

Claude Code `</>`

VAN BASIS TOT MEESTERSCHAP

De complete gids voor Agentic
softwareontwikkeling in 2026



Installeren &
Aan de slag



Prompt
Engineering



Multi-agent
workflows



Beveiliging &
Beste praktijken



Projecten uit
de praktijk

stripe

Case studies
van Stripe
& Wiz

wiz



CI/CD
automatisering



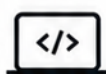
De toekomst van
agentic coderen

Een uitgebreide, praktische gids voor Claude Code, Anthropic's
agentic coding-systeem dat **verandert hoe software wordt gebouwd**.

Van de eerste installatie tot enterprise-uitrol—beheers de tools,
technieken en workflows die toonaangevende engineeringteams gebruiken.



Praktische voorbeelden
in elk hoofdstuk



Praktijkprojecten
die je zelf kunt bouwen



Lessen uit
echte teams



Gebaseerd op onderzoek
met 100K+ sessies

STEVE T.

Claude Code: Van basis tot meesterschap

De Complete Gids voor Agentic Softwareontwikkeling
in 2026

Steve T. Publications

Dit boek is te koop bij <https://leanpub.com/claudecodenl>

Deze versie is gepubliceerd op 2026-07-16



Dit is een [Leanpub](#) boek. Leanpub stelt auteurs en uitgevers in staat om volgens het Lean uitgeefproces te werken. [Lean Publishing](#) is het uitgeven van een boek dat nog onderhanden is met lichtgewicht gereedschap en vele iteraties om feedback te krijgen van de lezers. Op deze manier kun je aanpassingen maken tot je het juiste boek hebt, en als je zover bent helpt het om te zorgen dat je een positie krijgt in de markt.

© 2026 Steve T. Publications

Inhoudsopgave

De Complete Gids voor Agentic Softwareontwikkeling in 2026	1
Een Uitgehand Handboek voor het Beheersen van Claude Code, Multi-Agent Workflow en Enterprise Implementatie	1
Inleiding: De Ingenieur die Niet Meer Typt	3
Wat Dit Boek Je Zal Leren	3
Voor Wie Dit Boek Bestemd is	4
Hoe Dit Boek te Gebruiken	4
Het Bewijsbasis	5
Een Opmerking over Wat Dit Boek Niet Dekt	5
De Centrale These	5
Het Landschap in 2026	6
Hoofdstuk 1: De Agentic Revolutie in Softwareontwikkeling	7
Van Autocomplete naar Agents: Een Vijfjarige Evolutie	7
Wat Claude Code Anders Maakt: Terminal-First Agentic Architectuur	8
Boris Cherny’s Workflow: Hoe de Maker Eigenlijk Claude Code Gebruikt	10
De Aandeling van Arbeid: Mens als Orkestatior, Agent als Bouwer	11
Het Landschap van AI Coding Agents in 2026	13
Waarom Nu? Marktadoptie en de Verschuiving naar Agentic Workflows	13
Kritische Analyse: Trade-offs en Beperkingen van Agentic Tools	15
Hoofdstuk 2: Anatomie van Claude Code	19
De Agentic Lus: Verzamelen, Handelen, Verifiëren	19
Modellen Achter de Schermen: Sonnet, Opus, Fable en Modelselectie	19
Token Management Math: Je Uitgaven Begrijpen	20
Ingebouwde Tools: Bestandsoperaties, Zoeken, Uitvoering, Web, Code-inzicht	20
De .claude Directory: Configuratie, CLAUDE.md en Geheugen	20
Context Windows: Hoe Ze Vullen, Auto-compactie en Token Management	20
Hoofdstuk 3: Aan de Slag en Op Zetten	22

Installatie Over Platforms: macOS, Linux, Windows, WSL, Homebrew, WinGet	22
Authenticatie en Account Setup	22
Je Eerste Sessie: Walkthrough van Zero naar Code	22
Permission Modes: Default, Auto-accept, Plan en Auto Mode	23
IDE Integraties: VS Code, JetBrains, Remote Control, Webinterface	23
Hoofdstuk 4: Prompt Engineering voor Claude Code	24
De Kunst van de Claude Prompt: Specificiteit, Context en Verificatie	24
CLAUDE.md als Persistente Instructielaag	24
Few-shot Voorbeelden, XML-structurering en Role Assignment	24
Het “Interview Me” Patroon: Claude Vragen Stellen Laten voor het Coderen . . .	24
Prompt Chaining en State Management Over Sessies Heen	25
Veelvoorkomende Failure Patronen en Hoe Ze te Vermijden	25
Adversarial Testing: het Stress-testen van Claude Code’s Output	25
Hoofdstuk 5: Real-world Ontwikkelworkflows	26
Het Bouwen van een Full-stack Applicatie vanaf Nul (Hands-on Project)	26
Refactoring Legacy Codebases: Stripe’s Scala-naar-Java Migratie Case Study . . .	27
Complex Bugs Debuggen: Een Stap-voor-Stap Walkthrough	27
Code Migratie op Schaal: Wiz’s Python-naar-Go Migratie	28
Onboarding naar een Onbekende Codebase	28
Hoofdstuk 6: Testen, Verificatie en Kwaliteitszorg	29
Claude Iets Geven om Tegen te Verifiëren	29
Test-generatie en Self-healing Tests	29
Outcomes: Geautomatiseerde Rubric Grading	29
Het Writer/Reviewer Patroon met Parallele Sessies	29
Code Review Skills en Adversarial Review Strategieën	29
CI/CD Integratie: GitHub Actions, GitLab CI en Geautomatiseerde PRs	30
Hoofdstuk 7: Automatisering en Schalen	31
Headless Mode: Non-interactive Claude Code voor Scripts en Pipelines	31
Fan-out Patronen: Duizenden Bestanden Parallel Verwerken	31
Claude Code Routines: Geplande Taken en Event-driven Workflows	32
Pre-commit Hooks, Linting en Geautomatiseerde Code Kwaliteit Gates	32
Kostenmanagement en Token Budgettering op Schaal	32
Hoofdstuk 8: Geavanceerde Agentic Patronen	33
Subagents: het Delegeren van Investigatie en Verificatie	33
Skills: herbruikbare Domeinkennis en Workflows Creëren	33

