

Copilot Studio: Erweiterte Custom GPTs

Du baust individuelle Copilots für deine Teams schneller auf, GitHub-Copilot-Integrationen erleichtern die Entwicklung — und nebenbei verschiebt Microsoft die Grundlagen deiner KI-Strategie unter dir weg.

Executive Summary

Microsoft hat Copilot Studio 2026 umgebaut, und zwar nicht kosmetisch. Unter jedem Agenten läuft jetzt eine sogenannte Harness: eine Laufzeitumgebung zwischen deinem Entwurf und dem Sprachmodell. Sie entscheidet, wann das Modell gerufen wird, was es zu sehen bekommt, wie sie die Antwort interpretiert und welches Werkzeug danach anspringt. Drei Varianten stehen zur Wahl, und die Entscheidung fällt beim Anlegen des Agenten. Ein Agent auf der GitHub-Copilot-Harness lässt sich später nicht in die Standard-Harness überführen — und umgekehrt genauso wenig. Willkommen bei der ersten Architekturentscheidung, die dein Fachbereich trifft, ohne zu wissen, dass er eine trifft.

Die GitHub-Copilot-Harness ist das Schaustück. Sie nimmt ein Ziel entgegen, zerlegt es selbst in Schritte, ruft Werkzeuge über Konnektoren, Knowledge, MCP und verbundene Agenten, erzeugt und bearbeitet Word-, Excel-, PowerPoint- und PDF-Dateien nativ, merkt sich Dinge über Gespräche hinweg und sucht sich einen anderen Weg, wenn ein Schritt scheitert. Jede Aufgabe läuft in einer von Copilot Studio kontrollierten Sandbox. Statt Topics und Verzweigungslogik schreibst du Anweisungen in normalem Deutsch.

Der Haken hängt am Geldbeutel und an der Governance. Abgerechnet wird in Copilot Credits, und der Zähler startet nicht mehr bei der Veröffentlichung, sondern beim allerersten Prompt im Baukasten. Bauen, Vorschau, Testen, Evaluieren: alles kostet. Wer das nicht vorher regelt, erklärt hinterher der Buchhaltung, warum ein Prototyp, den nie ein Anwender gesehen hat, vierstellig zu Buche schlägt.

FAKTEN | Die drei Harnesses in einem Satz

GitHub-Copilot-Harness für mehrstufige Geschäftsprozesse mit eigenständigem Schlussfolgern, Standard-Harness für regelbasierte Agenten mit vorhersagbarem Verhalten, Copilot-Chat-Harness zum Andocken von Unternehmenswissen an Microsoft 365 Copilot Chat.

Worum geht es im Detail?

Der Begriff „Custom GPT“ ist Marktsprache. Microsoft nennt das Ding Agent, und die eigentliche Neuerung steckt eine Ebene tiefer. Bisher hast du in Copilot Studio Themen definiert, Auslöser gepflegt, Verzweigungen gezogen und dich anschließend gewundert, warum der Bot bei der dritten Umformulierung einer Frage in den Fallback rutscht. Diese Arbeit übernimmt jetzt eine verbesserte Orchestrierung. Du beschreibst in der Registerkarte „Build“, wer der Agent ist, was er weiß, was er darf und wo seine Grenzen liegen. Den Rest baut die Laufzeitumgebung.

Die Oberfläche ist entsprechend aufgeräumt: Build, Preview, Evaluate, Monitor. Vier Reiter, ein Agent. Auf Build liegen Anweisungen, Knowledge, Werkzeuge, Skills, Modellauswahl, verbundene

Agenten und der Memory-Schalter. Preview ist der Testchat, Evaluate die strukturierte Qualitätsprüfung, Monitor zeigt Aufgaben, angefasste Dateien, Aktivität und den Credit-Verbrauch des Agenten.



Abbildung 1: Die Harness ist die Laufzeitschicht zwischen deinem Entwurf und dem Modell.

Besonders hübsch ist der Einstieg über natürliche Sprache, derzeit als produktionsreife Vorschau. Du tippst hin, was du automatisieren willst – etwa das komplette Onboarding neuer Kollegen inklusive Willkommensmails, Kontenanlage, Schulungen und Nachfassen bis zum letzten offenen Punkt. Copilot Studio entscheidet selbst, ob daraus ein deterministischer Workflow, ein dialogfähiger Agent oder beides wird, stellt Rückfragen und baut die Artefakte live im Seitenbereich zusammen. Ein Detail für die Datenschutzabteilung: Diese Erfahrung nutzt gegebenenfalls Anthropic-Modelle, wenn dein Administrator externe Sprachmodelle für generative Antworten freigegeben hat. Wer bisher davon ausging, dass in einem Microsoft-Produkt ausschließlich Microsoft-Modelle rechnen, sollte diesen Satz zweimal lesen.

