlicher Bedarf nach mehr Security-Kompetenzen seitens der Entwickler*innen besteht. Dies ist somit ein weiteres Indiz, dass die FK und PO maximal gering sensibilisiert sind.

Nahezu alle FK und PO, deren Produkte nicht öffentlich zugänglich sind, sondern unternehmensintern oder intern bei ihren Kunden genutzt werden, sind höchstens gering sensibilisiert, da sie typischerweise das Thema Security als nahezu unbedeutend betrachten. Oftmals wird hier die Firewall erwähnt, die für ausreichend Schutz sorgen soll. Die Gefahr eines Innentäters oder dass eine Firewall keinen ausreichenden Schutz bietet, ist vielen Befragten jedoch nicht bewusst.

Darüber hinaus haben wir geprüft, inwieweit das Kundenverhalten zur Sensibilisierung der FK und PO beiträgt. Der Großteil der befragten FK und PO gibt an, dass das Thema Security von vielen und zum Teil sogar von allen Kunden im Allgemeinen eingefordert wird. Diejenigen Kunden, die Security-Analysen einfordern bzw. selbst eine Prüfung durchführen, stammen laut den FK und PO häufig von großen Unternehmen, in welchen feste Security-Richtlinien definiert sind. Aber auch diese Richtlinien sind nicht perfekt: So wurde uns beispielsweise berichtet, dass ein Pen-Test zwar beim ersten Release verlangt wird, bei allen weiteren jedoch nicht mehr, auch wenn das Produkt über mehrere Jahre hinweg weiterentwickelt wird. Bei Kunden von kleinen Unternehmen gibt es diese Security-Richtlinien eher selten, daher hängen die Security-Anforderungen laut den befragten FK und PO stark davon ab, wie hoch das Security-Verständnis der jeweiligen Projektbeteiligten des Kunden ist. Insgesamt gibt es somit dennoch einige Kunden, die Security gar nicht einfordern oder nur bei der Projektvergabe und im weiteren Projektverlauf keine Zeit mehr für Security investieren wollen. Die befragten FK und PO sehen dieses Kundenverhalten jedoch nahezu immer kritisch. Somit ist dies ein Indiz für eine geringe bis umfassende Sensibilisierung der FK und PO.

Andersherum geben die meisten FK und PO an, dass sie ihre Kunden bzgl. Security sensibilisieren bzw. explizit anbieten Security im Projekt zu berücksichtigen – jedoch schaffen sie es nicht, ihre Kunden regelmäßig hiervon zu überzeugen. Unserer Meinung nach ist dies ein Zeichen, dass insbesondere der Vertrieb aber auch die FK und PO hierfür Weiterbildungen benötigen, um ihre Kunden von der Wichtigkeit zu überzeugen. Dennoch spricht dies für eine geringe bis umfassende Sensibilisierung der FK und PO, da ihnen dieses Problem bewusst ist.

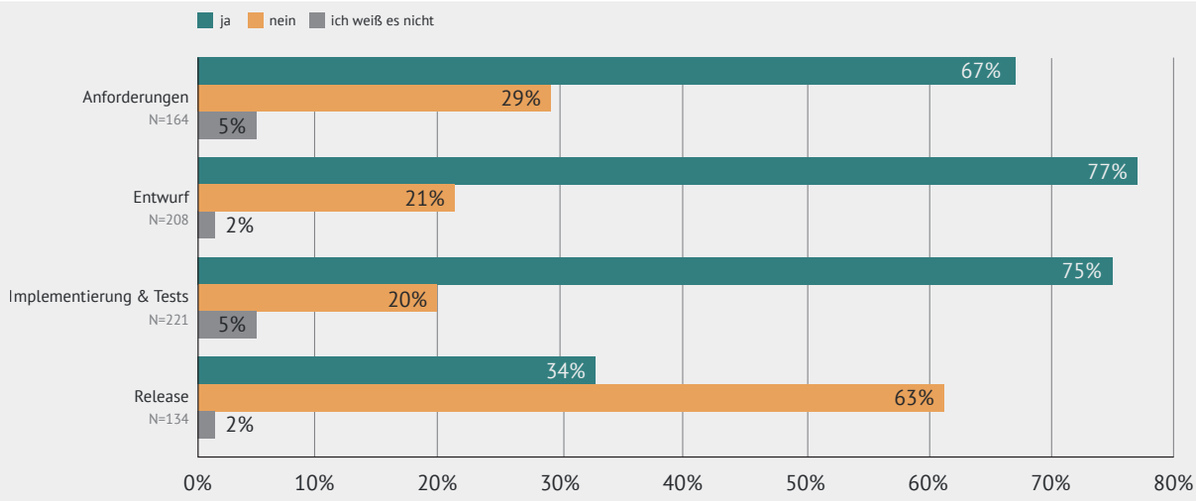
> „Die Sensibilität der Kunden wird zunehmend besser, gar keine Frage. Da aber auch unsere Anforderungen kontinuierlich steigen, würde ich sagen, das Verständnis wird nur in Teilen besser. “ – Interviewteilnehmer*in

