

CodingCareers 2025

Ein Themenspecial der Heise Medien GmbH & Co. KG

Open-Source-Szene

Git-Projekte auf Codeberg umziehen

Seite 116

KI-Hilfe

Was IT-Fachleute unersetzlich macht

Seite 120

Softwareentwicklung

Wie produktiv ein Geniestreich ist

Seite 122



> Git-Projekte auf Codeberg umziehen

Der Umgang der US-Regierung mit Wissenschaft und Forschung hat auch Auswirkungen auf die Open-Source-Szene.

Roland Freist

Die wissenschaftliche Forschung in den USA steht unter Beschuss. Die Trump-Regierung kürzt Universitäten und Forschungseinrichtungen die Mittel, sodass laufende Projekte eingestellt und Leute entlassen werden müssen. Außerdem können viele Institute die Kosten für die Verwaltung und die Publikation ihrer Arbeiten nicht mehr stemmen.

Auch darüber hinaus setzt die Regierung die Universitäten unter Druck. Um Kürzungen zu entgehen, sollen die Mitarbeiter keine „woken“ Themen mehr bearbeiten, also keine Forschung zu konkreten Fällen und zu den Strukturen von rassistischer, sexistischer und sozialer Diskriminierung mehr betreiben. Wer sich weigert, muss mit dem Verlust von Fördergeldern rechnen.

Neben der wissenschaftlichen Welt sind auch Unternehmen betroffen. So hat etwa der deutsche Softwarekonzern SAP seine Diversitätsprogramme eingestellt und damit das Ziel, in seiner Belegschaft einen Frauenanteil von 40 % zu erreichen, aufgegeben. US-Technologiekonzerne haben oft schon vorausseilend gehandelt.

Mittlerweile hat daher eine Absetzbewegung begonnen. Viele europäische Firmen wollen auf Alternativen zu amerikanischen Diensten und Herstellern ausweichen, da sie Einschränkungen bei ihrer Arbeit befürchten. Gleichzeitig suchen US-Forschungsinstitutionen nach Einrichtungen in anderen Ländern, zu denen sie ihre Daten auslagern können, sodass diese dem Zugriff der US-Regierung entzogen sind. Zu den weltweit genutzten US-Diensten zählt etwa die Open-Source-Versionsverwaltung GitHub, die 2018 von Microsoft übernommen wurde. Doch wohin sollen Software-Projekte gehen?

Die Geschichte von Codeberg

Die Kritik am Kauf von GitHub durch Microsoft setzte bereits kurz nach der Bekanntgabe der Übernahme ein. Die Aktiven, die ihre Projekte auf GitHub veröffentlichten, befürchteten eine Einflussnahme

des Konzerns auf die Weiterentwicklung der Software und eine Ausweitung der Datensammlung, die die Firma bereits über Windows und Office betreibt, auf die Open-Source-Szene. Schon bald kündigten erste Projekte an, die Plattform zu verlassen.

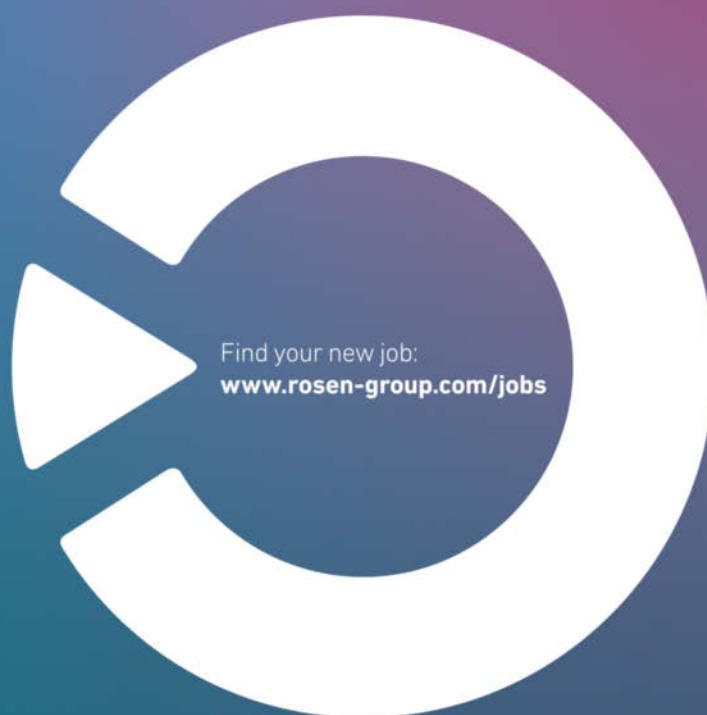
Noch 2018 spalteten die Entwickler Holger Wächter, Thomas Boerger und David Schneiderbauer die Entwicklungssoftware Gitea mit einem Projekt namens Teahub von GitHub ab. Im Januar 2019 wurde von anfangs 25 Mitgliedern in Berlin der Verein Codeberg e. V. (<https://codeberg.org>) gegründet. Aufgrund interner Auseinandersetzungen bei Gitea wechselte Codeberg im Jahr 2022 zu Forgejo (<https://forgejo.org>). Heute ist Codeberg ein Dienst auf Basis von Forgejo mit einigen zusätzlichen Funktionen wie etwa den Codeberg Pages.