WICHTIG | Testlizenz ist eine Sackgasse

Mit der Trial-Lizenz darfst du Agenten bauen und im Testchat ausprobieren, aber nicht veröffentlichen. Für den Piloten mit echten Anwendern brauchst du eine Copilot-Studio-Lizenz oder eine Microsoft-365-Copilot-Lizenz. Plane das ein, bevor der Fachbereich am Freitagnachmittag den Go-live ankündigt.

Zwei Bausteine sind neu genug, dass sie in kaum einem Architekturbild deines Hauses auftauchen. Erstens Skills: wiederverwendbare Fähigkeiten, definiert als Markdown mit YAML-Kopf für Name und Beschreibung. Die Orchestrierung aktiviert eine Skill, wenn die Anfrage zu ihrer Beschreibung passt. Als ZIP-Paket mit einer SKILL.md plus Skripten, Vorlagen und Referenzdokumenten wandert sie zwischen Agenten und zwischen Abteilungen. Das ist praktisch – und es bedeutet, dass Agentenverhalten künftig als Datei durch dein Unternehmen spaziert. Wer reviewt das eigentlich?

Zweitens Memory, ebenfalls produktionsreife Vorschau. Jeder Anwender bekommt einen eigenen Speicher in von Microsoft verwaltetem Storage. Der Agent schreibt Präferenzen und Kontext hinein und liest beim nächsten Mal wieder daraus. Die Regeln sind erfreulich klar: Erinnerungen sind privat, der Ersteller des Agenten sieht sie nicht, nach 28 Tagen ohne Interaktion löscht das System sie, in Gruppenchats und Teams-Kanälen ist Memory abgeschaltet. Anwender können über ein Portal alles einsehen und löschen. Ein Fallstrick bleibt: Wenn du Memory ausschaltest, verschwinden die gespeicherten Daten nicht — der Agent benutzt sie nur nicht mehr.

Baustein	Wofür er da ist	Wo er verwaltet wird
Anweisungen	Identität, Tonalität, Umfang, Verhalten	Identitätskonfiguration im Reiter Build
Knowledge	Daten, auf die sich Antworten stützen	Knowledge-Quellen, Auswahl durch die Orchestrierung
Werkzeuge	Aktionen gegen externe Dienste	Konnektoren, APIs, MCP-Server
Skills	Wiederverwendbare, aufgabenspezifische Modi	Markdown-Dateien oder ZIP-Pakete
Memory	Kontext über Gespräche hinweg	Pro Anwender, von Microsoft verwalteter Speicher

Die Qualitätssicherung heißt Evaluate und ist ehrlicher, als es der Marketingtext vermuten lässt. Du legst Testgespräche an — von Hand, per KI generiert oder als CSV-Import —, bündelst sie zu einer Evaluation, wählst eine Testmethode und ein Benutzerprofil und lässt laufen. Die einzige derzeit verfügbare Methode heißt „General quality“ und bewertet KI-gestützt Relevanz und Vollständigkeit. Sie vergleicht ausdrücklich nicht mit erwarteten Antworten. Wer also eine belastbare Regressionsprüfung für Fachlogik braucht, baut die weiterhin selbst.

Was sind Chancen? Was sind Risiken?

Die Chance ist real und nicht klein. Ein Prozess wie die Rechnungsprüfung — Beleg lesen, gegen die Bestellung abgleichen, Abweichungen zur Freigabe leiten — ist in der alten Welt ein Projekt mit Konnektoren, Flows und viel Verzweigungslogik. Auf der GitHub-Copilot-Harness ist es eine Beschreibung, ein paar Werkzeuge und mehrere Runden Feinschliff. Dass der Agent Office- und PDF-Dateien nativ erzeugt und bearbeitet, spart genau die Bastelei, die bisher jedes zweite Projekt aufgehalten hat.

Der Zähler läuft ab dem ersten Satz, nicht ab Inbetriebnahme

Copilot Credits decken LLM-Tokens, Werkzeuge inklusive Knowledge und MCP – und die Harness selbst.

STANDARD-HARNESS

kostenfrei bauen und testen

ab hier wird gezählt

GITHUB-COPILOT-HARNESS

verbrauchs basiert – vom allerersten Prompt an



ZWEI WEGE ZUM CREDIT

Prepaid-Paket

25.000 Credits für 200 US-Dollar im Monat.

Nicht Verbrauchtes verfällt zum Monatsende.

Pay-as-you-go über Azure-Abo: 0,01 US-Dollar je Credit, ohne Vorabbindung, dafür ohne Deckel nach oben.

WENN DAS BUDGET LEER IST

Der Agent antwortet nicht mehr.

Und du kannst nicht mehr per natürlicher Sprache bauen, nicht mehr testen, keine Evaluationen erzeugen. Microsoft lässt eine gewisse Überziehung zu – verlassen würde ich mich darauf nicht.

