

Copilot-Update September: Code, Autopilot, Managed Runtime und Copilot Credits

Executive Summary

Microsoft hat am 25. September das Copilot-Haus umgebaut, und zwar mit Abrissbirne. Die Copilot-App gliedert sich jetzt in drei Räume: **Home** mit Chat und Cowork, **Code** zum Bauen von Apps und **Autopilot** für Agenten, die ohne Aufforderung weiterarbeiten. Darunter liegt die **Copilot Managed Runtime**, eine von Microsoft betriebene Hosting-Plattform, die selbstgebaute Apps innerhalb deiner Tenant-Grenze ausführt. Und damit das alles nicht zu billig wird, kommen die **Copilot Credits** als Abrechnungswährung für alles, was nach Arbeit aussieht.

Übersetzt heißt das: Deine Buchhalterin kann ab sofort sagen „Bau mir ein Tool, das offene Rechnungen nach Kostenstelle anzeigt und mich jeden Montag erinnert“, und Copilot baut es. Mit Entra-Anmeldung, im eigenen Tenant, sichtbar im Admin Center. Das ist die gute Nachricht. Die schlechte: Ab Oktober sehen lizenzierte Anwender die Einstiege in Code, Cowork und Autopilot standardmäßig, weil Microsoft den Admin-Schalter zum Verstecken streicht. Wer bis dahin keine Spielregeln hat, bekommt eine Schatten-IT mit Microsoft-Logo.

★ Wichtig: Die drei Sätze für den Vorstand

Copilot wird vom Assistenten zur App-Fabrik, und Fachabteilungen können ab Herbst eigene Anwendungen bauen, die Microsoft in unserem Tenant betreibt. Die Lizenz deckt nur noch den Alltag ab; Bauen, Betreiben und Agentenarbeit kosten Copilot Credits. Wir brauchen bis Ende Oktober Freigaberegeln, Budgets und Verantwortliche, sonst übernimmt der Zufall die Governance.

Worum geht es im Detail?

Copilot Code: Programmieren für Menschen, die „Programmieren“ für eine Waschmaschinenfunktion halten

Code lässt dich ein Tool, ein Dashboard, einen Tracker, eine Automatisierung oder einen Workflow in normaler Sprache beschreiben. Copilot entscheidet dann selbst, was daraus wird: ein dauerhaftes Desktop-Widget, ein interaktives Dashboard oder eine in der Cloud gehostete interne App, die du mit deinem Team teilst. Unter der Haube steckt dieselbe Technik wie bei GitHub Copilot, ausgeführt in einer Sandbox in deinem Tenant. Zielgruppe sind ausdrücklich keine Entwickler, sondern Fachanwender mit einem Problem und zu wenig Geduld für den IT-Ticketprozess.

Der Hintergrund: Microsoft hat jahrelang mit Power Apps versucht, Fachabteilungen zum Bauen zu bewegen. Das klappte, aber nur bei denen, die Formeln im Excel-Stil mögen. Mit generativer KI

fällt diese Hürde weg. Wer einen Satz formulieren kann, kann jetzt eine App bestellen. Code ist seit Ende September im Frontier-Programm verfügbar, eine breitere Öffnung folgt laut Microsoft „in den kommenden Wochen“. Voraussetzung ist eine Microsoft-365-Copilot-Lizenz, abgerechnet wird über Credits.

Copilot Managed Runtime: der Laufstall für Bürger-Apps

Seit dem 25. September in Public Preview, und das eigentliche Herzstück. Die Runtime betreibt Fachanwendungen, die automatisch die Governance deines Tenants erben, und zwar ab der ersten Sekunde. Egal, ob die App in Cowork, in Copilot Studio oder per SDK und CLI entsteht: Sie läuft in einer von Microsoft verwalteten Umgebung, meldet Nutzer über Entra an und hält sich an Conditional Access, DLP, Connector-Richtlinien und Freigabelimits. Über 1.500 Connectors sind direkt aus JavaScript und TypeScript ansprechbar, jede App bekommt ein Git-Repository. Jeder Maker arbeitet in einer persönlichen Entwicklerumgebung, die von der Produktion getrennt ist.