Codeberg ist besser als GitHub

Codeberg hat von Anfang an auf eine größtmögliche Unabhängigkeit seiner Plattform geachtet. Der Verein finanziert sich durch Spenden, der Vereinszweck ist „der Betrieb von Internetsystemen zur Erstellung, Sammlung bzw. Verbreitung Freier Software (Open Source) und Freier Inhalte“. Das Codeberg-Repositorium liegt ausschließlich auf Servern in der Europäischen Union, die unter der Kontrolle

Git Commit to Your Future with Us!

Join the ROSEN Group.



Find your new job:
www.rosen-group.com/jobs

ROSEN
empowered by technology

von Codeberg stehen. Auf diese Weise will der Verein juristischen Auseinandersetzungen vorbeugen, die sich am amerikanischen Digital Millennium Copyright Act entzündeten. Um eine Kommerzialisierung auszuschließen, gibt es auf Codeberg kein User-Tracking und auch keine Verteilung von Third-Party-Cookies.

Hinzu kommt ein weiterer Punkt: 2021 stellte GitHub das Tool GitHub Copilot vor. Es soll die Entwicklung beim Schreiben und Vervollständigen von Code unterstützen – ein Mensch beschreibt sein Problem in natürlicher Sprache, das Sprachmodell von GitHub Copilot liefert darauf die passenden Codezeilen. Dazu muss das Tool allerdings trainiert werden. GitHub Copilot scannt daher die auf GitHub gespeicherten Repositories und verwendet die Codezeilen zur Lösung des Problems. Es handelt sich dabei um Programmiercode im Umfang von einigen Milliarden Zeilen.

Das Problem dabei: Nicht immer handelt es sich um frei verfügbaren, öffentlichen Code. Stattdessen laufen viele Projekte unter einer Copyleft-Lizenz wie etwa GPL, die verlangt, dass jede Änderung am Sourcecode unter der gleichen Lizenz verbreitet wird wie das Original. Oder die Software ist unter einer freizügigen Lizenz wie ISC veröffentlicht, die fordert, dass bei jeder Verwendung des Codes der Autorennamen und die Lizenz genannt werden.

GitHub Copilot jedoch kümmert sich nicht um die Urheberschaft oder um die Art der Lizenz. Damit besteht die Möglichkeit, dass der vom Programm präsentierte Code in einer Software rechtlich gesehen überhaupt nicht verwendet werden darf. Wer Software entwickelt, musste also befürchten, unbeabsichtigt in eine rechtlich schwierige Situation zu geraten. Codeberg und andere Entwicklungsplattformen bieten einen Ausweg, da sie ohne KI arbeiten und keine Codescans durchführen.

Vorbildliche Community

Trotz der Abgänge der letzten Jahre ist GitHub immer noch die bei Weitem größte Versionsverwaltung der Welt. Im Januar 2023 umfasste die Datenbank des Unternehmens rund 420 Millionen Repositories von mehr als 100 Millionen Beteiligten. Auf Codeberg dagegen waren bei Redaktionsschluss 236.772 Repositories von 157.048 Usern registriert. Die Community rund um GitHub ist also weitaus umfangreicher als die von Codeberg, und die Projekte können auf mehr Hilfe und Unterstützung zurückgreifen. Die Codeberg-Community hingegen gilt als vorbildlich hilfsbereit und freundlich, auch bei einfachen Projekten.

Wer Codeberg unterstützen will, muss allerdings mit seinem Projekt nicht gleich von GitHub umziehen. Codeberg bietet eine Mirror-Funktion an, sodass das Repository auf beiden Seiten vertreten sein kann. Eine Anleitung dazu gibt es unter https://codeberg.org/Recommendations/Mirror_to_Codeberg.

Die Funktionalität

Wie bereits erwähnt, bietet Codeberg gegenüber der zugrunde liegenden Software Forgejo einige zusätzliche Funktionen an. So gibt es z. B. Codeberg Pages (<https://codeberg.page/>), einen Webhosting-Dienst, für statische Websites mit einer benutzerfreundlichen Adresse wie etwa [Benutzername].codeberg.page.

Wie die meisten anderen Versionsverwaltungen auch hält Codeberg eine CI-Funktionalität (Continuous Integration) bereit. Das bedeutet, dass Codeänderungen regelmäßig und oft mehrere Male am Tag in ein gemeinsames Repository übertragen werden. Auf diese Weise lassen sich Integrationsprobleme bereits in einem frühen Stadium erkennen, und es ist weitgehend sichergestellt, dass die Codebasis stabil ist. In vielen Projekten ist CI die erste Station einer Pipeline, die das Projekt anschließend noch durch CD (Continuous Delivery) und CD (Continuous Deployment) führt. Codeberg greift dabei auf die Open-Source-Software Woodpecker CI zurück, die unter <https://ci.codeberg.org> erreichbar ist.