MCP Servers: Verbinden met Externe Systemen (GitHub, Databases, Jira)	33
Hooks: Geautomatiseerde Policy Enforcement Over Sessies Heen	33
Multi-agent Orchestration: Lead Agents, Specialist Sub-agents, Gedeelde Bestandssystemen	34
Claude Managed Agents en de “Dreaming” Feature	34
Hoofdstuk 9: Implementatie en Productie-klaarheid	36
het Implementeren van Claude Code-gebouwde Applicaties naar Cloud Platforms	36
Omgeving-configuratie en Secret Management	36
Monitoring en Observabiliteit: OpenTelemetry, Usage Analytics, Token Tracking .	36
Prestatie-optimalisatie: Prompt Caching, Modelselectie, Context-efficiëntie	36
Van Prototype naar Productie: Een Case Study met Rakuten	36
Hoofdstuk 10: Team Collaboratie en Enterprise Implementatie	38
Claude Code op Schaal: Stripe’s 1.370-ingenieurs Implementatie	38
Managed Settings, Enterprise Security en Compliance	38
Gedeelde CLAUDE.md, Skill Libraries en Plugin Distributie	39
Claude Code Integreren in Bestaande Team Workflows	39
Hoofdstuk 11: Security Diepte-in-duik	40
Gedocumenteerde Kwetsbaarheden: CVE-2025-59536 en CVE-2026-21852	40
Prompt Injectie en Malicious Bestand-inhoud	40
Secrets Exposure in Context Loading	41
De —dangerously-skip-permissions Flag: Wanneer en Hoe Het Veilig te Gebruiken	41
Sandboxing: Docker, Dev Containers en VM Isolatie	41
Enterprise Security Architectuur: MDM, LLM Gateways, Agent Monitoring . . .	41
Hoofdstuk 12: De Toekomst van Agentic Coderen	42
Anthropic’s Roadmap: Dreaming, Outcomes, Multi-agent Orchestration	42
het Onderzoek naar Expertise: Waarom Domeinkennis Coding Background Verslaat	42
Acht Trends die Agentic Coderen Vormen in 2026 en Verder	42
de Veranderende Rol van Software-ingenieurs	42
Wat Dit Betekent voor de Toekomst van Werk en Kenniseconomie	43
Hoofdstuk 13: Conclusie: Een Agentic Ingenieur Worden	44
het Synthetiseren van de Reis van Novice naar Master	44
de Nieuwe Mentale Modellen van Softwareontwikkeling	44
je Claude Code Practice Bouwen: Een Persoonlijk Actieplan	44
Vooruitkijkend: Waar de Volgende Vijf Jaar Ons Naartoe Nemen	44

Referenties 45

De Complete Gids voor Agentic Softwareontwikkeling in 2026

Een Uitgehand Handboek voor het Beheersen van Claude Code, Multi-Agent Workflow en Enterprise Implementatie

Eerste Editie, Juli 2026

Dit boek dekt Claude Code tot en met versie 2.1.131 en Claude-modellen tot en met Opus 4.8, Fable 5, Sonnet 4.6 en Haiku 4.5. Alle prijzen, benchmarks en functiebeschrijvingen weerspiegelen de staat van het platform per juli 2026.

Over dit boek: Dit is een uitgebreide gids voor Claude Code, het agentic coderingssysteem van Anthropic dat fundamenteel heeft veranderd hoe software wordt gebouwd. Of je nu een ontwikkelaar bent die nieuw is in AI-assisted programmeren, of een ervaren ingenieur die multi-agent workflows wil beheersen, dit boek neemt je mee van de eerste installatie tot en met enterprise implementatie. Het dekt de architectuur van Claude Code, prompt-engineering-technieken, real-world case studies van Stripe en Wiz, CI/CD-automatisering, security-best practices en de toekomst van agentic coderen. Elk hoofdstuk bevat praktische voorbeelden, hands-on projecten en bruikbare aanwijzingen die gebaseerd zijn op onderzoek naar honderdduizenden echte Claude Code-sessies.

Inleiding: De Ingenieur die Niet Meer Typt

Eind 2025 maakte Boris Cherny, de maker en hoofd van Claude Code bij Anthropic, een observatie die nog maar drie jaar eerder als sciencefiction zou hebben geklonken. Hij had sinds de vorige november niet eens een enkele regel code handmatig geschreven. In plaats daarvan beschreef hij wat hij wilde, Claude Code bedacht hoe het te bouwen, voerde de tests uit, repareerde de fouten en stuurde pull requests door. Hij voerde vijf lokale terminalsessies en vijf tot tien remote instanties gelijktijdig uit, waarbij hij tien tot dertig pull requests per dag doorstuurde [1].

De maker van het tool beschrijft hoe het bedrijf dat het bouwde daagselijks daadwerkelijk werkt, en dit is slechts één datapunt in een veel groter verhaal.

Per mei 2026 werd meer dan 80 procent van de code die gemerged werd in Anthropic's eigen productiecodebase geschreven door Claude, een stijging van lage enkelfcijfers voordat Claude Code lanceerde in research preview in februari 2025 [7]. De ingenieurs bij Anthropic typen geen implementaties vanaf nul. Ze beschrijven wat ze willen, Claude Code bedacht hoe het te bouwen, voert de tests uit, repareert de fouten en stuurt de pull requests door. De menselijke ingenieurs reviewen de output, maken architectonische beslissingen en gaan verder naar het volgende probleem [8].

Wat Dit Boek Je Zal Leren

Dit boek is een uitgebreide gids voor Claude Code, het agentic coderingssysteem van Anthropic dat fundamenteel heeft veranderd hoe software wordt gebouwd. Of je nu een ontwikkelaar bent die nieuw is in AI-assisted programmeren, of een ervaren ingenieur die multi-agent workflows wil beheersen, dit boek neemt je mee van de eerste installatie tot en met enterprise implementatie. Het dekt de architectuur van Claude Code, prompt-engineering-technieken, real-world case studies van Stripe, Wiz en Rakuten, CI/CD-automatisering, security-best practices en de toekomst van agentic coderen. Elk hoofdstuk bevat praktische voorbeelden, hands-on projecten en bruikbare aanwijzingen die gebaseerd zijn op onderzoek naar honderdduizenden echte Claude Code-sessies.

Aan het einde van dit boek zul je in staat zijn om:

- Claude Code te installeren en configureren over meerdere platforms en IDEs
- Prompts schrijven die betrouwbaar productie-kwaliteitscode produceren bij de eerste poging
- Full-stack applicaties vanaf nul bouwen met behulp van het interviewpatroon en specificatie-gedreven ontwikkeling
- Test- en verificatielussen instellen zodat Claude zichzelf corrigeert zonder jouw tussenkomst
- Herhalende taken automatiseren met headless-modus, fan-out patronen en geplande routines
- Multi-agent architecturen ontwerpen met lead agents, specialistische sub-agents en gedeelde bestandssystemen
- Claude Code implementeren op enterprise-schaal met managed settings, SSO en governance-beleid
- De securityrisico's die inherent zijn aan agentic coderingstools te begrijpen en te beperken, inclusief prompt injectie, secrets-exposure en supply chain-aanvallen

Voor Wie Dit Boek Bestemd is

Dit boek gaat uit van basisbekendheid met softwareontwikkeling: je weet wat een terminal is, je hebt code geschreven in ten minste één taal, en je begrijpt concepten zoals versiebeheer, testen en implementatie. Je hoeft geen expert te zijn in een bepaalde programmeertaal. Claude Code werkt met talen, en de mentale modellen die dit boek leert zijn overdraagbaar ongeacht jouw technische stack.

Als je een junior ontwikkelaar bent die autocomplete-tools heeft gebruikt en nuttig maar beperkt vond, zal dit boek je tonen wat er gebeurt wanneer je verder gaat dan autocomplete. Als je een senior ingenieur of teamlead bent die Claude Code evalueert voor jouw organisatie, zal dit boek je de technische diepgang en strategische context geven die nodig is om geïnformeerde beslissingen te nemen. Als je een productmanager, ontwerper, oprichter of niet-technische professional bent die software wil bouwen zonder code regel voor regel te schrijven, zal dit boek je tonen hoe Claude Code die deur opent.

