

GitHub Copilot

Tipps & Tricks Vol. 2

Best Practices für Entwickler*innen zum produktiveren Arbeiten mit GitHub Copilot



Vorwort

In meiner Zeit bei Microsoft Deutschland durfte ich schon einige E-Books begleiten. Das erste E-Book – wenn ich mich richtig erinnere – war das Visual Studio Inselbuch 2013 mit Beiträgen zu Visual Studio, Team Foundation Server, Windows 8 und Windows Azure (wie es damals noch hieß). Danach folgten weitere E-Books zu Visual Studio, Visual Studio Code oder auch zur effizienten Remote-Zusammenarbeit von Entwicklungsteams kurz nach dem Ausbruch von Covid. Für letzteres E-Book hat damals noch Thomas Dohmke einen Beitrag über den Kulturwandel bei GitHub geschrieben. Inzwischen ist er CEO von GitHub und der nächste Wandel ist bereits in vollem Gange – die Rede ist hier natürlich von GitHub Copilot.

Bereits im Jahr 2015 hat Satya Nadella, CEO von Microsoft, gesagt, dass jedes Unternehmen ein Software-Unternehmen sein wird. In ähnlicher Form hat Satya Nadella die Aussage bis heute wiederholt. Dabei geht es im Kern immer darum, Entwicklungsteams oder auch sogenannte Citizen Developer zu befähigen, ihre Kreativität bestmöglich einsetzen zu können, um Probleme zu lösen sowie neue Produkte und Lösungen zu schaffen, die einen Mehrwert darstellen. Im Vordergrund steht also zum einen die Befähigung, mit Technologie Neues zu erschaffen, und zum anderen die Konzentration auf das Wesentliche.

Und genau hier setzt GitHub Copilot an. Um Ihnen dieses neue Angebot aus der Perspektive von Nutzer*innen näher zu bringen, haben wir uns entschlossen, mal wieder ein E-Book zu veröffentlichen mit Tipps und Tricks zu GitHub Copilot aus dem echten Leben. Die Idee ist, das E-Book nach und nach mit neuen Tipps und Tricks zu erweitern. Falls Sie selber einen Beitrag in einer der nächsten Ausgaben beisteuern möchten, dann schreiben Sie mir gerne eine E-Mail an techwiese@microsoft.com. Ich würde mich sehr freuen.

Bis dahin wünsche ich Ihnen viel Spaß bei der Lektüre der ersten Ausgabe dieses E-Books.

Ihr Dennis Gassen

Malte Lantin

Als Principal Solutions Engineer bei GitHub unterstützt Malte Lantin strategische Kunden in Europa bei der Einführung von GitHub Enterprise als zentrale Plattform für die sichere Entwicklung und Bereitstellung von Software.



Einleitung: Die Zukunft der Softwareentwicklung mit GitHub Copilot

Von Malte Lantin

Nur knapp 3 Jahre nach dem ersten internen GitHub Whitepaper zu KI-gestützter Softwareentwicklung im Jahr 2020 und der Veröffentlichung von GitHub Copilot im Jahr 2022 hat sich GitHub Copilot heute als Standard für viele Entwickler*innen und Unternehmen etabliert. Dabei gerät fast in Vergessenheit, dass KI-Unterstützung in der Softwareentwicklung noch vor kurzem von vielen als Fiktion wahrgenommen wurde. GitHub Copilot stellt somit einen Wendepunkt in der Entwicklung von Softwareentwicklungstools dar. Mit kontinuierlichen Verbesserungen und speziell für den professionellen Einsatz in Unternehmen entwickelten Funktionen ist GitHub Copilot heute bereits ein kaum mehr wegzudenkendes Standardwerkzeug. Diese Einführung zeigt die grundlegende Funktionsweise und die Mehrwerte, die in den folgenden Kapiteln anhand zahlreicher Fallbeispiele veranschaulicht werden.

Technologische Innovationen und Funktionen

GitHub Copilot basiert auf mehrjähriger Forschung im Bereich Sprachmodelle, deren sichere Bereitstellung in den Azure OpenAI Services sowie der nahtlosen Integration in den Entwicklungsprozess. Entwickler*innen steht damit heute in Visual Studio Code, Visual Studio, den JetBrains IDEs sowie weiteren Editoren eine KI-basierte Codevervollständigung zur Verfügung, die ihren Kontext aus den aktuell in Bearbeitung befindlichen Quellcodedateien sowie Informationen aus dem jeweiligen Editor bezieht. Dadurch kann sie im laufenden Entwicklungsprozess passende und hochrelevante Codevorschläge generieren. Für diese Codevervollständigung wird das Fill-in-the-Middle (FIM) Paradigma verwendet, das es ermöglicht, jederzeit Vorschläge zu liefern, die bestmöglich an den aktuellen Kontext angepasst sind und den Stil des jeweiligen Projektes berücksichtigen.

Ende 2023 wurden mit der Einführung von GitHub Copilot Chat die Möglichkeiten der KI-gestützten Entwicklung nochmals deutlich erweitert. Über die in die Entwicklungsumgebung integrierte Chatoberfläche ist es nun unter anderem möglich, komplexere Anweisungen zu geben, weitergehende Empfehlungen zu erhalten, Codeerklärungen abzurufen, Tests generieren oder ganze Arbeitsumgebungen von GitHub Copilot erstellen zu lassen. Durch den iterativen Ansatz basierend auf natürlicher Sprache ist Copilot Chat leicht zugänglich und hilft allen Entwickler*innen schneller Software zu entwickeln, Probleme zu beheben und neue Technologien zu erlernen. Dabei ist das Kontextwissen, das GitHub Copilot aus dem Code-Editor gewinnt, ein wesentliches Unterscheidungsmerkmal zu anderen Chat-basierten Sprach-KIs. GitHub Copilot Chat ist dazu in der Lage, auf das Wissen von angeforderten Dateien oder Aussagen aus der Kommandozeile zurückzugreifen und auf diese Weise jederzeit passgenaue Vorschläge zu liefern.