Weniger als ein Drittel der FK und PO meint, dass sie Security gar nicht erst anbieten, da ihre Kunden hierfür keinen Bedarf hätten oder die Kunden den nötigen Aufwand hierfür nicht verstehen. Diese FK und PO sind somit höchstens gering sensibilisiert.

> „Also wenn Sie unsere Kunden jetzt fragen ‚Soll die Applikationen sicher sein?‘, sagt natürlich keiner ‚Nein, soll sie nicht‘, sondern sie soll natürlich sicher sein. Aber konkrete Maßnahmen sind bei unseren Kunden erschreckend selten verkaufbar. “
> – Interviewteilnehmer*in

7.2 Sind die Entwickler*innen sensibilisiert?

Ein deutliches Indiz für eine mindestens geringe bis umfassende Sensibilisierung bei einer Mehrheit der Entwickler*innen ist, dass während des Anforderungsmanagements, des Entwurfs und der Implementierung jeweils mindestens zwei Drittel der Entwickler*innen angeben, dass auf das Thema Security geachtet wird (vgl. Abbildung 25). Bzgl. des Release haben wir nicht direkt gefragt, ob auf Security geachtet wird, sondern ob es ein finales Security-Review vor dem Release gibt und ob es automatische Security-Checks nach einem Release gibt. Der Anteil der Entwickler*innen, die mindestens eine der beiden Fragen positiv beantwortet haben, liegt nur bei circa einem Drittel. Auch wenn somit die Fragestellung spezifischer ist, liegt dennoch der Verdacht nahe, dass die Mehrheit der Entwickler*innen bzgl. des Release der Software nicht umfassend sensibilisiert ist.



Fragen: „Wird während des Anforderungsmanagements auf den Aspekt Security geachtet?“, „Wird in Ihrem Team während des Entwurfs auf den Aspekt Security geachtet?“, „Wird während der Implementierung auf den Aspekt Security geachtet?“, „Gibt es bei Ihnen ein finales Security-Review vor einem Release?“, „Gibt es automatische Security-Checks nachdem ein Release gebaut ist?“. Die 4. und 5. Frage beziehen sich beide auf das Release. Falls eine von beiden Fragen mit Ja beantwortet wird, so haben wir dem Release auch ein Ja zugeordnet. Die Anworten „ja“ und „nein“ fassen sich zusammen aus den den Anwortmöglichkeiten „stimme voll zu“ und „stimme eher zu“ sowie aus „stimme eher nicht zu“ und „stimme gar nicht zu“.

Abbildung 25: Wird in den unterschiedlichen Phasen der Softwareentwicklung auf das Thema Security geachtet?

Wie in Kapitel 3 bereits erläutert, besteht jedoch ein Widerspruch bzgl. der Einschätzung der Entwickler*innen, dass in den drei Disziplinen Anforderungsmanagement, Entwurf und Implementierung & Test jeweils auf Security geachtet wird, und bzgl. der Angabe, dass nur selten spezifische Vorlagen, Standards, Prozesse und Werkzeuge existieren und es auch selten einen Security-Experten/eine Security-Expertin gibt . Dies ist somit ein klares Indiz, dass die meisten Entwickler*innen nicht umfassend sensibilisiert sind. Darüber hinaus geben 20% der Entwickler*innen an , dass sie ihre Software nie bzgl. Security analysieren (vgl. Abbildung 9 in Kapitel 3) – solange sie nicht vom PO oder Kunden davon explizit abgehalten werden, ist diese Gruppe somit ebenfalls nicht umfassend sensibilisiert.