Anwender finden ihre Apps zentral unter managedapps.cloud.microsoft. Admins sehen alles im Microsoft 365 Admin Center unter **Apps > All apps**: Inventar, Owner, Nutzung, Health und Lebenszyklus. Die Bauwege steuerst du unter **Apps > Overview > Set up app creation spaces** oder per PowerShell und API. Das ist der entscheidende Unterschied zur Access-Datenbank auf Laufwerk S: Die App existiert nicht im Verborgenen, sondern in einem Inventar, in das du hineinschauen kannst. Ob du das auch tust, ist eine andere Frage.

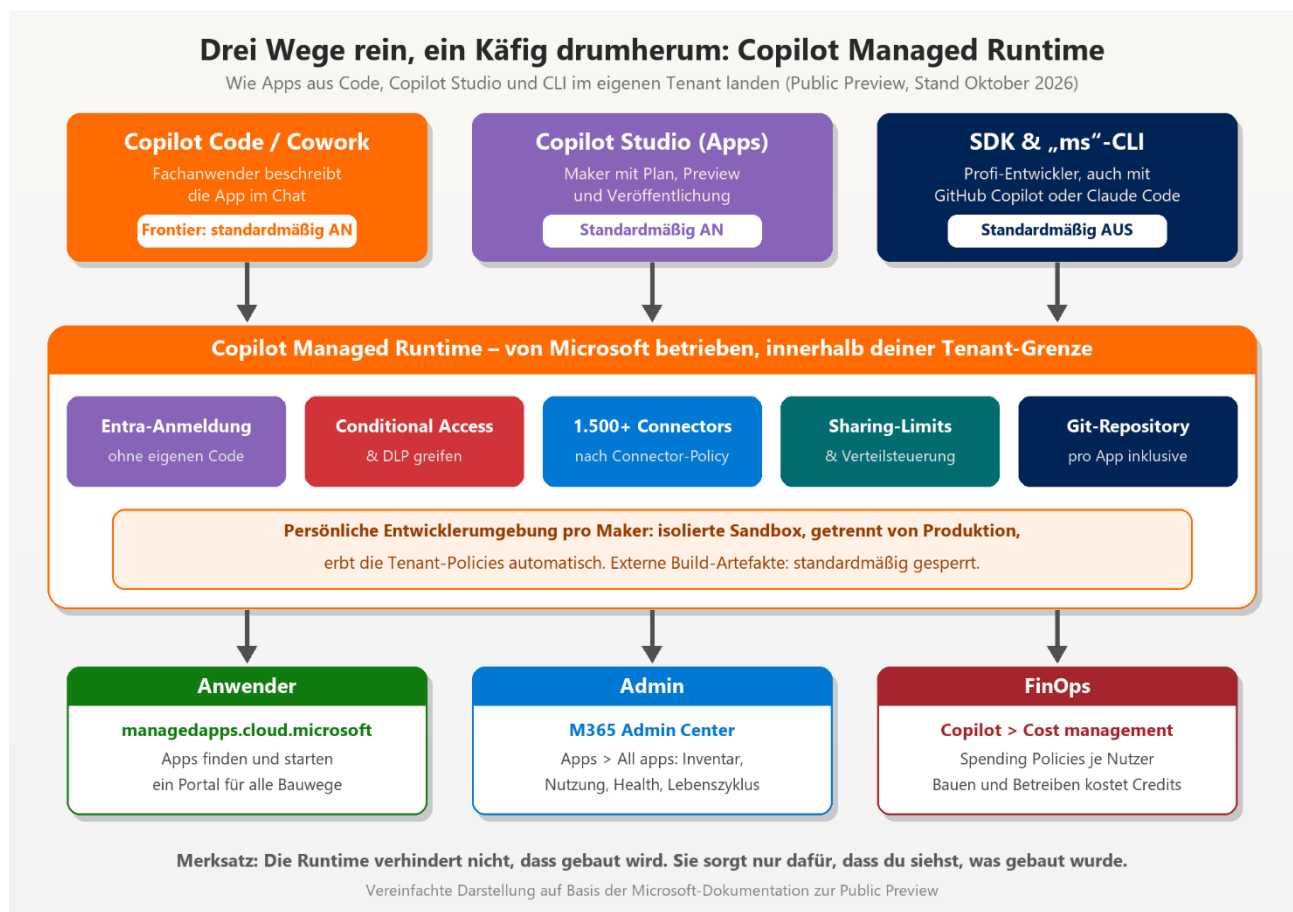


Abbildung 1: Drei Bauwege, eine Runtime – und am Ende drei Zielgruppen mit eigenen Werkzeugen