Hoe Dit Boek te Gebruiken

Het boek is georganiseerd in drie delen, elk opbouwend op het vorige:

Deel I (Hoofdstukken 1-3): Fundament. Hoofdstukken 1 en 2 leggen het conceptuele kader. Je zult begrijpen waar Claude Code past in de evolutie van AI-coderingstools, wat

zijn agentic architectuur anders maakt van autocomplete en inline-assistenten, en hoe zijn interne lus onder de motorkap werkt. Hoofdstuk 3 neemt je mee door de installatie en je eerste sessie.

Deel II (Hoofdstukken 4-8): Kernvaardigheden. Hoofdstukken 4 tot en met 6 leren je de praktische vaardigheden van het werken met Claude Code: prompt engineering, real-world ontwikkelworkflows en teststrategieën. Hoofdstukken 7 en 8 dekken automatisering, schalen en geavanceerde agentic patronen. Hier ontwikkel je de spiergeheugen die productieve sessies scheidt van frustrerende.

Deel III (Hoofdstukken 9-13): Productie en Verder. Hoofdstukken 9 en 10 adresseren implementatie, observabiliteit en enterprise-samenwerking. Hoofdstuk 11 is een diepte-in-duik in security. Hoofdstuk 12 kijkt naar de toekomst van agentic coderen. Hoofdstuk 13 synthetiseert alles tot bruikbare aanwijzingen om een effectieve agentic ingenieur te worden.

Je kunt dit boek van cover tot cover lezen, of springen naar de hoofdstukken die het meest relevant zijn voor je huidige behoeften. Elk hoofdstuk is ontworpen als een zelfstandige diepte-in-duik, terwijl het ook opbouwt op eerder geïntroduceerde concepten.

Het Bewijsbasis

De bewijsbasis voor dit boek omvat uitgebreide analyse van officiële Anthropic-documentatie en engineering-blogposts, het onderzoek van het bedrijf naar ongeveer 400.000 Claude Code-sessies van ruwweg 235.000 gebruikers tussen oktober 2025 en april 2026 [6], customer case studies gepubliceerd door Anthropic op claude.com/customers/, third-party technische analyses van Check Point Research en andere security-bedrijven, en community-gecontribueerde workflows. Waar claims niet onafhankelijk geverifieerd kunnen worden, worden ze toegeschreven aan hun bron. Waar het bewijs onbeslisbaar of betwist is, zegt de tekst dat.

Een Opmerking over Wat Dit Boek Niet Dekt

Dit boek focust op Claude Code als ontwikkeltool voor het werken met je eigen codebases. Het dekt niet de Claude API of het bouwen van applicaties die de Claude API gebruiken, hoewel de concepten in dit boek je een sterk fundament zullen geven om te begrijpen hoe de onderliggende modellen van Claude werken. Het dekt ook niet Claude Cwork (de collaboratieve werkruimte), Claude Design of Claude Security, die aparte producten zijn in Anthropic's portfolio.

De Centrale These

De centrale these van dit boek is straightforward: Claude Code vertegenwoordigt een paradigmaverschuiving van code completion naar agentic coderen. De mens wordt een orkestrator die doelen definieert en resultaten reviewt, terwijl Claude de uitvoering verzorgt. Maar deze paradigmaverschuiving komt met echte risico's: security-kwetsbaarheden in configuratiebestanden, contextbeheer-uitdagingen, kwaliteitsdegradatie-incidenten en de behoefte aan enterprise-grade governance. Dit tool beheersen vereist niet alleen het leren van commando's, maar ook het ontwikkelen van nieuwe mentale modellen voor delegatie, verificatie en samenwerking met een autonome agent.

Het Landschap in 2026

Claude Code is niet het enige agentic coderingstool dat beschikbaar is. Het competitieve landschap omvat GitHub Copilot, Cursor, OpenAI Codex, Google Antigravity en anderen, elk met distinct strategieën en sterktes. Het begrijpen van hun verschillen is essentieel voor het kiezen van het juiste tool voor je workflow. Maar Claude Code bezet een unieke positie: het is terminal-first maar draait over meerdere interfaces, het opereert op projectniveau in plaats van bestandsniveau, en het heeft de meest agressieve adoptie gedemonstreerd onder enterprise-teams.

De technologie beweegt zich snel. Claude Code is al door significante veranderingen gegaan sinds de lancering, met major feature releases in midden-2026 die Dreaming (cross-session geheugencuratie), Outcomes (geautomatiseerde kwaliteitswaardering) en multi-agent orchestration toevoegden op de 6 mei 2026 "Code with Claude" conferentie [3]. Het modelline-up omvat Claude Fable 5, die 95,0 procent scoort op SWE-bench Verified, Claude Opus 4.8 op 88,6 procent, en Claude Sonnet 4.6 als productie-standaard [2]. De kernprincipes van effectief gebruik, de mentale modellen van delegatie en verificatie, en het architectonische begrip van hoe Claude Code opereert, blijven constant ongeacht model-updates.

De volgende hoofdstukken bieden een complete gids voor het beheersen van Claude Code.

Hoofdstuk 1: De Agentic Revolutie in Softwareontwikkeling

Van Autocomplete naar Agents: Een Vijfjarige Evolutie

Om te begrijpen waar Claude Code past in het landschap, helpt het om de evolutie van AI-assisted coderen over ruwweg vijf jaar te traceren. De progressie is door vier distinct fasen gegaan, elk opbouwend op het vorige en uitbreidend wat mogelijk is.

De eerste fase, die ruwweg van 2019 tot 2021 liep, was rule-based code completion. Tools zoals TabNine gebruikten statistische modellen getraind op open-source code om de volgende regel of functie voor te stellen terwijl een ontwikkelaar typte. Deze tools waren werkelijk nuttig voor boilerplate en veelvoorkomende patronen, maar konden niet redeneren over intentie, architectuur of multi-file veranderingen. Ze voltooide regels; ze begrepen geen projecten.

De tweede fase, van 2021 tot en met ruwweg midden-2024, was large language model (LLM) autocomplete. GitHub Copilot, gelanceerd in 2021 met OpenAI's Codex-model, vertegenwoordigde een quantum leap. In plaats van statistische voorspelling gebruikte het een neurale taalmodel om code-suggesties te genereren. Ontwikkelaars konden een functie markeren en vragen om een testsuite, of beschrijven wat ze wilden in comments en werkende implementaties ontvangen. De ervaring was transformatief voor veel ontwikkelaars, maar bleef fundamenteel reactief. De ontwikkelaar typte, het model reageerde, en de ontwikkelaar accepteerde of verwierp elke suggestie.

De derde fase, die eind 2024 opkwam en versnelde door 2025 heen, introduceerde inline-assistenten en AI-native IDEs. Tools zoals Cursor (een VS Code-fork) en Claude's eigen integraties gingen verder dan single-file suggesties. Ze konden de volledige projectcontext lezen, begrijpen hoe bestanden met elkaar samenhangen, en gecoördineerde veranderingen maken over meerdere bestanden. De ontwikkelaar kon een vraag stellen over de codebase en een antwoord ontvangen dat gebaseerd was op de werkelijke code-structuur.

Claude Code, gelanceerd als een general-available tool op 22 mei 2025 naast Claude 4, vertegenwoordigt de vierde fase: agentic coderen. In tegenstelling tot zijn voorgangers wacht Claude Code niet af tot een suggestie geaccepteerd of verworpen wordt. Het ontvangt een

doel, plant een sequentie van acties, voert ze uit met echte ontwikkeltools (bestanden lezen, code bewerken, tests uitvoeren, git gebruiken), evalueert de resultaten en past zijn aanpak aan. De ontwikkelaar definieert het doel en reviewt het resultaat in plaats van elke stap te begeleiden [18].

