

TRIZ Skills für Claude

Diese Sammlung enthält 27 TRIZ-Skills (4 Business-TRIZ, 23 Technical-TRIZ), die als Claude-Skills genutzt werden können — in **Claude Desktop (Cowork)** oder **Claude Code**.

Jedes Skill liegt als fertige ZIP-Datei vor und kann direkt installiert werden.

Installation

In Claude Desktop (Cowork)

1. Öffne die Claude Desktop App.
2. Gehe zu **Customize** (Anpassen) → **Skills**.
3. Klicke auf „+“ → „+ Create skill“.
4. Wähle die gewünschte ZIP-Datei aus (z.B. `contradiction-solver.zip`).
5. Fertig — der Skill ist sofort verfügbar.

Wiederhole Schritte 3–4 für jedes weitere Skill, das du nutzen möchtest.

In Claude Code

1. Entpacke die ZIP-Datei.
2. Kopiere den entpackten Ordner an einen der folgenden Orte:

```
# Persönlich (in allen Projekten verfügbar)
cp -r contradiction-solver ~/.claude/skills/

# Oder projektspezifisch (nur in diesem Projekt verfügbar)
cp -r contradiction-solver .claude/skills/
```

Claude Code erkennt den Skill automatisch beim nächsten Start.

Verwendung

Automatische Erkennung

Im Normalfall musst du nichts Besonderes tun. Claude liest die Beschreibung jedes installierten Skills und entscheidet selbstständig, ob ein Skill zur aktuellen Aufgabe passt. Wenn ja, wird der Skill-Inhalt automatisch geladen und Claude folgt den darin enthaltenen Anweisungen.

Beispiele für Prompts, die automatisch den passenden Skill auslösen:

- „Hilf mir, einen technischen Widerspruch bei meiner Kaffeemaschine zu lösen.“ → `contradiction-solver`
- „Wie wird sich das Bankwesen in 20 Jahren entwickeln?“ → `business-msd`
- „Analysiere die Funktionen meines Staubsaugers.“ → `function-analysis`
- „Welche Ressourcen kann ich in meinem Produktionsprozess besser nutzen?“ → `resource-analysis`
- „Ich möchte Komponenten aus meinem System entfernen, ohne die Hauptfunktion zu verlieren.“ → `trimming`

Skill gezielt aufrufen

Wenn du sicherstellen möchtest, dass ein bestimmtes Skill verwendet wird — unabhängig davon, ob Claude es automatisch erkennen würde — kannst du es explizit anfordern:

In Claude Desktop (Cowork): Verwende einen Slash-Command mit dem Skill-Namen:

```
/contradiction-solver
```

Damit wird das Skill sofort geladen. Anschließend kannst du deine eigentliche Frage stellen oder sie direkt an den Slash-Command anhängen.

In Claude Code: Verwende ebenfalls den Slash-Command:

```
/contradiction-solver
```

Alternative: Im Prompt erwähnen. Du kannst Claude auch direkt bitten, ein bestimmtes Skill zu nutzen:

```
verwende das skill "resource-analysis" und hilf mir, die Ressourcen in meinem System zu identifizieren.
```

oder:

```
Bitte wende die TRIZ 76 Standardlösungen auf mein Problem an.
```

Solange du den Skill-Namen oder die Methode deutlich genug nennst, wird Claude das passende Skill aktivieren.

Aufbau eines Skills

Jedes Skill besteht aus einem Ordner mit mindestens einer `SKILL.md`-Datei:

```
contradiction-solver/  
├─ SKILL.md           ← Hauptdatei (Pflicht)  
├─ references/        ← Nachschlagewerke (optional)  
│   ├── 40_Inventive_Principles_EN.md  
│   ├── Altshuller_Contradiction_Matrix_AI.csv  
│   └─ ...
```

Die `SKILL.md` enthält:

- **YAML-Frontmatter** mit `name` und `description` — die Beschreibung bestimmt, wann Claude den Skill automatisch aktiviert
- **Markdown-Anweisungen** — die Schritt-für-Schritt-Anleitung, der Claude folgt

Einige Skills enthalten zusätzlich `references/` (Nachschlagewerke wie die Widerspruchsmatrix oder die 40 Erfindungsprinzipien) und `assets/` (Vorlagen wie die ISQ++-Fragebögen). Diese werden bei Bedarf von Claude gelesen.

Übersicht aller Skills

Business-TRIZ

Skill	Beschreibung
business-msd	Multi-Screen Diagram (9 Boxes) für Business- und Techniksysteme
perception-mapping	Perception Mapping — Stakeholder-Überzeugungen und Widersprüche aufdecken
solutions-at-system-levels	Lösungsideen auf verschiedenen Systemebenen (Sub-/Super-System)
business-function-analysis	Vereinfachte Funktionsanalyse für Nicht-Ingenieure

Technical-TRIZ

Skill	Beschreibung
system-description	Strukturierte Systembeschreibung (Mini-ISQ)
function-analysis	TRIZ-Funktionsanalyse (Tool-Action-Object)
function-analysis-advanced	Erweiterte Funktionsanalyse mit Zeit-/Ort-Dimensionen
function-interaction-analysis	Komponenteninteraktions- und Funktionsanalyse
76-standard-solutions	76 Standardlösungen mit Stoff-Feld-Analyse
cause-effect-chain-analysis	Ursache-Wirkungs-Kettenanalyse (CECA) und RCA
contradiction-solver	Widerspruchslöser mit 40 Erfindungsprinzipien, Altshuller- und 2003-Matrix
feature-transfer	Feature Transfer zwischen Systemen
function-oriented-search	Funktionsorientierte Suche, Scientific Effects, FOS/MOS
ideality	Idealitätssteigerung technischer Systeme
innovation-checklist	Innovation Situation Questionnaire (ISQ++)
mpv-analysis	Parameter of Value / Main Parameter of Value Analyse
patent-analyzer	Widersprüche in Patentschriften identifizieren
physical-contradictions	Physikalische Widersprüche und Separationsprinzipien
resource-analysis	Ressourcenanalyse (6 Ressourcentypen)

Skill	Beschreibung
<code>derivative-resources</code>	Derivative Ressourcen — versteckte Potenziale entdecken
<code>root-cause-analysis</code>	Root Cause Analysis für Ingenieurprobleme
<code>root-conflict-analysis</code>	Grundkonfliktanalyse (RCA+) — Widerspruchsursachen identifizieren
<code>smart-little-people</code>	Smart Little People (SLP) — kreative Mikro-Visualisierung
<code>system-operator</code>	System Operator (9 Boxes) — Evolution und Problemlösung
<code>trimming</code>	TRIZ Trimming mit Regeln A, B, C
<code>interactive-trimming</code>	Interaktives Trimming mit erweiterten Regeln (A-E, X)
<code>sustainability-assessment</code>	Nachhaltigkeitsbewertung mit TRIZ-Funktionsanalyse

Lizenz

Alle Skills in dieser Sammlung stehen unter der **MIT License**. Das bedeutet: Sie dürfen frei verwendet, kopiert, verändert und auch kommerziell eingesetzt werden — einzige Bedingung ist, dass der Urheberrechtshinweis und die Lizenz beibehalten werden.

Die Skills werden bereitgestellt, wie sie sind, ohne Gewährleistung oder Haftung jeglicher Art.

Den vollständigen Lizenztext finden Sie unter [LICENSE](#). Die jeweiligen Autoren sind in jeder SKILL.md-Datei angegeben.