Bauweg	Für wen	Standard in der Preview	Wer darf schalten
Code / Cowork-App-Builder	Fachanwender im Chat	An, sobald der Tenant im Frontier-Programm ist	Frontier-Onboarding durch berechtigte Admins
Copilot Studio (Apps)	Maker mit mehr Kontrolle	An für alle berechtigten Nutzer	Global Admin, Power Platform Admin
SDK und „ms“-CLI	Profi-Entwickler	Aus, Freigabe per Umgebungsregel	Global Admin, Power Platform Admin
Externe Build-Artefakte	Entwickler mit eigener Pipeline	Aus	Pro Umgebung oder Umgebungsgruppe

⚠ **Warnung: Copilot Studio steht schon offen**

Der App-Bauweg in Copilot Studio ist in der Preview **standardmäßig eingeschaltet** und gilt für alle berechtigten Nutzer. Eingrenzen kannst du ihn nur über eine Sicherheitsgruppe per PowerShell oder API. Wenn du gerade denkst „Bei uns baut noch keiner Apps“: Prüf das Inventar. Optimismus ist keine Governance-Strategie.

Autopilot: der Kollege, der nie Urlaub nimmt

Autopilot hieß bis vor Kurzem Scout und ist ein persistenter Agent. Du gibst ihm einen Namen, eine Rolle und ein Ziel, und er legt los: beobachtet Kanäle, hakt bei Threads nach, erledigt wiederkehrende Aufgaben und nimmt ein Projekt Tage später wieder auf, ohne dass ihn jemand anstupst. Er läuft in der Cloud, hat eine **eigene Identität im Verzeichnis**, einen eigenen Arbeitsbereich und taucht in Teams, Outlook, Chats, Kanälen und Dokumenten auf. Seit Ende September in einer wachsenden Private Preview, Lizenzvoraussetzung wie bei Code, Abrechnung über Credits.

Die eigene Identität ist Fluch und Segen. Segen, weil du einen Autopilot wie ein Benutzerkonto behandeln kannst: Berechtigungen, Audit, Lebenszyklus, alles mit vorhandenen Werkzeugen. Fluch, weil du plötzlich Konten hast, die niemand zum Mitarbeitergespräch einlädt und die trotzdem in der Projektgruppe mitlesen. Jeder Autopilot braucht deshalb einen verantwortlichen Menschen, eng geschnittene Rechte und eine vollständige Protokollierung.

Copilot Credits: Die Flatrate hat jetzt Kleingedrucktes

Microsoft fährt ab sofort zweigleisig. Die Lizenz pro Nutzer deckt den Alltag ab: Chat, Office in Copilot im Chat-Modus, auch lang laufende mehrstufige Aufgaben im Chat. Alles, was Copilot eigenständig für dich erledigt, kostet Credits: Cowork, Code, Autopilot, erweiterte Arbeiten in SharePoint und OneDrive sowie Frontier-Modelle wie Astra und Fable. Ein Credit kostet im Pay-as-you-go-Modell 0,01 US-Dollar. Wer ein Jahr im Voraus kauft (Pre-Purchase-Plan P3), bekommt je nach Volumen zwischen 5 und 20 Prozent Rabatt, die Staffel beginnt bei 300.000 Credits.

Bei der Managed Runtime gibt es zwei Kostentöpfe, und genau das sorgt für Verwirrung. **Bauen** kostet Credits: in Cowork über die Spending Policy des Erstellers, in Copilot Studio pro Umgebung über das Power Platform Admin Center, und zwar schon beim Erstellen, Testen und Evaluieren,

nicht erst nach der Veröffentlichung. **Betreiben** kostet ebenfalls Credits, aber pro Nutzer der App, gesteuert im Microsoft 365 Admin Center unter **Copilot > Cost management**. Ausnahme: Nutzer mit Power-Apps-Premium-Lizenz verbrauchen beim Ausführen keine Credits, solange keine separat abgerechneten Dienste wie Work-IQ-APIs im Spiel sind.