De verschuiving van autocomplete naar agency verandert de fundamentele relatie tussen mens en machine. Met autocomplete doe je het werk en assisteert het tool met suggesties. Met een agentic systeem delegateer je werk aan een capabele collaborator die onafhankelijk plant, uitvoert en iterateert.

Anthropic's eigen data weerspiegelt deze evolutie. Een privacy-preserving analyse van ongeveer 400.000 Claude Code-sessies van oktober 2025 tot en met april 2026 toonde aan dat de samenstelling van het werk dramatisch is verschoven in die zeven maanden [6]. Het aandeel sessies besteed aan het repareren van gebroken code daalde van dertig-drie procent naar negentien procent, terwijl het bedienen van software groeide van veertien procent naar eenentwintig procent. Het schrijven van nieuwe code en analyseren van data verdubbelde ruwweg, van ongeveer tien procent naar twintig procent. De taken zelf werden ook waardevoller: de geschatte economische waarde van de gemiddelde sessie steeg met zeven-en-twintig procent.

Naarmate agents capabeler worden, stoppen ontwikkelaars ermee ze te gebruiken voor kleine fixes en beginnen ze ze te gebruiken voor end-to-end ontwikkeling, implementatie en operaties. Het tool schrijft niet alleen meer code; het voert pipelines uit, implementeert applicaties, analyseert data en genereert documentatie.

Wat Claude Code Anders Maakt: Terminal-First Agentic Architectuur

Het AI-coderingstool-landschap in 2026 heeft zich geconsolideerd rond meerdere dominante platforms, elk met een distinct strategie en set van sterktes. Het begrijpen van hun verschillen is essentieel voor het kiezen van het juiste tool voor je workflow.

GitHub Copilot, onderhouden door Microsoft en GitHub, is de meest breed compatibele optie. Het integreert met tien of meer IDEs, inclusief VS Code, JetBrains, Neovim, Xcode, Eclipse, Zed en SQL Server Management Studio. De sterkte ligt in distributie en ecosysteeminTEGRATIE. Copilot ondersteunt modellflexibiliteit over OpenAI (GPT-5.1 tot 5.4), Anthropic (Haiku 4.5, Sonnet 4.6, Opus 4.6), Google (Gemini 3 Pro en Flash) en xAI Grok Code. De enterprise tier voegt custom knowledge bases, pull request samenvattingen, fine-grained admin controls en SAML single sign-on toe. Tegen \$10 per maand voor individuele ontwikkelaars en \$19 per seat per maand voor Business tier teams blijft het de beste

waarde-optie voor basis AI-coderingsassistentie. De zwaktes zijn de strakkere koppeling aan het GitHub-ecosysteem en de beperkte autonome multi-step uitvoeringscapaciteit.

Cursor, ontwikkeld door Anysphere, is een volledig AI-native IDE gebouwd op een VS Code-fork. De sterkte ligt in diepte van integratie. Elke bewerkingslaag is AI-aware: tab-completions, Composer 2 voor multi-file agentic bewerkingen, background cloud agents en Bugbot voor geautomatiseerde pull request review. Cursor biedt de diepste AI-native IDE-ervaring tegen \$20 per maand voor individuele ontwikkelaars en \$40 per seat per maand voor teams. Het is SOC 2 Type 2 gecertificeerd met jaarlijkse penetration testing, SCIM seat management en SAML/OIDC single sign-on. De primaire beperking is dat het volledige teammigratie naar de Cursor IDE vereist, en de gratis Hobby tier sluit cloud agents en geavanceerde features uit.

Claude Code, ontwikkeld door Anthropic, kiest een fundamenteel andere aanpak. Het is terminal-first maar draait over meerdere interfaces: de CLI, VS Code-extensie, JetBrains-plugin, desktop app, webinterface op claude.ai/code, Slack en CI/CD-pipelines. Het bepalende kenmerk is agentic capaciteit. Claude Code opereert op projectniveau, leest de volledige codebase, plant over meerdere bestanden, voert veranderingen uit, draait tests en iterateert op failures. De ontwikkelaar definieert het doel en reviewt het resultaat [18].

De competitieve landschapsdata van begin 2026 toont een opvallende verschuiving in ontwikkelaarsvoorkeur. Claude Code behaalde een 46 procent “most loved” rating onder ontwikkelaars in de Pragmatic Engineer’s februari 2026 enquête van 15.000 software-ingenieurs, vergeleken met Cursor op 19 procent en GitHub Copilot op 9 procent [50]. Dit gaat niet alleen over features; het weerspiegelt een groeiende erkenning dat agency belangrijker is dan autocomplete.

De onderstaande vergelijkingstabel vat de belangrijkste verschillen samen:

Dimensie	Claude Code	GitHub Copilot	Cursor
Primaire interface	Terminal + IDE-extensies	IDE-plugins (10+ editors)	AI-native IDE (VS Code-fork)
Agentic capaciteit	Volledige multi-step autonomie	Inline suggesties + workspace	Multi-file bewerkingen, cloud agents
Modelopties	Alleen Anthropic-modellen	OpenAI, Anthropic, Google, xAI	Meerdere modelproviders

Dimensie	Claude Code	GitHub Copilot	Cursor
Externe toolintegratie	MCP-servers (open standaard)	GitHub-ecosysteem-extensies	Cursor-specifieke extensies
Slack-integratie	Native async coderingsworkflow	Beperkt	Geen
Enterprise security	HIPAA-ready, SSO, RBAC	SAML SSO, IP-indemniteit	SOC 2 Type 2, zero data retention
Individuele prijsstelling	\$20/maand (Pro)	\$10/maand (Pro)	\$20/maand (Pro)

De keuze tussen deze tools is niet noodzakelijkerwijs either/or. Veel grote engineering-teams draaien Copilot voor multi-IDE breedte terwijl ze Claude Code of Cursor inzetten voor high-leverage, complexe taken die diepe codebase-reasoning en autonome uitvoering vereisen. De cruciale factor is het matchen van het tool aan de taak: eenvoudige autocomplete profiteert van brede IDE-compatibiliteit, terwijl complexe multi-file refactoring en end-to-end feature-ontwikkeling profiteren van Claude Code’s agentic architectuur.

Boris Cherny’s Workflow: Hoe de Maker Eigenlijk Claude Code Gebruikt

Het beste bewijs voor hoe Claude Code gebruikt moet worden komt van de maker. Boris Cherny, die eerder bij Meta werkte waar hij verantwoordelijk was voor codekwaliteit over het hele bedrijf, leidt het Claude Code-team bij Anthropic. Zijn carrière-traject van codekwaliteit-engineering naar het bouwen van AI-coderingstools geeft hem een uniek perspectief op wat ontwikkelaars nodig hebben [49]. Sinds eind 2025 is honderd procent van zijn eigen code geschreven door Claude Code. Hij stuurt tien tot dertig pull requests per dag door en voert vijf lokale terminalsessies uit naast vijf tot tien remote instanties gelijktijdig [1].

