



CURATED STREAM

Understanding the Role of AI in Coding

How modern AI is reshaping software development — tools, techniques, and what it means for your team

Nelson AI Sandbox · 2026

Why AI matters in coding today

Introduction

- AI is no longer a futuristic concept — it is part of the daily toolkit for software developers worldwide.
- Modern coding assistants like Claude Code, GitHub Copilot and Cursor accelerate writing, reviewing and refactoring code.
- In this stream we explore the technologies behind these tools, where they shine, and how to use them responsibly.

55%

Faster task completion with AI assistants (GitHub research)

76%

Of developers use or plan to use AI tools (Stack Overflow 2024)

80.8%

Claude Opus 4.6 score on SWE-bench Verified (2026)

Evolution of AI in software development

From scripts to agents



AI in coding at a glance

AI supports developers at every stage of the lifecycle



Design

Architecture sketches,
schema drafts



Write

Autocomplete,
scaffolding, refactors



Review

Inline review, security
flags



Deploy

CI/CD prediction, infra-
as-code



Monitor

Log triage, anomaly
detection

Teams typically use a stack of AI tools — Claude Code for agentic refactors, Copilot for inline, Snyk for security, and so on.

Types of AI technologies used in coding

The techniques behind the tools



Machine learning

Statistical models trained on past code predict the next token, flag likely bugs, and rank suggestions.



Natural language processing

Transformers translate plain-English instructions into runnable code and generate human-readable docs from source.



Generative AI

Large models like Claude Opus 4.8 and GPT-5.6 generate whole functions, refactors, tests and architectures from a prompt.



Agentic systems

Tool-using agents that plan, run shell commands, read files, and verify their own work autonomously over many steps.

Case studies: AI in action

Reported outcomes from teams adopting AI coding tools



Fintech engineering team · GitHub Copilot

~26% reduction in cycle time on routine pull requests in a controlled internal study



Anthropic engineering · Claude Code

Whole-codebase refactors that previously took days now run as overnight agent jobs



E-commerce platform · Snyk + AI scanning

Critical vulnerabilities surfaced in real time inside the IDE before merge

Outcomes vary by team, codebase and discipline. The constant: AI adds the most value when paired with strong review and testing practices.

Machine learning in coding

Predicting patterns and ensuring quality

Predict the next token

Models trained on billions of lines of code rank the most likely next line, function name or argument.

Flag risky patterns

Surface common pitfalls — mismatched brackets, null-pointer paths, off-by-one loops — before runtime.

Suggest optimisations

Spot inefficient queries, redundant loops or memory leaks and suggest cleaner alternatives.

How it learns

- 1 Ingest:** Public + permissioned source code
- 2 Train:** Predict held-out tokens; reward correct guesses
- 3 Fine-tune:** Human feedback shapes style and safety
- 4 Serve:** Suggestions ranked in your editor

Natural language processing

Bridging human language and code

You write

"Write a Python function that fetches a URL with retries and returns the JSON body."

The assistant returns

```
def fetch_json(url, retries=3):
    for attempt in range(retries):
        try:
            r = httpx.get(url, timeout=10)
            r.raise_for_status()
            return r.json()
        except httpx.HTTPError:
            if attempt == retries - 1:
                raise
            time.sleep(2 ** attempt)
```



Code generation

Plain English to runnable code in any major language



Code search

Find functions by what they do, not just their name



Explanation

Summarise a function or a whole file in plain language



Translation

Port code between languages (Python ↔ Go ↔ TypeScript)

Generative AI for rapid prototyping

Creating working software with minimal input

Skip the boilerplate

Generate API scaffolds, database schemas, test stubs and even whole React components from a single prompt.

Try many ideas fast

Spin up three different prototypes in an afternoon to test which approach feels right.

Refine, don't reinvent

Developers shift from typing to reviewing — focusing energy on architecture, edge cases and product judgement.