Schließlich gibt es bei Codeberg mit Weblate auch ein integriertes Open-Source-Lokalisierungstool, das Codeberg Translate heißt (<https://translate.codeberg.org>). Es bietet eine einfach zu bedienende Web-Oberfläche, die den in einer Software enthaltenen Text präsentiert und Leuten, die freiwillig helfen wollen, die Möglichkeit gibt, ihre Übersetzungen einzubringen. Weblate zeigt nicht den Code, sondern lediglich die Texte. So können auch Personen, die nicht programmieren, an einem Projekt mitarbeiten.

Anleitung zur Migration

Wer sich eine Zeit lang in Codeberg umgeschaut und beschlossen hat, die eigenen Repositories von GitHub oder einem anderen Dienst zu migrieren, wird den Umzug ausgesprochen einfach finden. Es genügt, ein neues Repository in Codeberg anzulegen. Dort findet sich dann eine Option namens „New Migration“. Sie leitet Schritt für Schritt durch den Prozess zur Übertragung der Software. Eine Anleitung gibt es unter <https://codeberg.org/r-codeberg/migration-guide>. In diesem Zuge empfiehlt es sich in vielen Fällen, einen Mirror des Codeberg-Repositories bei GitLab zu erzeugen.

Sobald die Migration abgeschlossen ist, sollte in das alte Repository eine Notiz eingefügt werden. Und noch ein Hinweis: Wer bislang GitHub Actions benutzt hat, sollte sich beim Mirroring auf GitHub darauf einstellen, dass sich die Funktion nicht nahtlos in Woodpecker CI abbilden lässt.

Weitere Informationen

Wer sich kontinuierlich über die weitere Entwicklung bei Codeberg informieren will, sollte regelmäßig den Blog des Projekts unter <https://blog.codeberg.org> aufsuchen. Dort veröffentlicht der Verein hinter Codeberg alle paar Wochen Ankündigungen und einen „Letter from Codeberg“ mit aktuellen Nachrichten, Ausblicken und Einschätzungen.

Eine höhere Frequenz haben die Toots bzw. Tröts von Codeberg auf der Social-Media-Plattform Mastodon (<https://social.anoxinon.de/@Codeberg>). Hier werden die User teilweise mehrmals am Tag über aktuelle Vorkommnisse und Hintergründe auf dem Laufenden gehalten.

Und schließlich gibt es auch bei der Chat-Plattform Matrix einen eigenen Bereich zu Codeberg. Über den Matrix-Client oder auch direkt im Browser können User an den Diskussionen rund um die Versionsverwaltung teilnehmen. Den Einstiegspunkt erreicht man über die Adresse <https://matrix.to/#/#codeberg-space:matrix.org>.

LEVEL UP YOUR CAREER!

Finde bei uns den IT-Job, der dein Game changed!

gamescom jobs & career

powered by  heise jobs

20. bis 24.08.2025 in Köln, Halle 10.1

- Über **30 TOP-Arbeitgeber** – entdecke spannende Jobs und Chancen!
- Große **Fun-Challenge** mit täglicher Preisverleihung und mega Gewinnen!
- Bühne – Insights** aus der Gaming-Szene und viele smarte Karriere-Tipps & Bewerbungs-Coachings für deinen nächsten Move!

UND VIELES MEHR!

Mehr Infos und das
gamescom-Ticket gibt es hier:



Partner:





> Was IT-Fachleute unersetzlich macht

KI-Modelle generieren nicht nur Texte, Bilder und Musik, sondern auch Programmcode. Für Software-Engineers ist das eine gute Entwicklung.

Roland Freist

In den vergangenen Jahren war die Softwareentwicklung von mehreren Seiten mit Horrorszenarien konfrontiert. Im Januar 2025 erklärte etwa Mark Zuckerberg: „Spätestens 2025 wird KI die Arbeit von Engineers der mittleren Ebene erledigen können, Code schreiben und potenziell Software-Developer ersetzen.“

Und bereits 2023 hatte OpenAI bekannt gegeben, dass es ChatGPT gezielt darauf trainiere, die Stellen in der Entwicklung eines Tages komplett zu ersetzen. Ebenfalls 2023 führte Google in einem Experiment ein Bewerbungsgespräch mit ChatGPT und fragte dessen Wissen über Softwareentwicklung ab. Ergebnis: Wäre der Chatbot ein Mensch, hätte Google ihm die Stelle eines Level-3-Software-Engineers angeboten. Einstiegsgehalt: 183.000 US-Dollar im Jahr.

