QM (yc-software/qm)

Multiplayer-Agent-Harness für Teams, das Mitarbeitern isolierte Arbeitsbereiche mit gemeinsamer Kollaboration in Slack und Web bietet.

STERNE	LIZENZ	SPRACHE	VERSION	ANGELEGT	ZULETZT GEÄNDERT
14.348	MIT	TypeScript	v0.1.5	2026-07-29	2026-08-28

Was es ist

QM ist ein Agent-Framework für Startups, bei dem jeder Mitarbeiter einen eigenen isolierten Arbeitsbereich erhält, gleichzeitig aber in Slack-Kanälen, Gruppennachrichten und Projekten mit dem Agenten zusammenarbeiten kann. Es unterscheidet sich von persönlichen Assistenten durch seine Team-Ausrichtung mit getrennten, aber verknüpften Scopes für Personen und Räume. Der Kern ist herstellerunabhängig: Pi, OpenCode, Codex und Claude Code können als Harness eingesetzt werden.

Was es kann

- Jeder Person und jedem Raum einen eigenen Scope mit isoliertem Memory, Dateien, Keychain, Berechtigungen, Crons und Sandbox zuweisen.
- In Slack und Web-App mit derselben Identität und Konfiguration arbeiten.
- Interne Web-Apps erstellen und gezielt an Personen oder Gruppen veröffentlichen.
- Skills im Scope erstellen, per Grant teilen und mit Admin-Freigabe auf Organisationsebene hochstufen.
- Hintergrundarbeit über Crons, Watches und eingehende Webhooks ausführen.
- In bestehenden Repositories arbeiten: Tests laufen lassen, PRs öffnen, CI überwachen, Systemlogs prüfen.

So fängst du an

Die Initialisierung erfolgt über das qm-CLI, das eine Deployment-Skill für einen Agenten materialisiert und durch Infrastruktur, Authentifizierung und Verifikation führt.

```
npm exec --yes --package=@yc-software/qm@latest -- qm init . --org  --target
npm install
```

Grenzen und Bedingungen

- Kein fertiges Hosting: Jede Deployment läuft im eigenen Cloud-Account des Betreibers (Fly oder AWS), es gibt keinen verwalteten Dienst.
- Keine automatische CI/CD: Die Initialisierung erzeugt keine Deployment-CI, und das Repository selbst hat keinen Production-Deployment-Workflow.
- Contributions nur als menschlich geschriebener Text, nicht als Code: Änderungswünsche werden als .txt oder .md in adrs/ eingereicht, die Implementierung erfolgt durch das Core-Team.

- Sicherheitsposten 'Dangerous' deaktiviert alle Inhaltsscreenings und Pausen zwischen Tool-Aufrufen, was mit Vorsicht zu genießen ist.
- Private Forks dürfen nicht über GitHubs Fork-Button erstellt werden, da dieser die Sichtbarkeit des Ursprungsrepos übernimmt und Commits über SHA aus dem öffentlichen Bereich abrufbar bleiben; stattdessen ist ein plain clone nötig.

Für wen sich das lohnt

QM lohnt sich für Startups und kleine Firmen, die einen eigenen Agenten für das gesamte Team betreiben wollen, dabei aber Herstellerunabhängigkeit wahren und die volle Kontrolle über Daten und Infrastruktur behalten möchten. Weniger geeignet ist es für Einzelnutzer oder Teams, die eine sofort einsatzbereite Cloud-Lösung ohne eigenes Deployment suchen.



Zusammengestellt von **localdev.ai**
Quelle: **github.com/yc-software/qm** · Stand 2026-08-28

Grundlage: README: Abschnitte 'What is QM?', 'Features', 'What you can do with it', 'Architecture', 'Security and secrets', 'Deploy it for your org', 'Customize your instance'; Metadaten: Sterne 14348, Lizenz MIT, Version v0.1.5, 352 offene Meldungen, Sprachanteile TypeScript 92.6%; Architecture: Postgres als Persistence Layer, Fastify für HTTP, Bolt für Slack, Vite/Lit für Web UI; Security: Drei Posten (Str