GitHub Copilot wird zudem kontinuierlich erweitert und steht ab Frühjahr 2024 auch als Erweiterung der GitHub CLI Tools zur Verfügung, um Befehle auf der Kommandozeile zu erklären oder zu generieren. Seit Februar 2024 haben Unternehmen, welche die GitHub Enterprise Cloud verwenden, außerdem die Möglichkeit mit GitHub Copilot Enterprise von einer noch tieferen Integration von GitHub Copilot in die eigene Entwicklung zu profitieren. Dies erfolgt über eine Integration von GitHub Copilot Chat mit der GitHub-Weboberfläche, eine KI-gestützte Bearbeitung von Pull-Request, Zugriff auf das Wissen der eigenen Quellcode-Repositories und Dokumentation, sowie einer zukünftigen Anpassung der KI-Modelle an die eigene Codebasis. Hiermit werden die Teile des Entwicklungsprozesses, welche sich außerhalb der Entwicklungsumgebung befinden, sinnvoll durch GitHub Copilot unterstützt.

Auswirkungen auf die Produktivität und Codequalität

Mit über drei Milliarden generierten Codezeilen hat GitHub Copilot seine Wirksamkeit und weitreichende Akzeptanz in der Entwicklergemeinschaft bereits eindrücklich bewiesen. Heute benutzen mehr als 1,8 Millionen Softwareentwickler*innen GitHub Copilot und mehr als 50.000 Unternehmen haben sich für eine Einführung der Technologie entschieden. Der Einfluss von GitHub Copilot auf die Produktivität der Entwickler*innen ist sowohl quantifizierbar als auch erheblich. In einer Studie, wurde festgestellt, dass mehr als 30 Prozent der Vorschläge von GitHub Copilot akzeptiert werden, was die Relevanz und Nützlichkeit in der täglichen Entwicklung unterstreicht. Mit einer Beschleunigung der Arbeit um bis zu 55 Prozent ist der Mehrwert offensichtlich. Beinahe wichtiger ist jedoch die Tatsache das Entwickler*innen sich bei einem Einsatz von GitHub Copilot auch in Befragungen als zufriedener und produktiver beschreiben. Zudem zeigen Studien, dass die Codequalität steigt und Code Reviews als zielführender und schneller beschrieben werden. Entwickler*innen berichten, dass sie mit GitHub

Copilot länger konzentriert arbeiten können, da Kontextwechsel und Ablenkung vermieden werden. Die deutlich verbesserte Entwicklungserfahrung bietet Vorteile über Produktivitätssteigerungen hinaus. Indem GitHub Copilot routinemäßige und repetitive Aspekte automatisiert, wird es den Entwickler*innen ermöglicht, sich auf komplexere und kreativere Aspekte ihrer Projekte zu konzentrieren, was die Arbeitszufriedenheit erhöht.

Fazit

Zusammenfassend stellt GitHub Copilot einen bedeutenden Meilenstein in der Geschichte der Softwareentwicklung dar. Es symbolisiert den Beginn einer neuen Ära der KI-unterstützten Programmierung, die die Produktivität der Entwickler*innen erheblich verbessert, die Codequalität erhöht und das allgemeine Wohlbefinden der Entwickler*innen positiv beeinflusst. Da GitHub Copilot kontinuierlich weiterentwickelt wird und sich in Zukunft noch stärker in den Softwareentwicklungsprozess integriert, wird sein Einfluss weiter wachsen und GitHub Copilot damit ein unverzichtbares Werkzeug im Softwareentwicklungsbereich werden.

In den folgenden Kapiteln werden Einsatzmöglichkeiten beschrieben und damit der praktische Nutzen im Alltag verdeutlicht. Wir wünschen viel Spaß beim Lesen und dem Einsatz von GitHub Copilot.

Weiterführende Informationen:

- [Copilot transforms GitHub into the AI-powered developer platform](#)
- [The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot](#)
- ¹ [Quantifying GitHub Copilot's impact on developer productivity and happiness](#)
- ² [Quantifying GitHub Copilot's impact on code quality](#)

Inhalt

Vorwort	2
Einleitung: Die Zukunft der Softwareentwicklung mit GitHub Copilot	3
GitHub Copilot: Versionen und Features im Überblick	7
GitHub Copilot im Einsatz	11
App-Migration zwischen Programmiersprachen mit GitHub Copilot Chat	13
Nie mehr Lorem Ipsum	20
Intelligentes Coding mit GitHub Copilot	24
Von der Idee zum Deployment in unter 30 Minuten	30
Tests schreiben mit GitHub Copilot Chat	34
Proaktive Fehlerbehebung auf Windows-Geräten	36
Wie war der Code doch gleich?	40
Automatisierung von nervigen Entwicklungsarbeiten mit GitHub Copilot Chat	42
Bessere Prompts für besseren Code – Tipps & Tricks für Developer	47
Aussagekräftige Funktionsnamen & Variablen	50
Unterstützung bei der Testfallerstellung mit GitHub Copilot	52
Wie wir drei Monate die Möglichkeiten von GitHub Copilot getestet haben	55
GitHub Copilot als KI-Assistent für barrierefreie Software	60
Von der Anforderung zur Komponente	70
Einsatz einer Chat-Syntax: Chat-Teilnehmer, Chat-Variablen, Slash-Befehle	76
Nutzen Sie GitHub Copilot für Ihre Projekte	81
Weiterführende Ressourcen	82