Funktion	Oberfläche	Abrechnung
Office in Copilot im Chat	Chat	Lizenz
Lang laufende mehrstufige Aufgaben	Chat	Lizenz
Office in Copilot in Cowork	Cowork	Credits
Code (App bauen)	Copilot-App	Credits
Autopilot	Copilot-App	Credits
Erweiterte Arbeit in SharePoint und OneDrive	SharePoint/OneDrive	Credits
App-Betrieb in der Managed Runtime	managedapps.cloud.microsoft	Credits pro Nutzer (Ausnahme Power Apps Premium)

Dazu kommen FinOps-Werkzeuge in der Preview: Modelle und Leistungsstufen pro Gruppe zuweisen, festlegen, welche Modelle die Auto-Auswahl in Cowork nehmen darf, Spending Policies per API verwalten, Credit-Anfragen durch bestehende Genehmigungsprozesse schicken und den Verbrauch in Dashboards gegen Geschäftsergebnisse halten. Copilot-Studio-Agenten kommen im Oktober dazu. Das ist der Teil, den dein Controlling lieben wird, sobald es davon erfährt.

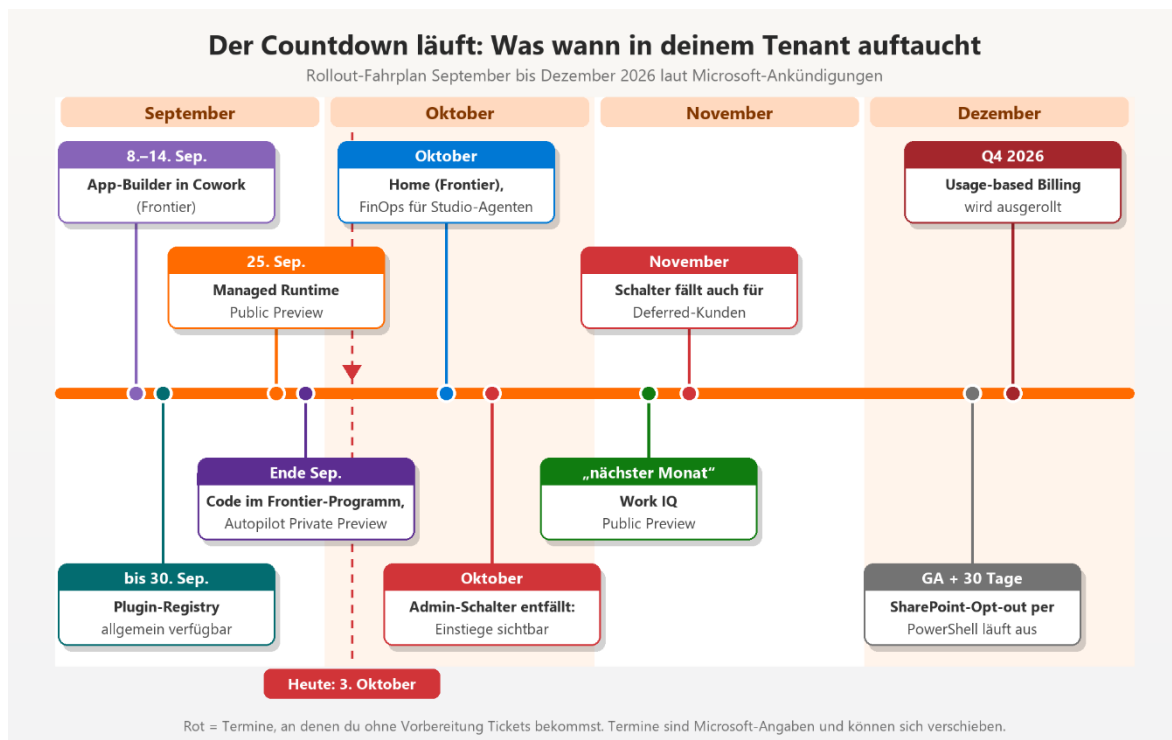


Abbildung 2: Rollout-Fahrplan – die roten Termine sind die, an denen dein Helpdesk aufwacht

i Fakten: Die 20-Klick-Falle der Preview

Wer eine App nutzt, aber in keiner Runtime-Spending-Policy steckt, bekommt in der Preview zunächst nur eine Warnung. Nach **20 App-Operationen oder fünf Minuten Nutzung**, je nachdem, was zuerst eintritt, ist Schluss. Klassiker: Die Fachabteilung baut ein schickes Tool, präsentiert es im Teammeeting, und beim sechsten Kollegen bricht es ab. Die Demo ist dann vorbei, das Ticket beginnt.