Cherny’s workflow is systematisch, niet casual. Hij vermijdt customisatie en vertrouwt op Claude Code out-of-the-box. Elke lokale sessie gebruikt een distinct git checkout in plaats van branches of worktrees, wat conflicten tussen parallelle sessies voorkomt. Remote sessies worden achtergrondged via de & operator en shifted met --teleport, hoewel hij opmerkt dat tien tot twintig procent worden opgegeven vanwege onvoorziene scenario’s [2,29].

Voor coderen geeft Cherny de voorkeur aan Opus 4.5 (nu opgevolgd door Opus 4.8) met thinking ingeschakeld, waarbij hij superieure kwaliteit, betrouwbaarheid en tool-use capaciteiten noemt vergeleken met Sonnet. Hij merkt op dat hoewel Sonnet sneller kan zijn per stap, de overall throughput sneller is met Opus omdat de eerste poging vaker slaagt [2,29,47].

Kennisretentie wordt geïnstitutionaliseerd via een team-onderhouden `CLAUDE.md` bestand in git. Dit document logt fouten, style-conventies, design-richtlijnen en PR-templates om de AI iteratief te verbeteren. Cherny tagt collega's pull requests met `@.claude` om nieuwe inzichten te injecteren, waarbij hij het bestand op ongeveer 2,5k tokens houdt [2]. Dit is een cruciaal inzicht: `CLAUDE.md` is geen statisch configuratiebestand, maar een levend document dat verbetert door de tijd heen via collectieve team-ervaring.

Zijn proces begint met een iteratieve planingsfase met behulp van Plan mode voordat hij overschakelt naar auto-accept edits, wat typisch slaagt bij de eerste poging. Dagelijkse operaties zoals commits en PRs worden geautomatiseerd via slash commands opgeslagen in `.claude/commands/`, wat expliciet prompting minimaliseert. Een `/commit-push-pr` commando voert tientallen keren dagelijks uit door git status inline voor te berekenen om model-latency te reduceren. Om CI-failures van formatterings-inconsistenties te voorkomen, configureert hij een PostToolUse hook die automatisch `bun run format` of `|| true` uitvoert op WriteEdit matches [2,46].

Security wordt afgehandeld door specifieke veilige bash-commando's (bijv. `bun run build:*`, `cc:*`) te verlenen via `/permissions`, wat de behoefte aan `--dangerously-skip-permissions` grotendeels elimineert. De dangerous flag is exclusief gereserveerd voor sandboxed long-running taken om herhaalde onderbrekingen te voorkomen. Cherny benadrukt dat het verstrekken van een verificatie-feedbacklus (zoals het uitvoeren van testsuites of browser-automatisering) cruciaal is, waarbij de finale output-kwaliteit met twee tot drie keer wordt verhoogd. Hij merkt op dat Claude elke enkele change die gelanded is op `claude.ai/code` test met behulp van de Claude Chrome-extensie [2,48].

Deze gestructureerde aanpak zorgt ervoor dat code in review al gepolijst aankomt, waardoor ingenieurs zich kunnen focussen op sturen en architectonisch toezicht. Het patroon is duidelijk: plan eerst, automatiseer dan de routine-stappen, verifieer rigoureus en laat Claude de uitvoering verzorgen.

De Aandeling van Arbeid: Mens als Orkestator, Agent als Bouwer

Een van de belangrijkste bevindingen uit Anthropic's onderzoek naar 400.000 Claude Code-sessies is de duidelijke verdeling van arbeid die ontstaat tussen mens en agent [6]. In een typische sessie nemen mensen ongeveer zeventig procent van de planningsbeslissingen (wat te doen, welke aanpak te kiezen, wat als klaar telt) terwijl Claude ongeveer tachtig procent van de uitvoeringsbeslissingen neemt (welke bestanden te wijzigen, welke code te schrijven, welke commando's uit te voeren).

Deze verdeling is niet toevallig. Het weerspiegelt zowel menselijke voorkeur als agent-capaciteit. Mensen worden natuurlijk aangetrokken tot plannen en besluitvorming, want daar telt domeinkennis het meest. Een accountant die Claude instrueert om een reconciliatie-script te bouwen brengt kennis van de business rules, edge cases bij month-end close en de verificatiecriteria die succes definiëren. Claude brengt kennis van code-structuur, library APIs, testframeworks en debugging-technieken.

De data toont deze verdeling consistent over beroepen heen. Computer en Mathematical Occupations (de grootste gebruikersgroep) bereiken verified success in ongeveer dertig procent van hun sessies overall [6]. Gebruikers uit andere professies bereiken verified success ongeveer zes-en-twintig procent van de tijd. Onder sessies die code produceren, zijn die cijfers vier-en-dertig procent en negen-en-twintig procent respectievelijk. Elke een van de tien grootste non-software occupation groups landt binnen zeven percentagepunten van software-ingenieurs qua succesrate. Management occupations rangschikken zich feitelijk het hoogst op verified success, iets boven software engineering occupations.

Hoe groter de domeinkennis die een persoon meebrengt naar een sessie, hoe meer werk Claude doet per instructie. In typische novice-sessies zet elke prompt ongeveer vijf Claude-acties en ruwweg 600 woorden output in gang. Expert-sessies zetten actieketens in gang die meer dan twee keer zo lang zijn (12 acties) met vijf keer de output (3.200 woorden) [6]. Deze kloof verschijnt binnen elke soort werk en elke band van taakwaarde.

De vorm van de samenwerking is belangrijk voor hoe je over Claude Code nadenkt. Het is geen vervanging voor een ontwikkelaar. Het is een force multiplier voor domeinkennis. Een senior ingenieur die zijn eerste Rust-vraag stelt is een beginner in Rust. Een accountant die Python nog nooit heeft gebruikt maar Claude precies vertelt welke reconciliatieregels een Python-script moet afdwingen en de edge case vangt die het fout afhandelt bij month-end close is een expert in die taak [6]. Het tool beloont begrip van het probleemdomein, niet vaardigheid in een programmeertaal.

Training zou zich moeten focussen op het ontwikkelen van helderheid in

probleemdefinitie, verificatiecriteria en domeinkennis, in plaats van op het leren van nieuwe syntax of API-referenties. De meest effectieve Claude Code-gebruikers zijn niet noodzakelijk de beste coders; het zijn degenen die kunnen articuleren wat er gebouwd moet worden, waarom het belangrijk is en hoe te verifiëren dat het werkt.

Het Landschap van AI Coding Agents in 2026

Claude Code is niet het enige agentic coderingstool dat beschikbaar is. Het competitieve landschap omvat meerdere opmerkelijke spelers:

OpenAI Codex, de opvolger van het originele Codex-model dat early GitHub Copilot features aandreef, is geëvolueerd naar een standalone agentic systeem. Het opereert vergelijkbaar met Claude Code met multi-file editing en autonome uitvoering. OpenAI's sterkte ligt in zijn massieve ontwikkelaarsbasis en integratie met ChatGPT Workspace. De zwakte is een meer opaque prijsstellingsmodel voor enterprise-gebruik.

Google Antigravity 2.0, aangekondigd in begin 2026, maakt gebruik van Google's Gemini-modellen voor coderingstaken. Het integreert diep met Google Cloud en Vertex AI, wat het een natuurlijke keuze maakt voor teams die al geïnvesteerd zijn in het Google-ecosysteem. De benchmark-prestatie op SWE-bench blijft achter bij Claude Fable 5 en OpenAI Codex [8].