Frontier models (2026)

Claude Opus 4.8

Complex agentic coding · 1M context

Claude Sonnet 5

Fast, high-intelligence coding work

GPT-5.6

1M context · strong knowledge work

Gemini 3.5 Flash

Stable Gemini 3 model · agentic coding

Llama 4 Scout

Open weights · 10M-token context

How AI assists in the coding process

Streamlining development end-to-end



Autocomplete

Editor inline suggestions (Copilot, Cursor Tab, Supermaven)



Debugging

Read stack traces, suggest fixes, run repro scripts (Claude Code)



Security scanning

Detect SQL injection, secrets, dependency CVEs (Snyk, CodeQL)



Testing

Generate unit tests, fuzz inputs, surface flaky tests



CI/CD

Predict deploy failures, generate IaC, summarise diffs in PRs



Code review

Inline review comments alongside human reviewers

Collaborative AI tools

AI as a partner across the team



Shared context

Cursor and Claude Code can read your team's whole repo, docs and recent PRs to give consistent suggestions across the team.



Pair programming

Treat the AI as a junior teammate — bounce designs off it, review its work, give it the tedious parts of the task.



Convention enforcement

Tools learn project-specific patterns (naming, error handling, lint rules) and nudge new contributors toward them.

Common tool stacks

Claude Code

Terminal-native agent; whole-codebase edits

Cursor

VS Code fork; ~72% Tab acceptance rate

GitHub Copilot

Inline + chat across most major IDEs

Windsurf • Replit Agent

Browser-first alternatives

Benefits of AI in coding

Why developers love these tools



Faster cycles

Less boilerplate · faster scaffolding · shorter PR review queues



Fewer defects

AI-assisted review catches a category of bug humans tend to skim past



Continuous learning

Junior devs absorb conventions and idioms by reading AI suggestions



Lower switching cost

Polyglot work gets easier — AI ports patterns between languages

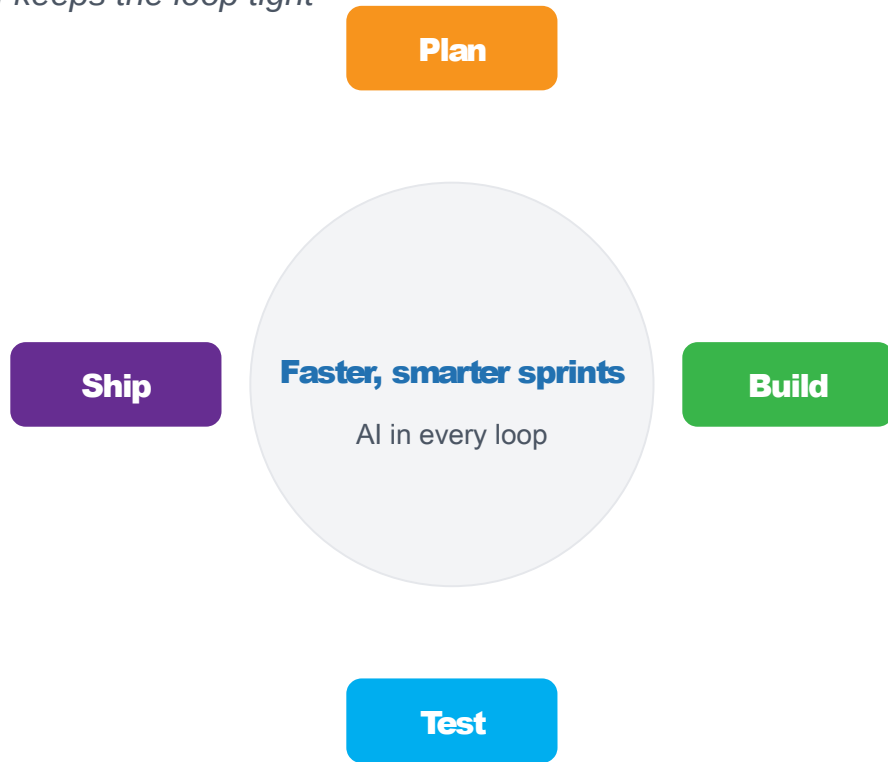
Challenges and ethical considerations

Navigating the pitfalls

Challenge	Mitigation
 Model bias	 Diverse training data and ongoing fairness audits
 Over-reliance / skill atrophy	 Treat AI as draftsman, not author — keep human review
 Insecure generated code	 Mandatory security scanning (Snyk, CodeQL) on every PR
 Licensing of AI output	 Use enterprise tiers with indemnity; track provenance
 Prompt injection (OWASP #1)	 Treat external content as untrusted; run AI in sandboxes

The role of AI in agile development

AI keeps the loop tight



Where AI helps most

- Backlog grooming — suggest priorities
- Estimation — flag suspect points
- Standup digest — summarise diffs since yesterday
- Retros — cluster feedback themes
- Velocity tracking — anomalous slowdowns flagged early

The future of AI in coding

What lies ahead



Self-healing systems

Services that detect their own faults, patch them, run regression tests, and roll forward without human intervention.



Always-on agents

Background agents triaging issues, drafting PRs and updating dependencies overnight while you sleep.