Was sind Chancen? Was sind Risiken?

Fangen wir mit der ehrlichen Einordnung an: Fachabteilungen bauen seit 30 Jahren Software an der IT vorbei. Sie haben es mit Access gemacht, mit Excel-Makros, mit SharePoint-Designer-Workflows und zuletzt mit Power Automate. Neu ist nicht das Bauen, neu ist die Geschwindigkeit. Was früher ein verregnetes Wochenende mit VBA-Handbuch brauchte, dauert jetzt eine Kaffeepause. Und neu ist, dass die Plattform diesmal von Anfang an ein Inventar führt.

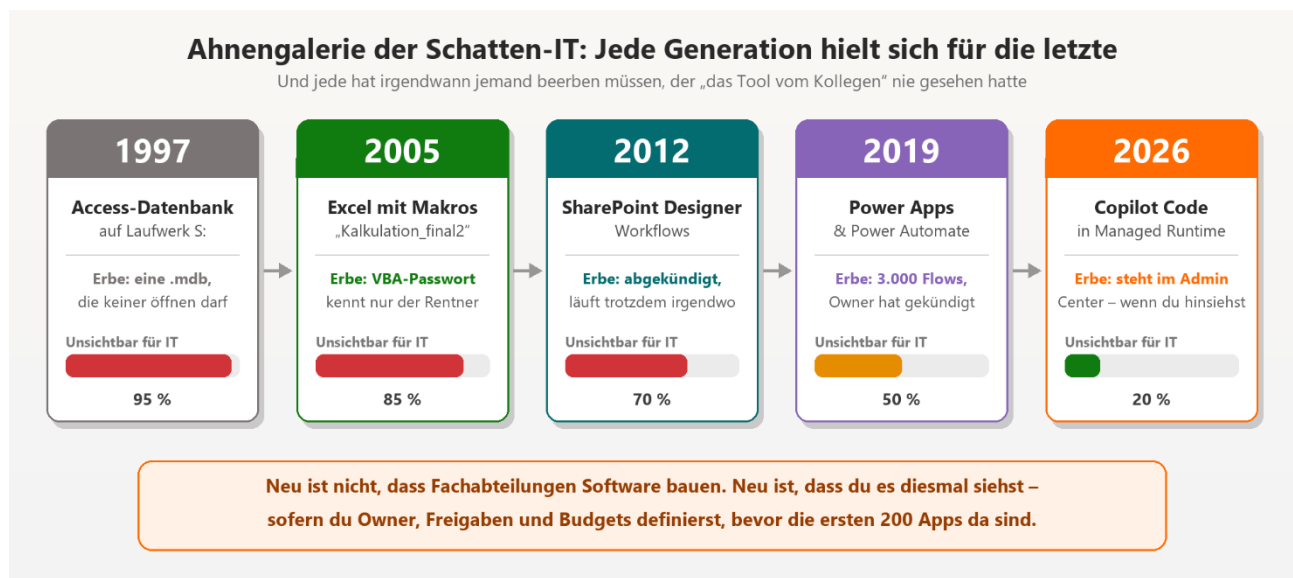


Abbildung 3: Schatten-IT im Generationenvergleich – die Sichtbarkeit steigt, das Grundproblem bleibt

Chancen	Risiken
Der Ticket-Rückstau für Kleinst-Tools schrumpft, weil Fachbereiche selbst bauen	App-Wildwuchs: 200 Apps in drei Monaten, davon 150 Varianten derselben Urlaubsliste
Apps laufen mit Entra, Conditional Access und DLP statt auf einem Laufwerk	Connectors öffnen Datenquellen, wenn die Connector-Policy zu großzügig ist
Zentrales Inventar mit Owner, Nutzung und Health	Owner verlassen das Unternehmen, Apps bleiben geschäftskritisch
Autopilot übernimmt Nachfass- und Routinearbeit rund um die Uhr	Agenten mit eigener Identität sammeln Rechte wie ein Praktikant Schlüssel
Credits machen Kosten pro Nutzen messbar	Variable Kosten ohne Budget, und das Bauen kostet schon beim Testen
Profi-Entwickler bekommen über CLI und SDK eine verwaltete Plattform	Betriebsrat und Datenschutz werden von sichtbaren Agenten-Einstiegen überrascht