Ein entscheidender Faktor zur Sensibilisierung für das Thema Security ist, inwieweit die Entwickler*innen Security-Kompetenz besitzen – also ob sie die verschiedenen Themengebiete kennen und welche Praxiserfahrungen sie hierin jeweils haben. Je höher die Kompetenz, desto höher stufen wir die Sensibilisierung für das Thema Security ein. Wie in Kapitel 5.1 berichtet, ist die Security-Kompetenz der Entwickler*innen sehr unterschiedlich verteilt. Die drei Kompetenzkategorien niedrig, mittel und hoch verteilen sich in etwa im Verhältnis 23 zu 41 zu 36. Entwickler*innen mit einer mittleren oder hohen Kompetenz sind unserer Einschätzung nach gering bis umfassend sensibilisiert. Die Gruppe an Entwickler*innen mit niedriger Security-Kompetenz wird von uns als gering sensibilisiert eingestuft. Die vier Entwickler*innen, die keines der zehn Themengebiete kannten, sind unserer Meinung nach, gar nicht sensibilisiert.

> „Bei vielen Entwickler*innen ist natürlich Security ein Begriff. Sie sind auch ein bisschen sensibilisiert, wie man das im Normalfall durch die Medien mitbekommt. Auch durch Konferenzen, wo das angesprochen wird. Aber ich habe jetzt nicht das Gefühl, dass das im täglichen Arbeiten wirklich gelebt wird.“
> – Interviewteilnehmer*in

Wie in Kapitel 5.4 beschrieben, hat die große Mehrheit der Entwickler*innen (89%) in unserer Umfrage den Wunsch geäußert, eine Weiterbildung im Bereich der sicheren Softwareentwicklung durch Expert*innen zu erhalten. Wir schließen daraus, dass ihnen die Relevanz des Themas bewusst ist und sie gerne mehr Kompetenzen ausbauen möchten. Dies ist für uns ein starkes Indiz, dass die meisten Entwickler*innen mindestens gering sensibilisiert sind. Bzgl. der restlichen 11% gibt es mehrere mögliche Gründe, dies nicht zu wollen, z.B. das Desinteresse am Thema, die Bevorzugung des Selbststudiums oder die Einschätzung, dass die eigene Kompetenz hoch genug ist.

Darüber hinaus haben wir die Führungskräfte (FK) und Product Owner (PO) gefragt, ob ihre Entwickler*innen sensibilisiert sind. Das Ergebnis ist, dass ein Drittel der FK und PO schätzt, dass innerhalb ihrer Teams alle Entwickler*innen umfassend sensibilisiert sind. Ein weiteres Drittel der FK und PO ist hingegen der Meinung, dass die meisten Entwickler*innen gar nicht oder nur gering sensibilisiert sind, jedoch einzelne Personen des Teams jeweils umfassend für das Thema Security sensibilisiert sind. Typischerweise haben diese einzelnen Personen entweder die explizite Aufgabe, sich um Security zu kümmern (z.B. der Security Champion des Teams) oder sie haben hierfür ein persönliches Interesse. Ein Drittel aller befragten FK und PO vermutet jedoch, dass selbst eine geringe Sensibilisierung nicht vorhanden ist. Insgesamt lässt sich somit feststellen, dass nach Einschätzung der FK und PO mindestens die Hälfte ihrer Entwickler*innen gar nicht oder nur gering sensibilisiert sind.

> „So ein Drittel der Entwickler haben durchaus Bauchschmerzen, wenn sie Sachen machen und die Security zu unwichtig ist. Das teilt sich auch nicht über Senior oder Junior, das ist eher persönliche Neigung.“ – Interviewteilnehmer*in