Abbildung 2: Bei der neuen Harness zählt schon die Entwurfsphase — anders als bisher gewohnt.

Das Risiko ist ebenso real und kommt in drei Geschmacksrichtungen. Die erste ist finanziell. Credits decken LLM-Tokens, Werkzeuge einschließlich Knowledge und MCP sowie die Harness selbst. Das Prepaid-Paket umfasst 25.000 Credits für 200 US-Dollar im Monat, Nichtverbrauchtetes verfällt zum Monatsende. Pay-as-you-go über ein Azure-Abo kostet einen Cent je Credit und hat dafür keinen Deckel nach oben. Ist die Umgebung leer, antwortet der Agent nicht mehr, und die Erbauer können weder bauen noch testen noch evaluieren. Microsoft lässt eine gewisse Überziehung als Karenz zu. Darauf zu planen ist ungefähr so klug, wie den Dispo als Liquiditätsreserve zu verbuchen.

WARNUNG | Der Prototyp, der niemandem gehört

Ein Fachbereich baute in drei Wochen sieben Agenten, von denen keiner je veröffentlicht wurde. Die Rechnung kam trotzdem, denn jeder Testchat und jede generierte Evaluation kostet Credits. Die anschließende Frage „Wer hat das genehmigt?“ hatte übrigens keine Antwort — der Erbauer war zwischenzeitlich in einem anderen Projekt. Kostenstelle: unbekannt. Verantwortung: vollständig diffus.

Die zweite Geschmacksrichtung ist die Zerfaserung. Verbundene Agenten, teilbare Skills und Werkzeuge über MCP ergeben schnell ein Geflecht, das niemand mehr im Kopf hat. Wenn Agent A an Agent B delegiert, der eine Skill nutzt, die ein Kollege vor einem halben Jahr als ZIP von einem Kollegen bekam, dann ist die Frage nach der Verantwortung für eine falsche Auskunft keine juristische Spitzfindigkeit mehr, sondern das Protokoll deiner nächsten Krisensitzung.

Die dritte ist die Vorhersagbarkeit. Eine Harness, die selbst entscheidet, welches Werkzeug sie ruft und wie sie einen gescheiterten Schritt umgeht, ist genau deshalb stark — und genau deshalb schwer zu auditieren. Für alles mit Rechtsfolgen bleibt die Standard-Harness mit ihren definierten Pfaden das ehrlichere Werkzeug. Kein Rückschritt, sondern eine Zuordnung nach Risiko.

Szenario	Empfohlene Harness	Begründung
Rechnungsprüfung, Onboarding, mehrstufige Fallbearbeitung	GitHub Copilot	Ziel statt Skript, Werkzeugketten, Dateiausgabe, Selbstkorrektur
Auskunftsagent im Servicedesk mit festen Antworten	Standard	Vorhersagbar, prüfbar, günstiger im Dauerbetrieb
Regulatorisch heikle Auskunft mit Rechtsfolge	Standard	Definierte Pfade, nachvollziehbare Entscheidungen
Unternehmenswissen in Microsoft 365 Copilot Chat	Copilot Chat	Kein Kontextwechsel, meist von der M365-Lizenz gedeckt

Was müssen wir jetzt schon vorbereiten?

Fang nicht mit dem Agenten an, sondern mit dem Rahmen. Die Werkzeuge dafür sind vorhanden, sie liegen nur in fünf verschiedenen Konsolen und gehören fünf verschiedenen Teams.



Abbildung 3: Governance entsteht aus fünf Schichten — und scheitert regelmäßig an ihren Nahtstellen.

Das Wichtigste zuerst: Klär die Credits, bevor jemand den ersten Prompt tippt. Der Verbrauch wird pro Umgebung zugeteilt und gesteuert, im Copilot Hub des Power Platform Admin Center lassen sich Obergrenzen für Pay-as-you-go setzen. Der Agent Usage Estimator von Microsoft liefert dir eine Hausnummer, bevor du eine Freigabe erbittest. Eine Hausnummer, wohlgemerkt, keine Zusage.

TIPP | Zwei Umgebungen, klare Kante

Richte eine Spielwiese mit knapp bemessenem Credit-Kontingent für die Erbauer ein und eine produktive Umgebung mit engen Datenrichtlinien und benannten Verantwortlichen. Environment Routing schickt neue Erbauer automatisch in die Spielwiese, statt in die Standardumgebung, in der erfahrungsgemäß alles landet, was später niemand mehr anfassen will.

Danach die Datenrichtlinien. Im Power Platform Admin Center steuerst du Authentifizierung für Erbauer und Anwender, zugelassene Knowledge-Quellen, Aktionen, Konnektoren und Skills, HTTP-Aufrufe, die Veröffentlichung in Kanäle, die Anbindung an Application Insights sowie Trigger für autonom laufende Agenten. Genau diese Trigger sind der Punkt, an dem aus einem Chatbot ein Prozessteilnehmer wird, der ohne Zuschauer handelt.