Das Experiment wurde nicht nur mit ChatGPT, sondern auch mit dem australischen Chatbot Alphacode durchgeführt, der auf Softwareentwicklung spezialisiert ist. Beiden Systemen wurde anschließend die Frage gestellt, ob sie Software-Engineers ersetzen könnten. Interessanterweise verneinten beide. ChatGPT schrieb: „Es ist unwahrscheinlich, dass ChatGPT oder Alphacode Programmierende ersetzen werden“, denn beide seien „nicht in der Lage, das Fachwis-

sen und die Kreativität menschlicher Programmierfachleute vollständig zu ersetzen [...] Programmieren ist ein komplexes Gebiet, das ein gründliches Verständnis der Informatikprinzipien und die Fähigkeit zur Anpassung an neue technische Entwicklungen erfordert.“

Größerer Bedarf an Software

Ähnlich sehen das die Fachleute aus Fleisch und Blut. Zachary Tatlock, Professor für Computerwissenschaften an der University of Washington, prognostiziert in einem Interview mit dem amerikanischen Online-Magazin Business Insider sogar, dass KI zu einem wachsenden Bedarf an Programmierfachkräften führen wird. Er vergleicht die Situation heute mit dem Aufkommen der Bankautomaten in den 80er-Jahren. Damals glaubten alle, dass die Zahl der Beschäftigten in den Banken sinken würde, weil die Arbeit der Geldausgabe weitgehend entfiel. Doch das Gegenteil war der Fall, die Menge der Beschäftigten stieg zwischen 1980 und 2010 sogar an. Der Grund: Das Jobprofil veränderte sich, die Angestellten wechselten sozusagen vom Kassendienst in den Vertrieb, pflegten die Beziehungen zur Bankkundschaft, boten Kredite an und gaben Anlagetipps.

Eine ähnliche Entwicklung sieht Tatlock für die Softwareentwicklung voraus. Einfache Programmierarbeiten werden seiner Meinung nach in Zukunft von KI-Systemen erledigt. Die menschlichen Software-Engineers hingegen konzentrieren sich dann zunehmend auf das Prüfen von KI-generiertem Code und werden weniger eigenen Code schreiben. Gleichzeitig, so Tatlock, werde der Bedarf an Software in Zukunft sogar steigen. Die erhöhte Nachfrage werde sich nur bedienen lassen, wenn in der Softwareentwicklung zunehmend KI-Tools zum Einsatz kommen.

Informatik ist mehr als coden

Auch Kan Min-Yen, Informatikprofessor an der National University of Singapore, prognostiziert in einem Interview mit Business Insider einen steigenden Bedarf an Computerfachleuten. Er weist darauf hin, dass KI transformativ sei und in viele Lebensbereiche integriert werden müsse. Dafür brauche man spezialisierte Fachkräfte. Ums Programmieren gehe es dabei gar nicht so sehr, das könne oft auch eine KI erledigen. Doch das Schreiben von Code sei eben nur ein Teil der Aufgaben von Software-Engineers.

Stattdessen arbeiten Developer in den meisten Fällen an einem Ansatz, mit dem sie ein bestimmtes Problem lösen wollen. KI sei dabei letztlich nur ein weiteres Werkzeug für die Arbeit. Allerdings erfordere die Entwicklung und Verwendung von KI zusätzlich grundlegende Kenntnisse in Softwaretechnik, Datenmanagement und Sicherheit. Außerdem müssten Software-Engineers auch die Anforderungen von Design, Produktmanagement und Business verstehen und in funktionalen Code umsetzen.

Der gleichen Meinung ist der Informatikprofessor David Malan von der Harvard University. Zu einer ganzheitlichen Informatikausbildung gehört seiner Ansicht nach mehr als nur die Fähigkeit zum Coden. Wer eine solche Ausbildung durchlaufen habe, könne mithilfe von KI die eigene Produktivität deutlich steigern. Denn mit dem virtuellen Assistenten lassen sich in derselben Zeit erheblich mehr Funktionen in Programme implementieren und Fehler beheben.

Der Fachkräftebedarf bleibt hoch

Vergangenes Jahr befragte der Digitalverband bitkom 852 seiner Mitglieder nach der Zukunft der IT-Fachkräfte. Die Antworten fielen stark unterschiedlich aus: 15 % sahen aufgrund des vermehrten Einsatzes von KI einen Stellenabbau voraus. 20 % waren pragmatisch – in ihren Augen wird KI vor allem solche Stellen ersetzen, für die ohnehin keine Bewerbungen eingehen. Doch immerhin 38 % gaben an, dass KI einen zusätzlichen Bedarf an Fachkräften schaffen werde.

Rund ein Drittel (34 %) der Unternehmen zeigte sich in der Umfrage davon überzeugt, dass durch KI neue Berufsbilder entstehen, etwa im KI-Training oder im Prompt Engineering. Gleichzeitig erwarten jedoch 27 %, dass einzelne IT-Berufe künftig weitestgehend durch KI ersetzt werden. IT-Fachkräfte ohne Erfahrung im Umgang mit KI-Tools könnten laut 18 % der befragten Unternehmen schon bald kaum noch gefragt sein.