7.3 Zwischenfazit

Die befragten Führungskräfte (FK) und Product Owner (PO) sind typischerweise nur gering sensibilisiert. Es überwiegt die Haltung, dass Security wichtig ist, jedoch nicht wichtig genug, um diese systematisch und mit hoher Priorität bzgl. der Themen Prozesse, Werkzeuge und Kompetenzausbau anzugehen, obwohl dies – wie in den vorherigen Kapiteln aufgezeigt – notwendig wäre. Alle FK und PO sind insbesondere aufgrund der DSGVO für den Datenschutz sensibilisiert, jedoch eher selten für die Angriffssicherheit oder das Risiko von Innentäter*innen bei einem nicht-öffentlichen Einsatz ihrer Produkte. Sensibilisierte und engagierte FK oder PO sehen sich darüber hinaus häufig mit Unverständnis, Widerständen durch Vorgesetzte oder starren Prozessen konfrontiert.
Die absolute Mehrheit der Entwickler*innen ist ebenfalls nur gering sensibilisiert, was durch die Einschätzung der FK und PO bestätigt wird. Darüber hinaus gibt es aber jeweils eine Minderheit an Entwickler*innen, die entweder gar nicht oder umfassend sensibilisiert sind. Bemerkenswert ist hierbei der Widerspruch in der unzutreffenden Selbsteinschätzung vieler Entwickler*innen, die der Meinung sind, das Thema Security während der Entwicklung und Betrieb ihrer Produkte zu beachten, jedoch oftmals keine Methoden, Werkzeuge und Expert*innen haben und somit die Security nicht systematisch gewährleisten können.

8

FAZIT UND AUSWIRKUNGEN

Unsere Studie hat gezeigt, dass die Gewährleistung von Security für die Unternehmen in Deutschland eine vielschichtige Herausforderung darstellt. Im Folgenden fassen wir die Kernaussagen unserer Studie kurz zusammen:

- Die befragten FK, PO und Entwickler*innen sind zumeist nur gering sensibilisiert: Sie empfinden Security als wichtig, handeln jedoch nicht danach. Insbesondere fehlt vielen FK und PO das Bewusstsein für die Themen Angriffssicherheit und Innentäter*innen.

- Die meisten Entwickler*innen haben eine unzutreffende Selbsteinschätzung zum Thema Security: Sie meinen, dass sie das Thema Security beachten – geben jedoch an, keine passenden Maßnahmen, Prozesse, interne Experten und Werkzeuge zu haben.

- Alle Beteiligten der Softwareentwicklung benötigen mehr Kompetenz im Thema sichere Softwareentwicklung. Obwohl nahezu alle Teilnehmenden der Studie sich einen Kompetenzausbau wünschen, geschieht dieser bisher nur vereinzelt und unsystematisch.

Die derzeitige Situation der Softwareentwicklung hat mittel- und langfristig negative Auswirkungen auf die Sicherheit der Softwareprodukte. Die Kombination aus geringer Sensibilisierung und unzutreffender Selbsteinschätzung aller an der Softwareentwicklung beteiligten Personen führt dazu, dass der aktuelle Zustand als ausreichend empfunden wird und keine Verbesserung (bspw. eine Systematisierung der Prozesse und ein Kompetenzausbau) angestrebt wird. Die stetige Zunahme der Angriffe sowie die zahlreichen Sicherheitsvorfälle der letzten Jahre zeigen jedoch, dass die Gewährleistung der Security immer wichtiger wird, um Gefahren für Unternehmen und Kunden abzuwenden[9]. Somit ist es erforderlich, sich als Unternehmen kontinuierlich im Thema der sicheren Softwareentwicklung weiterzuentwickeln. Im nachfolgenden Kapitel geben wir daher Empfehlungen für FK, PO und Entwickler*innen, um eine Verbesserung des aktuellen Zustands zu erreichen.

[9] https://www.bsi.bund.de/DE/Publikationen/Lageberichte/lageberichte_node.html

9

EMPFEHLUNGEN ZUR VERBESSERUNG DES STATUS QUO

Zur Verbesserung der sicheren Softwareentwicklung empfehlen wir insbesondere einen Ausbau des Security-Basiswissens bei allen Entwickler*innen, aber auch den Führungskräften (FK) und Product Ownern (PO). Während Entwickler*innen ein umfangreiches Wissen und praktische Erfahrungen haben sollten, genügt es, wenn FK und PO ein grundlegendes Verständnis haben, sodass sie ihre Aufgaben und Pflichten bzgl. Security wahrnehmen können.

Zudem empfehlen wir, dass sich alle involvierten Beteiligten gegenseitig sensibilisieren, sodass es zu einer grundsätzlichen Wertschätzung von Security im ganzen Unternehmen kommt. Zum Beispiel sollte das Thema der sicheren Softwareentwicklung stets offen angesprochen werden – sei es im Sprint Review oder hinsichtlich möglicher neuer Werkzeuge oder Schulungen. Auf diese Weise können alle einen Beitrag zur sichereren Softwareentwicklung leisten.