Ein Praxisbeispiel aus einem Projekt bei einem Maschinenbauer: Der Vertriebsinnendienst hatte schon in der Power-Apps-Ära ein „kleines Angebotstool“ gebaut. Drei Jahre später lief darüber ein Viertel aller Angebote, die Erstellerin war in Elternzeit, und der Flow hing an ihrem persönlichen Konto. Als ihr Konto nach Ablauf der Frist deaktiviert wurde, stand der Vertrieb still. Niemand war schuld, alle waren betroffen. Mit Copilot Code passiert dasselbe, nur schneller, und mit hübscherer Oberfläche.

⚠ Warnung: Der „request access“-Tsunami

Ab Oktober (Deferred-Kunden ab November) entfernt Microsoft die Einstellung, mit der du die nutzungsbasierten Einstiege verstecken konntest. Lizenzierte Nutzer sehen Cowork, Code und Autopilot. Wer keine Spending Policy hat, bekommt einen „request access“-Hinweis, und jede dieser Anfragen landet irgendwo. Vermutlich bei dir. Du kannst die Meldung auf „learn more“ umstellen; dann gibt es Lesestoff statt Tickets.

Und noch ein Risiko, das gern übersehen wird: Wer per PowerShell die SharePoint-Copilot-Preview abgewählt hat, behält diese Einstellung nur 30 Tage nach der allgemeinen Verfügbarkeit. Danach ist das Opt-out Geschichte. Wenn dein Datenschutzkonzept auf diesem Schalter beruht, hat es ein Verfallsdatum.

Was müssen wir jetzt schon vorbereiten?

Die gute Nachricht: Microsoft liefert die Werkzeuge mit. Die schlechte: Sie stehen größtenteils auf „an“, und die Entscheidungen, wer was darf, kann dir kein Admin Center abnehmen. Hier ist die Reihenfolge, die sich in den ersten Projekten bewährt.

Maßnahme	Warum	Bis wann
Bauwege festlegen: Studio per Sicherheitsgruppe eingrenzen, CLI bewusst freigeben, Frontier-Teilnahme entscheiden	Studio ist standardmäßig an, Code kommt über Frontier	Sofort
Spending Policies für Cowork, Code und Runtime anlegen, mit Limits pro Nutzer	Ohne Policy keine Nutzung, mit falscher Policy keine Kostenkontrolle	Vor Oktober-Rollout
„request access“ auf „learn more“ umstellen oder Anfrageprozess definieren	Sonst entstehen ungeplante Tickets	Oktober bzw. November
Connector- und DLP-Richtlinien prüfen	Apps erben genau diese Policies, nicht mehr und nicht weniger	Vor den ersten Apps
Owner- und Stellvertreterpflicht für jede veröffentlichte App	Verwaiste Apps sind die teuerste Form der Schatten-IT	Mit der Freigaberichtlinie
Autopilot-Richtlinie: Owner, Rechtemumfang, Audit, Ablaufdatum	Agenten mit eigener Identität sind ein neuer Kontotyp	Vor jeder Private-Preview-Teilnahme
Betriebsrat und Datenschutz einbinden	Agenten beobachten Kanäle und Threads	Jetzt, nicht nach dem ersten Agenten

Hintergrund zur Owner-Pflicht: Das Admin Center zeigt dir, wem eine App gehört. Es zeigt dir aber nicht, was passiert, wenn dieser Mensch geht. Leg deshalb fest, dass jede App, die über einen persönlichen Gebrauch hinausgeht, einen zweiten Owner und eine verantwortliche Abteilung bekommt. Koppel das an deinen Offboarding-Prozess: Wer geht, übergibt seine Apps wie seinen Laptop. Das klingt banal, ist aber der Unterschied zwischen einem Inventar und einem Friedhof mit Grabsteinbeschriftung.