Für die wirtschaftliche Lage ist KI nach Auffassung der Unternehmen eine überwiegend positive Entwicklung. 44 % geben an, dass

KI die Produktivität in der IT steigern könne, indem Routineaufgaben automatisiert und Mitarbeitende entlastet würden.

KI verändert die Softwareentwicklung

KI verändert zwar die Softwareentwicklung und kann etliche Aufgaben automatisieren, doch ist es unwahrscheinlich, dass sie Software-Engineers vollständig ersetzen wird. KI wird die Fähigkeiten der Developer eher erweitern, eventuell den Schwerpunkt ihrer Arbeit verlagern und ihnen neue Möglichkeiten eröffnen. Im Einzelnen sehen die voraussichtlichen Änderungen so aus:

- Automatisierung: KI kann wiederkehrende Programmieraufgaben automatisieren. Dazu gehören u. a. das Generieren von Standardcode und eine einfache Fehlersuche. Insofern könnte sich der Personalbedarf in einigen Bereichen verringern.
- Produktivitätssteigerung: KI kann bei der Codeanalyse, beim Testen und bei der Restrukturierung von Code helfen und so die Softwareentwicklung effizienter machen.
- Verschiebung der Aufgabenbereiche: Die Nachfrage nach Software-Engineers könnte zurückgehen; stattdessen werden Unternehmen in Zukunft eher Leute für Softwarearchitektur, KI-Systemintegration und Programmieren suchen, die den von der KI generierten Code überwachen.
- Erhöhte Nachfrage nach speziellen Fähigkeiten: Die KI-gesteuerte Softwareentwicklung wird wahrscheinlich einen Bedarf an Fachleuten für KI-Integration, maschinelles Lernen und verwandte Bereiche erzeugen.

Verstand ist unersetzlich

Die wichtigste und oft schwierigste Aufgabe bei der Softwareentwicklung ist nicht das Schreiben von Code, sondern das Verständnis für ein Problem sowie das Suchen und Finden von Lösungswegen. Während der Code in Zukunft voraussichtlich immer öfter von KI-Modellen erzeugt werden wird, dürfte die Industrie zunehmend Software-Engineers benötigen, die mit menschlicher Intuition und Kreativität die zugrunde liegenden Strategien entwickeln.

Allerdings lässt sich auch der Code nicht komplett in die Obhut der Maschinen geben. Erfahrene Engineers müssen die Zuverlässigkeit, Genauigkeit und Sicherheit der Software überwachen und letztlich auch dafür die Verantwortung übernehmen.

Dennoch werden Unternehmen, um ihre Entwicklungskosten zu reduzieren, zunehmend bestrebt sein, so viel Arbeit wie möglich an die günstigeren KI-Modelle auszulagern und im Gegenzug Personal abzubauen. Diese Strategie kann jedoch bestenfalls zu einem Teilerfolg führen. Denn zum einen werden sich, wie gezeigt, die Aufgaben der Fachkräfte größtenteils einfach nur verlagern. Zum anderen müssen sich die Firmen dessen bewusst sein, dass es bei der Entwicklung immer wieder zu unerwarteten Problemen kommt, deren Lösung menschlichen Einfallsreichtum erfordert.

Es spricht nichts dagegen, KI als Tool einzusetzen. KI kann viele Arbeitsvorgänge beschleunigen und sie effektiver machen. Für die Kontrolle und den zielgerichteten Einsatz von KI werden jedoch nach wie vor menschliche Fachleute gebraucht.



> Wie produktiv ein Geniestreich ist

Eine Messung der Developer-Produktivität ist, wenn nicht unmöglich, zumindest komplex. Sie findet jedenfalls statt.

Dirk Bongardt

Produktivität soll messbar sein, und Zahlen suggerieren Kontrolle. In der Softwareentwicklung zerplatzt diese Illusion jedoch schnell. Metriken locken zwar mit Klarheit, scheitern aber oft an der Komplexität der Wirklichkeit. Der Versuch, Leistung zu erfassen, stößt auf Widerstand: auf Systemgrenzen, Teamdynamiken und menschliche Eigenheiten. Wo beginnt echte Produktivität, was bleibt bloße Aktivität?

Softwareentwicklung besteht aus Kreativarbeit, nicht aus Fließbandtätigkeit. Jede Aufgabe unterscheidet sich von der nächsten. Lösungen entstehen iterativ, oft durch Versuch und Irrtum. Was heute zehn Minuten dauert, verlangt morgen einen ganzen Tag. Metriken wie Codezeilen oder Commit-Zahlen greifen dabei in aller Regel zu kurz. Sie bewerten Aktivität, nicht Wirksamkeit.

Hinzu kommt, dass in der Entwicklung viele im Team arbeiten, die Leistung entfaltet sich im Zusammenspiel mit anderen. Wer Wissen teilt, Code gegencheckt oder Architekturentscheidungen vorbereitet, erzeugt oft keinen sichtbaren Output. Dennoch bleiben diese Arbeiten absolut wichtig. Klassische Produktivitätskennzahlen blenden solche Beiträge aber meist aus. So entsteht ein schiefes Bild der Wirklichkeit.