Parallel gehört die Nachweiskette aufgebaut: Auditlogs der Erbauer laufen in Microsoft Purview, Agentenaktivität lässt sich in Microsoft Sentinel überwachen und alarmieren. Wenn dein Haus Microsoft Agent 365 einführt, bekommen Agenten eine Entra-Identität und damit Conditional Access, rollen- und attributbasierte Zugriffe sowie Zugriffsgovernance. Der Preis dafür steht im Kleingedruckten: Sicherheits-Auditlogging und bestimmte Agent-365-Governance-Ereignisse fallen ausdrücklich nicht unter Customer Lockbox. Wer Lockbox als lückenlose Zusage verkauft hat, korrigiert das besser, bevor es der Auditor tut.

Was	Wer	Bis wann
Credit-Budget je Umgebung und Obergrenze für Pay-as-you-go festlegen	IT-Controlling und Power-Platform-Administration	vor dem ersten Prototyp
Datenrichtlinien für Knowledge, Konnektoren, Skills und Trigger schärfen	Sicherheit und Plattformbetrieb	vor der ersten Veröffentlichung
Entscheiden, ob externe Sprachmodelle freigegeben werden	Datenschutz und Sicherheit	sofort
Purview- und Sentinel-Auswertung für Agentenaktivität aufsetzen	Security Operations	parallel zum Piloten
Freigabeprozess für geteilte Skills definieren	Architektur und Fachbereich	vor der Skalierung
Zuordnung Anwendungsfall zu Harness dokumentieren	Architektur	vor der Skalierung

FAKTEN | Erst denken, dann anlegen

Weil sich ein Agent nachträglich nicht zwischen den Harnesses umziehen lässt, ist die Auswahl beim Anlegen eine Architekturentscheidung. Halte sie in deinen Richtlinien fest, sonst trifft sie derjenige, der zufällig als Erster auf „Neuer Agent“ klickt.

Häufig gestellte Fragen

Kann ich einen bestehenden Copilot-Studio-Agenten auf die GitHub-Copilot-Harness umstellen?

Nein. Agenten, die mit der GitHub-Copilot-Harness erstellt wurden, lassen sich nicht in die Standard-Harness überführen, und umgekehrt ebenso wenig. Du wählst die Harness beim Anlegen des Agenten, ein Wechsel bedeutet Neubau. Bestehende Agenten auf der Standard-Harness werden weiterhin vollständig unterstützt.

Ab wann kostet ein Agent Geld — ab dem Bauen oder ab der Veröffentlichung?

Bei der GitHub-Copilot-Harness ab dem Bauen. Anders als bei der Standard-Harness, wo die Abrechnung nach der Veröffentlichung beginnt, verbrauchen bereits das Erstellen per natürlicher Sprache, die Vorschau, der Testchat und das Erzeugen von Evaluationen Copilot Credits. Plane das Budget deshalb für die Entwurfsphase mit ein, nicht erst für den Betrieb.

Was passiert, wenn die Copilot Credits einer Umgebung aufgebraucht sind?

Dann stoppt alles, was Credits benötigt: Agenten antworten Anwendern nicht mehr, und Erbauer können weder per natürlicher Sprache entwickeln noch testen noch Evaluationen erzeugen. Microsoft erlaubt eine gewisse Überziehung als Karenz, damit Geschäftsprozesse nicht abrupt stehenbleiben. Abhilfe schaffen das Umverteilen vorhandener Kapazität, ein Zukauf oder ein eingerichteter Pay-as-you-go-Zähler.

Wie privat sind die Erinnerungen, die ein Agent über Anwender speichert?

Jeder Anwender hat einen eigenen Speicher, auf den weder andere Anwender noch der Ersteller des Agenten zugreifen können. Bleibt jemand 28 Tage inaktiv, löscht das System dessen Erinnerungen zu diesem Agenten, und in Gruppenchats sowie Teams-Kanälen ist Memory abgeschaltet. Wer Memory als Ersteller deaktiviert, verhindert nur die weitere Nutzung — die bereits gespeicherten Daten werden dadurch nicht gelöscht.

Reicht die eingebaute Evaluationsfunktion als Qualitätssicherung aus?

Für einen ersten Qualitätseindruck ja, als vollständige Abnahmeprüfung nein. Die derzeit einzige Testmethode bewertet KI-gestützt allgemeine Qualität wie Relevanz und Vollständigkeit und gleicht die Antworten ausdrücklich nicht gegen hinterlegte Sollantworten ab. Fachliche Korrektheit musst du weiterhin selbst gegenprüfen, idealerweise mit einem festen Satz Testgespräche vor jeder Veröffentlichung.