✓ Tipp: Die Drei-Stufen-Ampel für Bürger-Apps

Grün: persönliche Apps in der eigenen Entwicklerumgebung, keine Freigabe nötig. **Gelb:** Team-Apps mit bis zu einer festen Nutzerzahl, Owner und Stellvertreter Pflicht, kurzer Check durch die IT. **Rot:** organisationsweite Apps oder solche mit personenbezogenen Daten, also echtes Projekt mit Review, Test und Betriebsverantwortung. Drei Stufen merkt sich jeder, zwölf Seiten Richtlinie liest keiner.

Zur Budgetplanung ein einfaches Rechenbeispiel, ausdrücklich ohne Microsoft-Verbrauchswerte: Wenn du 50 Makern je 2.000 Credits im Monat zum Bauen gibst, sind das 100.000 Credits oder 1.000 US-Dollar im Pay-as-you-go-Modell. Das ist überschaubar. Teuer wird es nicht durch die Maker, sondern durch die Nutzer erfolgreicher Apps und durch Autopiloten, die rund um die Uhr arbeiten. Miss deshalb in den ersten acht Wochen tatsächlich, bevor du Jahrespakete kaufst. Die P3-Rabatte laufen dir nicht weg, die Fehlinvestition schon.

✓ Tipp: Der Montagmorgen-Check

Plane einen festen 15-Minuten-Termin pro Woche: Apps > All apps öffnen, neue Apps ansehen, Owner prüfen, Kostenkurve in Copilot > Cost management anschauen. Nach acht Wochen weißt du mehr über die digitale Realität deiner Fachbereiche als nach jedem Workshop.

Zum Schluss ein Wort an alle, die jetzt reflexartig alles abschalten wollen: Das wird nicht funktionieren. Wer seinen Fachbereichen die verwaltete Variante verweigert, bekommt die unverwaltete. Die gibt es im Browser, mit privater Kreditkarte und ohne jedes Inventar. Die Managed Runtime ist das erste Mal seit Jahrzehnten, dass Citizen Development und IT-Governance auf derselben Plattform stattfinden. Nutze das, solange die Apps noch in einer Tabelle Platz haben.

Häufig gestellte Fragen

Was ist die Copilot Managed Runtime?

Die Copilot Managed Runtime ist eine von Microsoft betriebene Plattform, auf der selbstgebaute Fachanwendungen innerhalb des eigenen Microsoft-365-Tenants laufen. Apps aus Copilot Code, Cowork, Copilot Studio oder dem SDK nutzen dort automatisch Entra-Anmeldung, Conditional Access, DLP und Connector-Richtlinien und erscheinen im Admin Center unter Apps > All apps.

Brauche ich für Copilot Code eine eigene Lizenz?

Copilot Code setzt eine Microsoft-365-Copilot-Lizenz voraus, die eigentliche Nutzung wird aber zusätzlich über Copilot Credits abgerechnet. Damit Anwender bauen können, muss die nutzungsbasierte Abrechnung aktiviert und der Nutzer einer passenden Spending Policy zugeordnet sein.

Was kostet ein Copilot Credit?

Im Pay-as-you-go-Modell kostet ein Copilot Credit 0,01 US-Dollar und wird monatlich nachträglich abgerechnet. Beim einjährigen Pre-Purchase-Plan P3 gibt es je nach Abnahmemenge ab 300.000 Credits einen Rabatt zwischen 5 und 20 Prozent.

Kann ich Copilot Code und Autopilot in meinem Tenant ausblenden?

Den bisherigen Admin-Schalter zum Verstecken nutzungsbasierter Funktionen entfernt Microsoft im Oktober, für Deferred-Kunden im November. Steuern lässt sich der Zugriff dann über Spending Policies, Sicherheitsgruppen für die Bauwege und die Teilnahme am Frontier-Programm; ohne Policy sehen Nutzer einen Hinweis zum Anfordern des Zugriffs.

Wie verhindere ich, dass mit Copilot Code neue Schatten-IT entsteht?

Grenze die Bauwege über Sicherheitsgruppen ein, prüfe Connector- und DLP-Richtlinien und verlange für jede geteilte App einen Owner mit Stellvertreter. Kontrolliere zusätzlich regelmäßig das App-Inventar im Microsoft 365 Admin Center und binde die Übergabe von Apps in den Offboarding-Prozess ein.