Devin, de originele autonome coding agent van Cognition Labs (overgenomen door Amazon), pioneerde het concept van een agent die volledige softwaretaken kan plannen en uitvoeren. Hoewel het de categorie vestigde, zijn de capaciteiten overtroffen door Claude Code op de meeste benchmarks per midden-2026.

Amazon Q Developer, ingebouwd in het AWS-ecosysteem, biedt agentic coderingscapaciteiten met diepe integratie naar AWS-services. Het is een sterke keuze voor teams waarvan de infrastructuur zwaar AWS-afhankelijk is.

De competitieve dynamiek in deze ruimte verschuift zich snel. Wat ooit een modelrace was (welke LLM is het slimst in coderen) is geworden een harness race (welk platform orchestreert het beste modelcapaciteiten, externe tools en menselijk toezicht). Anthropic's strategie op het mei 2026 "Code with Claude" evenement versterkte dit: er werden geen nieuwe modellen aangekondigd. In plaats daarvan werden vijf infrastructurele features gelanceerd: Dreaming, Outcomes, multi-agent orchestration, Claude Finance met tien pre-built agents en Add-ins [3,38]. Het bericht was duidelijk. De bottleneck voor productie agent-systemen is niet modelcapaciteit. Het is het scaffolding rond het model.

Waarom Nu? Marktadoptie en de Verschuiving naar Agentic Workflows

De versnelling van agentic coderingsadoptie in 2026 wordt aangedreven door meerdere convergerende factoren.

Ten eerste hebben modelcapaciteiten een drempel bereikt waar autonome agents betrouwbaar multi-step taken kunnen afhandelen. Claude Fable 5 scoort 95 procent op SWE-bench Verified [22], en Claude Opus 4.8 (uitgebracht op 28 mei 2026) levert substantieel beter agentic judgment met ongeveer vier keer minder over het hoofd geziene code-gebreken dan zijn voorganger [28]. Deze vertegenwoordigen een kwalitatieve verschuiving in betrouwbaarheid.

Ten tweede is de infrastructuur voor het draaien van agents op schaal gerijpt. Prompt caching bereikt een 92 procent prefix reuse rate over Claude Code-sessies, wat kosten reduceert met ongeveer 81 procent en sustained agent-gebruik economisch haalbaar maakt [26]. Het Model Context Protocol (MCP), nu beheerd door de Agentic AI Foundation, is geïmplementeerd door Anthropic, OpenAI Agents SDK, Cursor, Cloudflare Agents en Microsoft [17], waardoor een open standaard voor agent-to-tool integratie ontstaat.

Ten derde heeft marktadoptie een tipping point overschreden. Het aandeel GitHub-projecten met coding agent-activiteit is meer dan verdubbeld sinds eind 2025 [6]. Claude Code overschreed \$2,5 miljard aan annualized revenue per februari 2026 [8]. Bij Anthropic zelf wordt tachtig procent van de nieuwe productiecode nu geschreven door Claude Code [7]. Community-momentum trackt naast enterprise-adoptie: de Claude Code repository op GitHub passeerde 127.000 stars per midden-2026 [25].

Ten vierde is het economische incentive overtuigend. De case studies van Stripe, Wiz, Ramp en Rakuten demonstreren productiviteitswinsten die moeilijk te negeren zijn. Een team dat een 50.000-regelige Python-bibliotheek kan migreren naar Go in twintig uur in plaats van twee tot drie maanden heeft fundamenteel zijn capaciteitsbeperkingen veranderd. Een bedrijf dat feature-delivery reduceert van vier-en-twintig dagen naar vijf heeft zijn productontwikkelingscyclus met ongeveer tachtig procent gecomprimeerd.

De verschuiving naar agentic coderen is geen tijdelijke trend of hype-cyclus. Het vertegenwoordigt een structurele verandering in hoe software wordt gebouwd, vergelijkbaar met de transitie van punch cards naar keyboards of van assembly language naar high-level languages. De tools veranderen, de workflows veranderen en de vaardigheden die het meest tellen veranderen.

De vraag is niet langer óf agentic coderen mainstream zal worden, maar hóe teams zich effectief kunnen aanpassen. Dit boek streeft ernaar de technische diepgang en strategische

context te verstrekken die nodig is om die vraag te beantwoorden.

Kritische Analyse: Trade-offs en Beperkingen van Agentic Tools

Voordat we duiken in hoe Claude Code effectief te gebruiken, is het de moeite waard te onderzoeken wat agentic coderingstools níet kunnen en waar ze nieuwe risico's introduceren buiten traditionele ontwikkelworkflows. Het begrijpen van deze limieten voorkomt kostbare fouten.

Context rotation is echt. Zelfs met een 1M token-window zullen sessies die uitgebreide tool-output, error logs en multi-file diffs accumuleren uiteindelijk auto-compaction triggeren. Wanneer compaction plaatsvindt, kan Claude het spoor verliezen van eerdere instructies of subtiele context die in de eerste helft van een sessie werd vastgesteld. Anthropic's april 2026 kwaliteitsdegradatie-incident demonstreerde dit levendig: een caching-bug die reasoning history op elke turn veegde veroorzaakte vergeetachtigheid, herhaling en vreemde tool-keuzes over duizenden sessies [12]. Het postmortem van 23 april 2026 onthulde dat drie separate product-layer veranderingen de degradatie veroorzaakten tussen 4 maart en 20 april 2026: een default reasoning effort verschuiving van high naar medium (4 maart), een caching-optimalisatie bug die reasoning op elke turn verwierp in plaats van één keer per idle uur (26 maart), en een system prompt verbosity-reductie die tool call text beperkte tot minder dan vijf-en-twintig woorden (16 april) [12,33,34]. Alle drie werden gerepareerd op 20 april in versie 2.1.116. De les is niet dat Claude Code onbetrouwbaar is; het is dat zelfs een enkele subtiele verandering in de harness rond een LLM kan cascade naar meetbare kwaliteitsdegradatie over een grote gebruikersbasis.

Claude Code kan niet betrouwbaar taken afhandelen die het niet getest zijn op. SWE-bench meet het vermogen van een model om bekende bugs te repareren in open-source repositories. Het meet niet het vermogen van een model om een onbekende applicatie vanaf nul te bouwen. Claude Code kan beide doen, maar de confidentieniveaus verschillen significant. De 95 procent SWE-bench Verified score voor Fable 5 weerspiegelt prestaties op een specifieke benchmark met bekende testcases en een goed gedefinieerd failure-sigitaal. Het bouwen van een nieuwe applicatie vanuit een high-level beschrijving omvat ambiguous requirements, architectonische beslissingen en trade-offs die geen benchmark vangt.

Het begrijpen van de kloof tussen benchmark-prestatie en real-world capaciteit vereist het begrijpen van hoe SWE-bench werkt. Elke SWE-bench taak is een GitHub-issue van een echt open-source project, gepaard met de repository-status op het moment dat de issue werd ingediend en een set bestaande tests die definiëren of een reparatie correct is. Het model

ontvangt de issue-beschrijving, verkent de repository en probeert een code-verandering te schrijven die de bestaande tests laat slagen. De benchmark heeft drie eigenschappen die het makkelijker maken dan real-world ontwikkeling: het probleem is goed gedefinieerd (de issue beschrijft een specifieke bug), het succes criterium is deterministisch (tests slagen of falen), en de codebase is bekend (het model getraind is op open-source code die deze repositories omvat).