Technische Faktoren erschweren die Messung zusätzlich. Unterschiedliche Repositories, fragmentierte Toolchains oder fehlende Standards führen zu unvollständigen oder widersprüchlichen Daten. Auch moderne Systeme liefern oft nur aggregierte Zahlen ohne Kontext. Die Interpretation bleibt subjektiv. Automatisierung reduziert zwar den Messaufwand, schafft aber neue Probleme: Welche

der gemessenen Aktivitäten sind erfolgsentscheidend, welche bloß trivial? Messungen bleiben anfällig für Fehlrteile und technische Verzerrungen.

Vom Zeilenzählen bis zu Story Points

Frühe Ansätze zur Messung von Entwicklungsproduktivität setzten ganz auf Zählbares. Lines of Code galten lange als Maßstab: Mehr Code = mehr Leistung. Diese Sicht ignoriert jedoch Effizienz und Qualität. Viel Code kann auch schlecht strukturiert oder redundant sein. Ebenso galt die Anzahl der Commits als Indikator. Doch auch häufiges Speichern kleiner Änderungen – dessen Sinnhaftigkeit niemand leugnen wird – sagt wenig über den tatsächlichen Fortschritt aus.

Fehlerzahlen sind eine weitere klassische Metrik: Bugs pro Release. Weniger Fehler galten als Zeichen guter Arbeit. Doch dieser Wert hängt stark vom Testumfang und vom Risikoprofil der Anwendung ab. Ein komplexes System produziert naturgemäß mehr Fehler. Gleichzeitig belohnt diese Metrik defensiven Code statt mutige Architekturentscheidungen. Die Aussagekraft bleibt insofern begrenzt, besonders ohne Kontext.

Mit dem Aufkommen agiler Methoden hat sich der Blickwinkel verschoben. Statt isolierte Outputs zu zählen, richtet sich der Fokus auf den Fortschritt im Team. Story Points wurden eingeführt, um den geschätzten Aufwand einer Aufgabe zu bewerten – unabhängig von Codezeilen oder Komplexität. Diese Maßzahlen beruhen auf Teamkonsens und Erfahrungswerten. Sie machen vergleichbare Aussagen über die Geschwindigkeit innerhalb eines Teams, nicht zwischen Teams.

Velocity wurde nun zur zentralen Metrik. Diese Größe zeigt, wie viele Story Points ein Team pro Sprint abschließt. Der Vorteil liegt im Überblick über Trends – steigt oder sinkt die Produktivität? Doch auch diese Zahl kann täuschen. Änderungen in der Teamzusammensetzung oder in der Aufgabenstruktur beeinflussen sie stark. Zudem besteht die Gefahr, dass Teams Story Points bewusst hoch schätzen, um die Velocity stabil zu halten.

Burndown-Charts liefern ein visuelles Abbild des Fortschritts. Sie zeigen, wie viele Aufgaben noch offen sind, idealerweise sinkt die Kurve stetig. Die Aussagekraft hängt aber stark von der Datenqualität ab. Ungeplante Aufgaben oder unpräzise Schätzungen können das Bild verzerren.

Moderne Ansätze der Produktivitätsmessung

Aktuelle Tools zur Messung von Entwicklungsproduktivität basieren auf etablierten Frameworks wie DORA und SPACE. GitLab bietet z. B. integrierte DORA-Metriken an: Deployment-Frequenz, Lead Time for Changes, Change Failure Rate und Time to Restore Service. Diese Metriken helfen Teams, die Geschwindigkeit ihrer Softwarebereitstellung und deren Stabilität zu bewerten. Andere Tools wie LinearB, Swarmia und Sleuth integrieren sich in vorhandene Entwicklungsumgebungen und liefern Echtzeitdaten zu diesen Metriken; die Tools sollen z. B. Engpässe identifizieren und die Effizienz steigern.

CYBER- FORCE

Digitale Transformation der Bundeswehr vorantreiben.
Innovative Technologien entwickeln und bereitstellen.
Für stabile und leistungsfähige IT sorgen.

Was ist deine Mission?

BWIr dich jetzt!

bwi.de/professionals



◀ **Operation: Digitale Bundeswehr** ▶

ICIG BUSINESS SERVICES



Zur Verstärkung unseres stark wachsenden Bereichs **Information Management** suchen wir zum nächstmöglichen Zeitpunkt am Standort **Wuppertal** (nahe Düsseldorf und Köln) oder **Frankfurt am Main** einen

(Junior) IT Service Desk Agent (m/w/d)

Ihre Aufgaben

- Entgegennahme, Weiterleitung und Bearbeitung von IT-Anfragen und Störungen per Telefon, E-Mail und Ticketsystem
- Analyse und Lösung von IT-Problemen in den Bereichen Clients, Hardware, Software
- Dokumentation von Anfragen und Lösungen in unserem Ticketsystem Matrix42
- Unterstützung bei der Installation und Konfiguration von IT-Systemen und Anwendungen
- Zusammenarbeit mit anderen IT-Teams zur Lösung komplexer Probleme
- Schulung und Unterstützung der Endanwender bei IT-Fragen
- Dokumentation von Anfragen und Lösungen in unserem Ticketsystem Matrix42