Bzgl. der Themen Prozesse und Werkzeuge gibt es keine allgemeingültige Lösung, die für alle Unternehmen, Personen und Produkte sinnvoll ist. Wir empfehlen daher, dass die Entwickler*innen, FK und PO gemeinsam definieren, wie sie die Security im gesamten Entwicklungsprozess sicherstellen wollen.

Im Folgenden haben wir spezifische Empfehlungen für einzelne Rollen zusammengefasst. Diese sind als Anregung zu verstehen, um den aufgedeckten der Studie entgegen zu wirken.

9.1 Empfehlungen für Führungskräfte

- Definieren Sie gemeinsam mit der Geschäftsführung und den Product Ownern (PO), wer welche Verantwortung für Security-Themen hat. Insbesondere gilt es zu definieren, wer eine dauerhafte Gewährleistung der Security Ihrer Produkte sicherstellt und wie die Prozesse und Verantwortlichkeiten bei einem Sicherheitsvorfall sind (Stichwort: PSIRT).

- Sie sollten zumindest die zehn Security-Themengebiete kennen. Dieses Basiswissen ist relevant, um die security-spezifische Weiterentwicklung Ihrer Entwickler*innen zu begleiten aber auch um Ihrer Verantwortung bzgl. Security nachkommen zu können. Tiefgehende technische Details sind hierbei nicht zwingend notwendig.

- Achten Sie bei der Akquisition von Projekten darauf, dass Security Raum findet. Ohne entsprechende Budgetierung werden Sie später „draufzahlen".

- Sensibilisieren Sie Ihre PO und Teams hinsichtlich Security, sodass alle gemeinsam die Notwendigkeit des Handelns verstehen. Nutzen Sie das bestehende Schulungsangebot oder lassen Sie sich beraten, um Ihren individuellen Bedarf zu klären.

- Definieren Sie, was das Basiswissen für Security umfasst, dass alle PO bzw. alle Entwickler*innen haben sollen – unabhängig vom konkreten Projekt bzw. Produkt. Fordern Sie dieses mittelfristig ein.

- Stimmen Sie sich mit Ihrem PO und den Entwickler*innen ab, ob erweiterte Security-Kompetenzen für das zu entwickelnde Produkt nötig sind. Leiten Sie falls nötig Weiterbildungsbedarfe und entsprechende Maßnahmen ab.

- Motivieren Sie Ihre Mitarbeitenden, sich in Security-Themen weiterbilden zu lassen.

- Prüfen Sie, ob Security-Expert*innen (oftmals Security Champions genannt), mit der Aufgabe, Security-Wissen in die Teams zu tragen, eine sinnvolle Strategie für ihr Unternehmen sein kann. Im Allgemeinen empfehlen wir, dass jedes Team mindestens einen Security Champion haben sollte.

- Planen Sie im Budget Geld für Schulungen und Werkzeuge bzgl. Security ein.

9.2 Empfehlungen für Product Owner

- Falls Sie mit Entwickler*innen zusammenarbeiten, sollten Sie zumindest die zehn Security-Themengebiete (vgl. Kapitel 5.1) kennen, um entsprechend mitreden zu können (tiefe technische Details sind jedoch nicht zwingend notwendig).
- Falls Sie ein Proxy-PO sind: Klären Sie Ihre Kunden über Security-Risiken und -Maßnahmen auf. Sorgen Sie dafür, dass Ihre Kunden das Themengebiet wertschätzen und budgetieren.
- Sensibilisieren Sie Ihre Führungskräfte (FK) und Ihr Team hinsichtlich Security, sodass alle gemeinsam die Notwendigkeit des Handelns verstehen.
- Definieren Sie mit Ihrem Team, wie hoch die Bedeutung der Security für Ihr Produkt ist. Leiten Sie gemeinsam Security-Anforderungen ab, die Ihr Produkt erfüllen muss. Nutzen Sie Security-Reifegradmodelle, um Ihr Team und Ihr Produkt entsprechend weiterzuentwickeln.
- Definieren Sie Anforderungsprofile an die Entwickler*innen für Ihr Produkt. Machen Sie die FK hierauf aufmerksam.
- Fordern Sie Security vom Team während des Plannings, des Reviews und der Retroperspektive ein. Lassen Sie sich vom Team erklären, wie Sie die Security gewährleisten. Stellen Sie sicher, dass Sie die Erklärung verstehen, um selbst zu beurteilen, ob die Security gewährleistet ist.
- Planen und räumen Sie Ihrem Team Zeit ein, Security-Themen kontinuierlich in ihren Sprints anzugehen.