Real-world ontwikkeling mist alle drie eigenschappen. Wanneer je Claude vraagt om “een taakbeheer-API te bouwen”, zijn de requirements ambiguous (welke velden zou een taak moeten hebben? welk authenticatiemodel?), is het succes criterium subjectief (voelt de API goed? is de error-handling adequaat?), en kan de codebase volledig nieuw zijn (Claude heeft je project nog nooit gezien). De 95 procent SWE-bench score betekent niet dat Claude 95 procent van de tijd zal slagen op echte ontwikkelingstaken. Het betekent dat Claude 95 procent van de tijd slaagt op een specifieke klasse van goed gedefinieerde bug-fixing taken met deterministische testsuites.

Dit onderscheid is belangrijk voor architectonische beslissingen. Als je Claude Code gebruikt om bugs te repareren in een bestaande codebase met een uitgebreide testsuite, kun je hoog vertrouwen hebben in de output. Als je Claude Code gebruikt om een nieuw systeem vanaf nul te ontwerpen en implementeren, heb je significant meer menselijk toezicht, verificatie en iteratieve verfijning nodig. De benchmark-score is een nuttig signaal van raw coding capaciteit, maar het is geen betrouwbare predictor van real-world succes op nieuwe taken.

Claude Code kan hallucineren in multi-step ketens. Een enkele stap kan correcte output produceren, maar fouten compileren over een keten van tien of twintig stappen. Een bestand dat gelezen moest worden in stap drie kan in stap vijf bewerkt zijn, en Claude’s begrip van de codebase-status wordt verouderd. Daarom is verificatie bij elke stap essentieel. Onderzoek door Huang et al. (2026) testte zeven veelgebruikte MCP-clients inclusief Claude Code en vond dat tool-poisoning-aanvallen malicious instructies kunnen injecteren via data teruggekomen van verbonden externe systemen [14]. Wanneer Claude een GitHub README leest, een database-query resultaat of een Slack-bericht dat embedded instructies bevat, worden die instructies onderdeel van zijn context en kunnen ze zijn gedrag manipuleren.

Claude Code kan geen beslissingen nemen over business requirements. Het kan implementeren wat je vraagt, maar het kan niet bepalen of een feature moet bestaan, welk API-design optimaal is, of welke trade-offs acceptabel zijn. Dit zijn menselijke oordelen. Het onderzoek naar 400.000 sessies bevestigt dit: mensen nemen ruwweg zeventig procent van de planningsbeslissingen terwijl Claude ongeveer tachtig procent van de uitvoeringsbeslissingen afhandelt [6].

Claude Code introduceert echte securityrisico’s in je ontwikkelomgeving. Zoals

gedocumenteerd door Check Point Research, stond CVE-2025-59536 remote code execution toe via malicious project configuratiebestanden voordat er een trust-dialog verscheen, en CVE-2026-21852 maakte API-key-diefstal mogelijk via ANTHROPIC_BASE_URL manipulatie [11]. Dit zijn gedocumenteerde kwetsbaarheden die echte implementaties beïnvloedden en demonstreerden hoe configuratiebestanden, traditioneel behandeld als passieve metadata, uitvoerbare aanvalsvector kunnen worden.

Buiten Claude Code om staat het bredere MCP-ecosysteem voor systemische security-uitdagingen. Een vulnerability advisory van Ox Security (april 2026) identificeerde dat IDEs en coding assistants inclusief Windsurf, Claude Code, Cursor, Gemini CLI en GitHub Copilot kwetsbaar zijn voor command injection via hun MCP JSON configuratiebestanden [17]. Het enige uitgegeven CVE op dat moment was voor Windsurf, maar het aanvalspatroon geldt over het ecosysteem heen. NIST karakterizeert prompt injectie als “generative AI’s grootste security-gebreke” en OWASP rangschikt het als nummer één vulnerability in hun LLM Applications Top 10 [35].

Claude Code heeft gebruiksbeperkingen en compute-constraints. In begin 2026 rapporteerden veel Pro plan-gebruikers dat ze gebruiksbeperkingen bereikten binnen tien tot vijftien minuten van Sonnet-gebruik, met Opus dat kort daarna onbeschikbaar werd [27]. Anthropic verdubbelde gebruiksbeperkingen in mei 2026 als reactie op compute-bottlenecks, maar de fundamentele constraint blijft: Claude Code is een gedeelde resource en zwaar gebruik zal uiteindelijk rate-limits raken [39]. Het Pro plan omvat \$20 per maand aan agent credits, Max 5x omvat \$100 en Max 20x omvat \$200 [2]. Voor sustained ontwikkelwerk vinden de meeste teams dat upgraden naar Max 5x of hoger noodzakelijk is om onderbrekingen te voorkomen.

Claude Code kan menselijke code review niet vervangen. Elke AI-gegenereerde verandering zou door hetzelfde reviewproces moeten gaan als developer-geschreven code. Elke AI-gegenereerde verandering die security-critical bestanden aanraakt, zou twee reviewers moeten vereisen. De agent is een collaborator, geen autonome shipper [20]. Dit principe werd versterkt door Stripe’s ervaring: Scott MacVicar, die het developer infrastructure-team van Stripe leidt, benadrukte dat het behandelen van AI als vervanging in plaats van collaborator de grootste uitdaging is bij adoptie [8].

Vendor lock-in en ecosysteem-afhankelijkheid. Claude Code is ontworpen om het beste te werken met Anthropic-modellen en integreert strak met het Claude-platform. Hoewel het open standaarden zoals MCP gebruikt, draagt het migreren van Claude Code naar een andere agent of terug naar traditionele ontwikkeling echte kosten: geleerde workflows, CLAUDE.md bestanden, custom skills, hooks en MCP-server configuraties zijn allemaal specifiek voor Claude Code. Teams moeten evalueren of de productiviteitswinsten opwegen tegen het lock-in risico, vooral voor projecten met lange lifecycles.

Deze beperkingen zijn geen zwaktes van Claude Code per se. Het zijn kenmerken van elk systeem dat werk delegeert aan een LLM-based agent. Het begrijpen ervan is de eerste stap naar effectief en veilig gebruik van Claude Code.

Hoofdstuk 2: Anatomie van Claude Code

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Agentic Lus: Verzamelen, Handelen, Verifiëren

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Agentic Lus Tracen: Een Complete Terminal Walkthrough

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Wat de Agentic Lus Anders Maakt dan Autocomplete

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Het Onderbreken en Herdirigeren van de Lus

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Modellen Achter de Schermen: Sonnet, Opus, Fable en Modelselectie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Modellen Kiezen: Een Praktisch Beslisframework

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Token Overhead van Tool Definities

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Token Management Math: Je Uitgaven Begrijpen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Ingebouwde Tools: Bestandsoperaties, Zoeken, Uitvoering, Web, Code-inzicht

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De .claude Directory: Configuratie, CLAUDE.md en Geheugen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Context Windows: Hoe Ze Vullen, Auto-compactie en Token Management

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoe Context Vult: Een Token Budget Walkthrough

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoe Auto-compactie Onder de Motorkap Werkt

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Controleren Wat Compactie Overleeft

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Strategien voor Effectief Contextbeheer

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 3: Aan de Slag en Op Zetten

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Installatie Over Platforms: macOS, Linux, Windows, WSL, Homebrew, WinGet

