



Gemini Code Assist Workshop

ETLS 2025

Google Cloud





Simon Margolis
Associate CTO, AI&ML
SADA, an Insight Company



8x WINNER GOOGLE CLOUD PARTNER OF THE YEAR



Google was born as a data & AI company...

Organize the **world's information (Data)** and make it universally **accessible and useful (Intelligence)**



...and is a world leader in applying AI & ML to real-world situations

Search	Translate	Photos	Gmail
--------	-----------	--------	-------

- Search ranking
- Speech recognition
- Text, graphic and speech translations
- Photos search & smart sharing
- Smart reply
- Spam classification

Self Driving Car	Data Center Power Usage	AlphaGo	YouTube
------------------	-------------------------	---------	---------

- 25M miles driven
- Reduced cooling energy
- First AI to beat a world Go champion (2016)
- Video recommendations
- Better thumbnails



People

One Google

Sparring partner

Partnership & trust

10x Ideation

DEI & culture

A deep history of research and innovation at Google



2017
Transformer

Google invents Transformer kickstarting LLM revolution



2018
BERT

Google's groundbreaking large language model, BERT



2019
T5

Text-to-Text Transfer Transformer LLM 10B P model open sourced



2020
LaMDA

Google LaMDA model trained to converse



2021
AlphaFold

AlphaFold predicts structures of all known proteins



2022
PaLM

Industry leading large language model



2023
Bard

A conversational AI Service powered by LaMDA. (now known as Gemini)

Responsible AI at the foundation

2024 was all about rapid AI innovation

Google Agentspace



Q1
Evaluations



Q2
Gemini 1.0

2M
tokens

Imagen

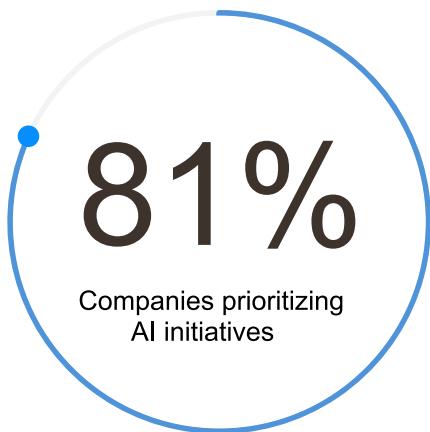


NotebookLM

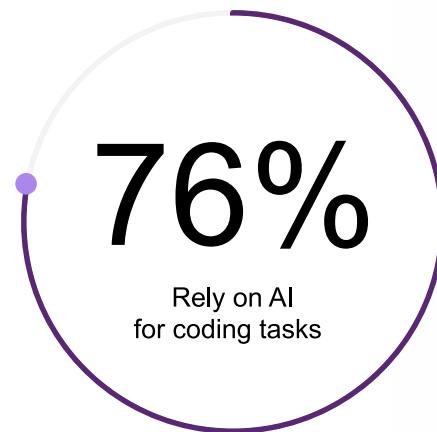
Q3
Gemini 1.5
Imagen

Q4
Gemini 2.0 is #1
NLM & Veo go viral
Agentspace is GA

AI is transforming software development



Leadership feels they **have to change**



Developers **want** to change



Developer **feel** more productive

<https://dora.dev/dora-report-2024>

“... We are using AI for coding, we are continuing to make a lot of progress there in terms of people using coding suggestions. I think, the last time I said a number, it was, like, 25% of code that's checked in involves people accepting AI-suggested solutions. That number is well over 30% now.”

Sundar Pichai

CEO, Alphabet (Q1 2025 earnings call)

Many organizations are just getting started with Gen AI

Generative AI is a top boardroom agenda

82% See gen AI as significant or transformative to their industry

81% Have urgency to adopt

23% Feel they are behind

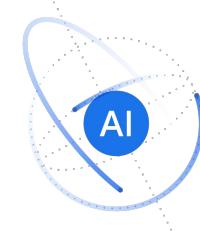
15% Self reported experts

Room for improvement

Only

10%

are very satisfied with their existing AI program

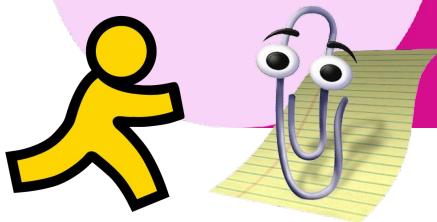


Source: Google Cloud internal Data Survey, 1000 decision makers.

AI capabilities continue to evolve

Picking the right modality for the job is key

The Deterministic Chatbot



Generative Chatbots

ReAct Agents

Autonomous Agents



Google AI's Ecosystem

01

Gemini Code Assist

Built for SDLC use cases

SaaS

Available in many
clients/surfaces

No skills required

Built-in SDLC Agents

Fully managed and supported

codeassist.google

02

Agentspace

Custom Agent Platform

Not strictly for software
development

Low Code/No Code UX

Build your own experiences

Customize with integrations

Maintain yourself

agentspace.google.com

03

Vertex AI

Build/Run any AI application

Full control end to end DIY

Models, Platforms, Architecture,
Languages

AI skill set and maintenance
required

vertexai.google.com

04

Jules

Asynchronous Coding
Agent

Long running jobs
Autonomous (runs in
Google infra)

Integrate with GitHub
Public Beta

Public GA

jules.google

Enterprise Products

Google Labs
labs.google



PRDs > Prototypes

Google's approach

Empower enterprise AI adoption by focusing on quality and software delivery performance

01

Empower Enterprise AI Adoption

Secure
Fully integrated
Enterprise operations
Available everywhere
Built for Enterprise