9.3 Empfehlungen für Entwickler*innen

- Kennen Sie die zehn Security-Themengebiete und verstehen Sie wie entsprechende Maßnahmen technisch umgesetzt werden können. Rufen Sie sich die Maßnahmen während des Programmierens ins Gedächtnis und sammeln Sie praktische Erfahrungen in möglichst allen Themengebieten. Fordern Sie von Ihrem Product Owner (PO) bzw. Ihrer Führungskraft (FK) Zeit und Budget für Weiterbildungen (intern und extern) ein.
- Definieren Sie, welche Rolle und Aufgaben Sie selbst und alle weiteren Softwareentwickler*innen bzgl. Security haben.
- Sensibilisieren Sie Ihre FK und PO hinsichtlich Security, sodass alle gemeinsam die Notwendigkeit des Handelns verstehen.
- Verschaffen Sie sich einen Überblick über existierende Werkzeuge. Verwenden Sie Werkzeuge zur Verbesserung der Security in Ihrer Entwicklungsumgebung (IDE) und Ihrer Build-Pipeline. Prüfen Sie zudem, ob Ihnen Werkzeuge für das Anforderungsmanagement und den Entwurf weiterhelfen.
- Verbreiten Sie Wissen im Team, beispielsweise zu Security-Grundwissen, Prozessen und Werkzeugen, sodass sie nicht als Einzelkämpfer*in agieren müssen.
- Überlegen Sie sich, wie ein Hacker Ihr System angreifen oder ausnutzen könnte – z.B. mittels Evil User Stories. Diskutieren Sie dies mit Kolleg*innen und Ihrem PO. Entwickeln Sie effektive Gegenmaßnahmen.
- Falls Ihr Team bisher keine Person mit hoher Security-Expertise (oftmals Security Champion genannt) hat, sollten Sie im Team mit dem PO diskutieren, ob die Einführung einer solchen Rolle sinnvoll ist.
- Stellen Sie in Ihrem Team sicher, dass es für jedes verwendete Werkzeug mindestens eine Person gibt, die Expertise für dieses Werkzeug hat und Ansprechpartner*in für das Team ist. Stellen Sie sicher, dass alle anderen Personen des Teams das Werkzeug gut genug kennen, um es sinnvoll nutzen zu können. Nutzen Sie interne oder externe Schulungen, falls nötig.
- Prüfen Sie Ihren eigenen Entwicklungsprozess, insbesondere auch hinsichtlich Security. Verbessern bzw. erweitern Sie Ihren Prozess fortlaufend.
- Sortieren Sie sich in ein Security-Reifegradmodell für agile Teams ein, um zu verstehen, wie gut Ihr Team bereits bzgl. Security ist. Leiten Sie ggf. Maßnahmen ab, um höhere Stufen im Reifegradmodell zu erreichen.

Ihr Ansprechpartner:

Dr. Stefan Dziwok | Senior Experte
Tel.: +49 5251 5465155 | Fax +49 5251 5465102
E-Mail: stefan.dziwok@iem.fraunhofer.de

Herausgeber
Fraunhofer-Institut für Entwurfstechnik Mechatronik IEM

Redaktion
Stefan Dziwok, Thorsten Koch, Sven Merschjohann,
Boris Budweg, Sebastian Leuer

Konzeption und Gestaltung
Caroline Just, Badir Al-Sayyed, Sarah Mrosek

Impressum

AppSecure.nrw

Empfohlene Zitierweise:
Dziwok et al.: Die AppSecure.nrw Software Security Studie, Paderborn, 2020.

Bildnachweise:
REDPIXEL | Adobe Stock (Titelbild)
Alle anderen Fotos und Grafiken: Fraunhofer IEM

Auflage: 01

EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

EFRE.NRW
Investitionen in Wachstum
und Beschäftigung

Fraunhofer
IEM

Diese Arbeit ist Teil des Forschungsvorhabens „AppSecure.nrw – Security-by-Design von Java-basierten Applikationen“.
Das Vorhaben wird aus Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE-0801379) gefördert.