Ihr Profil

- Abgeschlossene Ausbildung im Bereich Informatik oder ähnliche Vorkenntnisse als Quereinsteiger (m/w/d)
- Erste Berufserfahrung im IT-Support oder einer ähnlichen Position
- Gute Kenntnisse in den Bereichen Windows-Betriebssysteme, MS Office und im Umgang mit Ticketsystemen
- Analytische Fähigkeiten und Organisationstalent
- Ausgeprägte Kunden- und Serviceorientierung
- Sichereres persönliches Auftreten, Zuverlässigkeit und Teamfähigkeit, gepaart mit einem hohen Maß an Engagement und Leistungsbereitschaft
- Gute Deutsch- und Englischkenntnisse

Unser Angebot

- Weiterentwicklung ► Schulungsmaßnahmen ► Flexibles Arbeiten (Mobile Office, Gleitzeit, 30 Tage Urlaub + 10 Tage Workation) ► Flache Hierarchien ► Kostenlose Fix- oder Weithub-Mitgliedschaft oder Bezuschussung von EGYM Wellpass ► JobRad ► Mitarbeiter-PC-Programm (MPP) ► Pluxee-Gutscheine ► Betriebliche Altersvorsorge ► Corporate Benefits ► Attraktive Lage mit modernen Büroräumen ► Kostenlose Parkmöglichkeiten oder Übernahme der Kosten des Deutschland-Tickets

Die **ICIG Business Services GmbH** ist die interne Dienstleistungs- und Beratungsgesellschaft der International Chemical Investors Group (ICIG). Von unseren Standorten in Wuppertal und Frankfurt aus betreuen und beraten wir die weltweiten Gesellschaften der ICI Group in den Bereichen IT, Human Resources, Legal sowie Accounting und Finance.

Klingt dies genau nach Ihnen? Wir freuen uns von Ihnen zu lesen!

Senden Sie uns gerne Ihre aussagekräftigen Bewerbungsunterlagen (Ansreiben, Lebenslauf, Zeugnisse) zu. Bei Fragen rund um die Bewerbung erreichen Sie uns per E-Mail an Annika.Doerflinger@icig-bs.com.

Mehr über ICIG Business Services und die ICI Group erfahren Sie im Internet unter: www.icig-bs.com

Das SPACE-Framework erweitert den Blick auf Aspekte wie Zufriedenheit, Kommunikation und Zusammenarbeit. Tools wie Flowtrace analysieren Kommunikationsmuster, Unterbrechungen und Fokuszeiten und wollen Einblick in die Teamdynamik geben. Awesome Graphs for Bitbucket ermöglicht die Visualisierung von Commit-Aktivitäten und Code-Reviews. Damit lassen sich Aktivität und Zusammenarbeit unter Umständen besser bewerten. Insgesamt sollen diese Tools sowohl quantitative als auch qualitative Aspekte der Produktivität erfassen und gezielte Verbesserungen ermöglichen.

Viele Unternehmen nutzen Metriken, um daran Trends abzulesen, nicht um Einzelne zu bewerten. Sie identifizieren Engpässe, verbessern Abläufe und sorgen für Transparenz. Wer dagegen versucht, individuelle Leistung aus Zahlen abzuleiten, übersieht die Zusammenhänge. Denn meist kommt es auf den Kontext an: Ein stabil laufendes System erzeugt weniger sichtbare Aktivität – das kann positiv sein. Und: Metriken zeigen in der Regel das, was ist. Was fehlt, bleibt dagegen oft verborgen.

Was die Zahlen verschweigen

Zwei Beispiele: Das Dev-Team bei einem großen E-Commerce-Anbieter hat durch Pair Programming die durchschnittliche Durchlaufzeit von Tickets um 20 % reduziert – sichtbar in LinearB. Ein anderes Team bei Google hat anhand von DORA-Metriken die Release-Zyklen halbiert. In beiden Fällen haben Zahlen geholfen, Muster zu erkennen. Doch die Veränderungen waren letztlich Teamentscheidungen, keine Vorgaben von oben. Bei erfolgreichen Teams werden Metriken zur Selbststeuerung genutzt, nicht zur Kontrolle.

Falsch angewandtes Entwicklungscontrolling kann im schlimmsten Fall die Produktivität eigentlich starker Teams schmälern, etwa dann, wenn die Teams die Codequalität vernachlässigen und stattdessen auf schnellere Ergebnisse setzen. Um die Metriken zu verbessern, steigen die Leute auf einfache oder repetitive Aufgaben um, statt komplexe Probleme zu lösen.