Regulated AI in code

EU AI Act fully applicable from 2 August 2026 — high-risk AI components in software need conformity assessments.



Spec-first development

Developers write rigorous specs in plain language; agents implement, test and ship them.

AI in open-source contributions

Driving innovation in shared codebases

Maintainers gain leverage

AI triages incoming issues, drafts reproduction scripts and proposes initial patches.

Newcomers get a friendlier onramp

AI explains the codebase, suggests good first issues, and reviews early contributions before they reach a human.

Open weights matter

Llama 4 Scout (10M-token context, open weights) is increasingly used to build domain-specific assistants on top of niche codebases.

Open source +

AI

amplifies maintainer capacity, lowers contributor barriers, and keeps niche projects alive.

Case study: AI-powered debugging

Faster, smarter bug fixes

1

Paste the stack trace

Drop the failing log into Claude Code or Copilot Chat

2

Agent reads context

Tool follows the trace into your files, reads related code

3

Hypothesis + fix

Proposes the root cause and a candidate patch

4

Run the test

Agent executes the failing test to confirm the fix



Why it works

- Whole-codebase context — the agent sees what your IDE-only autocomplete can't
- Can run shell commands to verify, not just suggest
- Iterates on its own output until the test passes

Education and upskilling with AI

Learning and growing alongside AI

Personalised practice

Platforms like HackerRank and LeetCode tailor challenges to your weak spots and give code-aware feedback.

Learning mode in assistants

Claude's Learning mode and ChatGPT Study Mode coach via Socratic questions — not just answers.

AI in the classroom

Claude for Education partnerships at universities, plus official NZ Digital Government guidance for the public service.

By the numbers

92%

of students used AI tools by 2025

39%

of student AI chats are about creating learning content (Anthropic 1M-chat study)

44%

of NZ AI adopters cite education as a barrier (MBIE)

Summary and next steps

Key takeaways and the road ahead

AI is now a daily coding tool

Not a novelty — it accelerates writing, review, debugging and ops across the lifecycle.

Use it with judgement

Human review, security scanning and clear ownership of AI-authored code are non-negotiable.

Stay curious

The frontier moved twice this year — keep experimenting, keep learning, share what you find.

This week, try...

- ✓ Install Claude Code or Cursor
- ✓ Run an agentic refactor on a small repo
- ✓ Add Snyk or CodeQL to one project
- ✓ Join an AI-focused dev community
- ✓ Bring one experiment to your next NAIS session

Embrace the future of coding

AI is not just a tool — it's a partner. Let's build the next era of software together.

Join a workshop

aisandbox.org.nz/workshops

Start the conversation

kia_ora@aisandbox.org.nz

Share what you build

with the NAIS community