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Claude Code Desktop Applicatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Veelvoorkomende Installatieproblemen Oplossen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Authenticatie en Account Setup

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Je Eerste Sessie: Walkthrough van Zero naar Code

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Terminal Sessie Trace: Je Eerste Interactie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Stap-voor-Stap Configuratie: .claudeignore, settings.json en Permission Modes

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Permission Modes: Default, Auto-accept, Plan en Auto Mode

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

IDE Integraties: VS Code, JetBrains, Remote Control, Webinterface

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 4: Prompt Engineering voor Claude Code

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Kunst van de Claude Prompt: Specificiteit, Context en Verificatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

CLAUDE.md als Persistente Instructielaag

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Few-shot Voorbeelden, XML-structurering en Role Assignment

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Waarom Deze Technieken Werken: De Mechanica Achter Prompt Engineering

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Het “Interview Me” Patroon: Claude Vragen Stellen Laten voor het Coderen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Prompt Chaining en State Management Over Sessies Heen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Veelvoorkomende Failure Patronen en Hoe Ze te Vermijden

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Adversarial Testing: het Stress-testen van Claude Code's Output

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 5: Real-world Ontwikkelworkflows

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Het Bouwen van een Full-stack Applicatie vanaf Nul (Hands-on Project)

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fase 1: Verkenning en Specificatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fase 2: Implementatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fase 3: Frontend Integratie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fase 4: het Debuggen van een Echte Bug

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fase 5: Adversarial Testing

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Context Compactie Mid-Implementatie Afhandelen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Wat Deze Workflow Je Leert

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Refactoring Legacy Codebases: Stripe's Scala-naar-Java Migratie Case Study

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Stripe Case Study: Diepte-in-duik

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Migratieworkflow op Schaal

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De Review Bottleneck

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Complex Bugs Debuggen: Een Stap-voor-Stap Walkthrough

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Code Migratie op Schaal: Wiz's Python-naar-Go Migratie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Onboarding naar een Onbekende Codebase

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 6: Testen, Verificatie en Kwaliteitszorg

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude iets Geven om Tegen te Verifiëren

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Test-generatie en Self-healing Tests

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Outcomes: Geautomatiseerde Rubric Grading

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Een Outcomes Rubric Schrijven: Een Stap-voor-Stap Walkthrough

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Het Writer/Reviewer Patroon met Parallele Sessies

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Code Review Skills en Adversarial Review Strategieën

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

CI/CD Integratie: GitHub Actions, GitLab CI en Geautomatiseerde PRs

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Production-Grade CI/CD Configuratie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Geautomatiseerde PR Reviews met Conditionele Executie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Geautomatiseerde Pull Request Creatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Security Overwegingen in CI/CD

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Compliance en Audit Requirements

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 7: Automatisering en Schalen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Headless Mode: Non-interactive Claude Code voor Scripts en Pipelines

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fan-out Patronen: Duizenden Bestanden Parallel Verwerken

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Sequentieel Fan-out (Eenvoudig)

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Parallel Fan-out (High-throughput)

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fan-out met Result Aggregatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Fan-out met Concurrentie Control en Backoff

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Kostenanalyse voor Fan-out Patronen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Code Routines: Geplande Taken en Event-driven Workflows

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Pre-commit Hooks, Linting en Geautomatiseerde Code Kwaliteit Gates

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Kostenmanagement en Token Budgettering op Schaal

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 8: Geavanceerde Agentic Patronen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Subagents: het Delegeren van Investigatie en Verificatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Subagent Token Economie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Skills: herbruikbare Domeinkennis en Workflows Creëren

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

MCP Servers: Verbinden met Externe Systemen (GitHub, Databases, Jira)

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hooks: Geautomatiseerde Policy Enforcement Over Sessies Heen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Multi-agent Orchestration: Lead Agents, Specialist Sub-agents, Gedeelde Bestandssystemen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Multi-agent Orchestration Configureren: Stap-voor-Stap

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Merge Conflicten Oplossen in Multi-agent Workflows

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Observability Data Parsen van Multi-agent Runs

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Managed Agents en de “Dreaming” Feature

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Dreaming: Self-improving Agents

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Outcomes: Geautomatiseerde Kwaliteit Grading

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Managed Agents Prijsstelling en Implementatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Multi-agent Orchestration: Patronen en Failure Modes

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Finance en Add-ins

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 9: Implementatie en Productie-klaarheid

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

het Implementeren van Claude Code-gebouwde Applicaties naar Cloud Platforms

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Omgeving-configuratie en Secret Management

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Monitoring en Observabiliteit: OpenTelemetry, Usage Analytics, Token Tracking

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Prestatie-optimalisatie: Prompt Caching, Modelselectie, Context-efficiëntie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Van Prototype naar Productie: Een Case Study met Rakuten

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Rakuten's "AI-nization" Strategie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Technische Implementatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Parallele Ontwikkeling op Schaal

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Toekomstige Trajectorie: Ambient Agents

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 10: Team Collaboratie en Enterprise Implementatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Code op Schaal: Stripe's 1.370-ingenieurs Implementatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Managed Settings, Enterprise Security en Compliance

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Identity en API Key Governance

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Gecentraliseerde Traffic Routing

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

MCP Tool Governance

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Locale Sandbox en Permission Enforcement

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

End-to-End Enterprise Implementatie Walkthrough

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Compliance Frameworks

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Role-Based Access Control

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Gedeelde CLAUDE.md, Skill Libraries en Plugin Distributie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Claude Code Integreren in Bestaande Team Workflows

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 11: Security

Diepte-in-duik

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Gedocumenteerde Kwetsbaarheden: CVE-2025-59536 en CVE-2026-21852

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Prompt Injectie en Malicious Bestand-inhoud

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Prompt Injectie via Bestanden

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

CI/CD Prompt Injectie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Privacy Safeguards

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Aanvullende Safeguards

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

NIST en OWASP Classificaties

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Secrets Exposure in Context Loading

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

De —dangerously-skip-permissions Flag: Wanneer en Hoe Het Veilig te Gebruiken

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Sandboxing: Docker, Dev Containers en VM Isolatie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Enterprise Security Architectuur: MDM, LLM Gateways, Agent Monitoring

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 12: De Toekomst van Agentic Coderen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Anthropic's Roadmap: Dreaming, Outcomes, Multi-agent Orchestration

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

het Onderzoek naar Expertise: Waarom Domeinkennis Coding Background Verslaat

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Acht Trends die Agentic Coderen Vormen in 2026 en Verder

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Vendor Lock-in en Ecosysteem-afhankelijkheid

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

de Veranderende Rol van Software-ingenieurs

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

het Juridische en Ethische Landschap

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Wat Dit Betekent voor de Toekomst van Werk en Kenniseconomie

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Hoofdstuk 13: Conclusie: Een Agentic Ingenieur Worden

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

het Synthetiseren van de Reis van Novice naar Master

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

de Nieuwe Mentale Modellen van Softwareontwikkeling

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

je Claude Code Practice Bouwen: Een Persoonlijk Actieplan

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Vooruitkijkend: Waar de Volgende Vijf Jaar Ons Naartoe Nemen

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.

Referenties

Deze inhoud is niet beschikbaar in het sample boek. Het boek kan aangekocht worden op deze Leanpub link: <https://leanpub.com/claudecodenl>.