Ihren absurden Höhepunkt erreicht diese Logik, wenn gezielt nur auf Metriken gespielt wird (Gaming the System), etwa durch übertriebene Schätzungen oder künstlich häufige Commits. Solche Prak-

tiken verzerren das Bild der Produktivität und verhindern echte Fortschritte. Die Folgen sind sinkende Motivation, Qualitätsverlust und ineffiziente Prozesse.

Gerade die wertvollste Arbeit entzieht sich regelmäßig der quantitativen Erfassung. Ein guter Einfall kann tagelange Arbeit ersetzen – lässt sich aber kaum vorab bewerten. Manche Fachleute lösen komplexe Probleme mit einer eleganten Zeile Code, andere stabilisieren ein System durch präzise, weitsichtige Architekturentscheidungen. Solche Beiträge tauchen selten in Metriken auf. Sie wirken unschwellig, entfalten ihren Wert erst im Zusammenspiel. Wer nur Zählbares misst, verpasst genau jene Impulse, die Projekte entscheidend voranbringen.

2004 entwickelte ein einzelner Google-Entwickler die erste Version von Gmail – als internes Projekt. Der Mehrwert zeigte sich erst Jahre später und wäre mit klassischen Metriken kaum darstellbar. Auch bei Facebook entstand der Like-Button durch eine kleine Gruppe, deren Impact zunächst unterschätzt wurde. In der Open-Source-Welt machte Linus Torvalds' erster Kernel kaum Eindruck – bis sich das System dann doch rasch verbreitete. Solche Leistungen erscheinen in keiner Velocity-Statistik, schaffen unter Umständen aber ganze Welten.

Zahlen und Zwischentöne

Typische Metriken betreffen die Commit-Frequenz, die Dauer von Code-Reviews, Ticket-Durchlaufzeiten und die Beteiligung an Pull Requests. Sie lassen Rückschlüsse auf Beteiligung und Aktivität zu, aber nicht unbedingt auf die Qualität. Wer programmiert, sollte in jedem Fall wissen, welche Zahlen im eigenen Unternehmen erhoben werden. Einmal mehr liegt der Schlüssel zum Erfolg in transparenter Kommunikation. Gute Teams diskutieren offen, welche Metriken sinnvoll sind – und wo deren Grenzen liegen.

Im besten Fall können die Teams die Metriken aktiv mitgestalten. Denn wer versteht, was gemessen wird und warum, kann besser mit den Daten umgehen. Tipp: Fragen stellen, Visualisierungen prüfen, Auffälligkeiten ansprechen. Zahlen sind Gesprächsanlässe, keine Urteile.

Impressum CodingCareers

Redaktion: just 4 business GmbH,
Telefon: 0511 5352-826

E-Mail: tj@just4business.de

Verantwortliche Redakteure: Thomas Jannot (v.i.S.d.P.),
Ralph Novak; Florian Eichberger (Lektorat)

Autoren: Dirk Bongardt, Roland Freist

Layout und Satz: Matthias Timm,
Heise Medienwerk GmbH & Co. KG, Rostock

Titelbild: @uuhouse

Verlag und Anzeigenverwaltung:

Heise Medien GmbH & Co. KG, Postfach 61 04 07,
30604 Hannover; Karl-Wiechert-Allee 10, 30625 Hannover;
Telefon: 0511 5352-395, Fax: 0511 5352-129

Anzeigenleitung: Michael Hanke -167,

E-Mail: michael.hanke@heise.de,

<https://mediadaten.heise.de/heise-medien-2/mediadaten>

// heise devSec()

Die Konferenz für sichere Softwareentwicklung

30. September und 1. Oktober 2025 • Regensburg

Aus dem Programm:

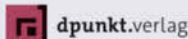
- Software nachhaltig sicher entwickeln – ein Praxisbericht
- Software-Supply-Chain-Security: Mehr als nur Dependency-Management
- LLMs im Secure Dev Lifecycle
- Cyber Resilience Act – Cheatsheet für Entwickler
- Fünf Scanner, drei Dashboards, null Überblick? Zeit für eine ASPM-Lösung?
- Protectors of the Realm: Wie man einen Keycloak sicher hält

Jetzt
Frühbucher-
tickets
sichern!

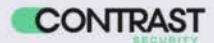
Workshops zu sicherer Legacy-Software sowie OAuth 2.0 & Open ID Connect

heise-devsec.de

Veranstalter



Gold-Sponsoren



Web Security Day

Der Sicherheits-Deep-Dive
für JavaScript-Entwickler:innen

9. Oktober 2025 • Online

Webentwicklung – aber sicher

- OWASP Top Ten 2025: die Neuerungen im Überblick
- KI-basierte Angriffe auf Webanwendungen abwehren
- Passkeys einsetzen und Phishing verhindern
- Automatisierte Sicherheitsprüfungen mit GitHub Advanced Security
- Mutation XSS – die dunkle Seite von HTML verstehen

Jetzt
Frühbucher-
tickets
sichern!

Workshop zu Web Security in der Praxis

enterjs.de/security.php

Veranstalter

