# eve Software Factory Template (Foreman)

Ein auf Vercel deploybarer Software-Factory-Workflow, der Aufgaben aus GitHub und Linear durch vier spezialisierte KI-Agentenstationen leitet und als Entwurf-Pull-Request abschließt.

| STERNE | LIZENZ | SPRACHE | VERSION | ANGELEGT | ZULETZT GEÄNDERT |
|---|---|---|---|---|---|
| 1.117 | MIT | TypeScript | - | 2026-08-12 | 2026-08-20 |

## Was es ist

Foreman ist ein Template für eine sogenannte Software Factory auf Basis des eve-Frameworks. Es automatisiert den Entwicklungszyklus, indem es eingehende Aufgaben klassifiziert, plant, implementiert und reviewt – jeweils durch separate Agenten mit eigenen Sandboxes. Menschen bleiben bei den Entscheidungen: Sie reviewen den Entwurf-Pull-Request und mergen erst bei Freigabe. Besonderheit ist die Trennung der Stationen, insbesondere dass der Reviewer nur den gepushten Branch sieht, nie die Gedankengänge des Implementierers.

## Was es kann

- Aufgaben aus GitHub-Issues (via Label 'factory' oder Mention) und Linear automatisch entgegennehmen und verarbeiten.
- Vier Stationen nacheinander durchlaufen: Classifier triagiert, Analyst erstellt einen Plan, Implementer führt aus und pusht einen Branch, Reviewer bewertet den Diff unabhängig.
- Einen Entwurf-Pull-Request auf dem konfigurierten FACTORY_REPO erstellen, der auf menschliche Freigabe wartet.
- Rote CI-Builds auf eigenen Factory-Branches diagnostizieren und korrigieren, ohne die Hauptbranches zu berühren.
- Lokal über eine TUI (Terminal User Interface) mit 'pnpm dev' gestartet werden; lokale Änderungen warten auf explizite Bestätigung, bevor sie GitHub erreichen.
- Zwischen den Durchläufen ein 'factory brain' pflegen: Notizen über das Repository, die jeder neue Durchlauf als Kontext erhält.

## So fängst du an

Das Projekt wird über Vercel deployt, verlangt aber Vorbereitung: Ein echtes Repository in owner/repo-Format und ein installierter GitHub-App-Zugriff darauf.

```
vercel link
vercel env pull
pnpm dev
```

## Grenzen und Bedingungen

- Erfordert ein Vercel-Konto und die Bereitschaft, Connectoren für GitHub und Linear über Vercel zu verwalten; ohne diese Infrastruktur ist das Template nicht nutzbar.
- FACTORY_REPO muss ein echtes, erreichbares Repository sein; bei unzureichenden Berechtigungen für den GitHub-App-Connector schlägt das Deployment mit 'Cannot clone' fehl.
- Die Freigabe (Release) ist auf 'None' gesetzt, was auf keinen stabilen Versionsstand hindeutet; Nutzer deployen einen Entwicklungsstand.
- Lokale Entwicklungsläufe werden als 'untrusted' behandelt, was bedeutet, dass keine automatische GitHub-Integration erfolgt – jeder Schritt erfordert manuelle Bestätigung in der TUI.
- Die Lizenz ist MIT, was kommerzielle Nutzung erlaubt, aber das Projekt ist ein Template von Vercel Labs; langfristige Wartung und Support sind nicht garantiert.

## Für wen sich das lohnt

Das Template lohnt sich für kleine Teams oder Selbständige, die bereits Vercel nutzen und einen wiederkehrenden Strom klar definierbarer Entwicklungsaufgaben haben, die sie gerne automatisiert vorbereitet, aber selbst final absegnen möchten. Wer keine Vercel-Infrastruktur betreibt, keine GitHub- oder Linear-Integration braucht, oder direkte Kontrolle über jeden einzelnen Agentenschritt behalten will, ist hier schlechter aufgehoben als mit manuellen Workflows.

---

Zusammengestellt von **localdev.ai**
Quelle: **github.com/vercel-labs/eve-software-factory-template** · Stand 2026-08-20

Grundlage: Liesmich-Datei, Abschnitt 'eve Software Factory Template': Beschreibung der vier Stationen und deren Trennung; Liesmich-Datei, Abschnitt 'How work arrives': Auslöser via Label 'factory', Mention, Linear, TUI, rote CI, PR-Kommentar; Liesmich-Datei, Abschnitt 'Deploy': Vercel-Deploy-Button, Umgebungsvariablen FACTORY_REPO und FACTORY_LABEL, Notwendigkeit des GitHub-App-Zugriffs, Fehlermeldung 'Canno