02

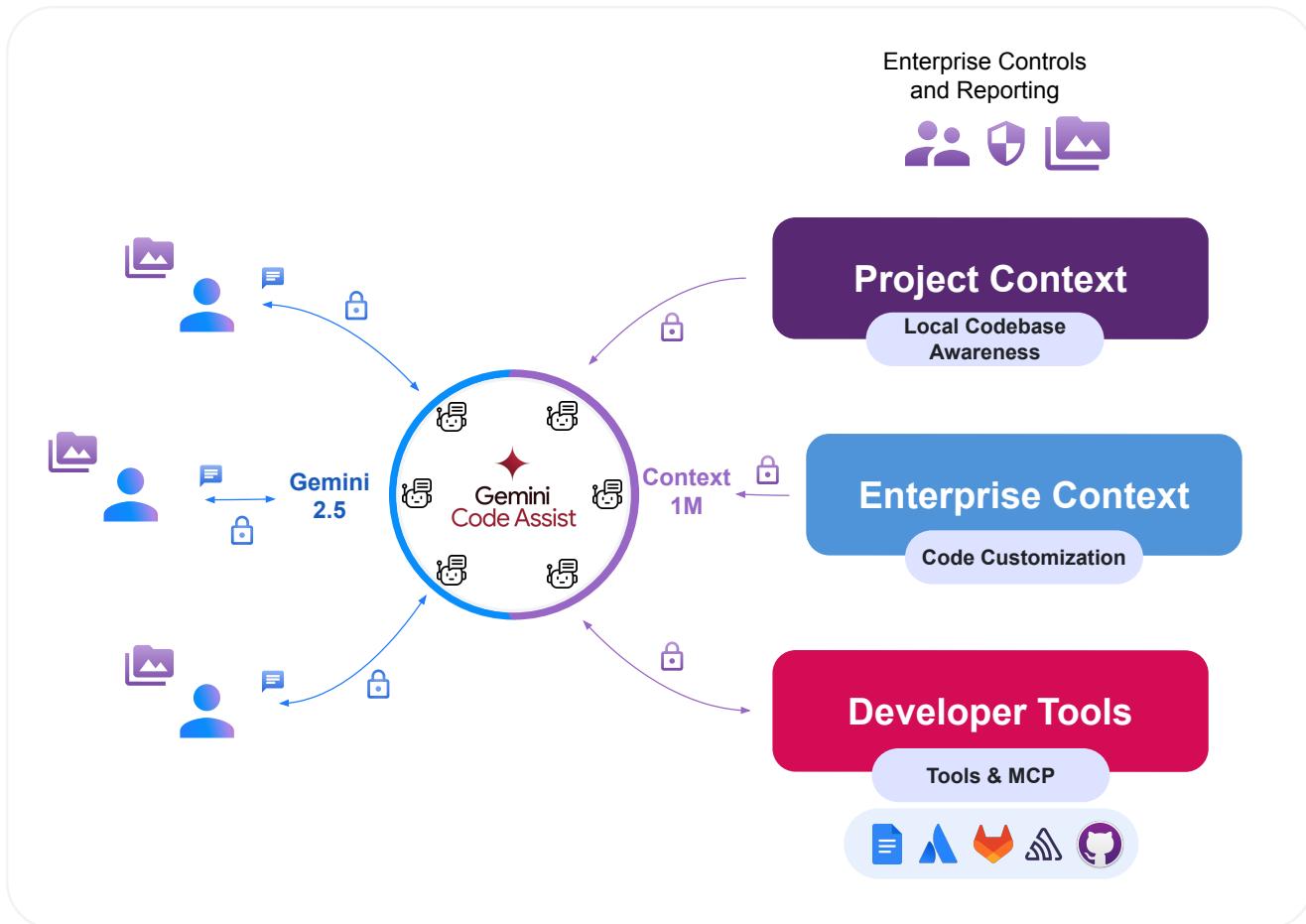
Improve Quality not just speed

Codebase Awareness
Enterprise Context
Maintainable
Consistent
Developer Experience

03

Boost Software Delivery Performance

Focus on the entire SDLC
Inner Loop
Outer Loop
Plan to Production
Agentic flows in Dev and Ops





Gemini Code Assist

Gemini 2.5 Pro

bit.ly/gemini25pro-blog

Benchmark	Gemini 2.5 Pro Experimental (03-25) High		OpenAI o3-mini	OpenAI GPT-4.5	Claude 3.7 Sonnet 64k Extended Thinking	Grok 3 Beta Extended Thinking	DeepSeek R1
Reasoning & knowledge Humanity's Last Exam (no tools)	18.8%		14.0%*	6.4%	8.9%	—	8.6%*
Science GPQA diamond	single attempt (pass@1)	84.0%	79.7%	71.4%	78.2%	80.2%	71.5%
	multiple attempts	—	—	—	84.8%	84.6%	—
Mathematics AIME 2025	single attempt (pass@1)	86.7%	86.5%	—	49.5%	77.3%	70.0%
	multiple attempts	—	—	—	—	93.3%	—
Mathematics AIME 2024	single attempt (pass@1)	92.0%	87.3%	36.7%	61.3%	83.9%	79.8%
	multiple attempts	—	—	—	80.0%	93.3%	—
Code generation LiveCodeBench v5	single attempt (pass@1)	70.4%	74.1%	—	—	70.6%	64.3%
	multiple attempts	—	—	—	—	79.4%	—
Code editing Aider Polyglot	74.0% / 68.6% whole / diff		60.4%	44.9%	64.9%	—	56.9%
	diff		diff	diff	diff	—	diff
Agentic coding SWE-bench verified	63.8%	49.3%	38.0%	70.3%	—	49.2%	—
Factuality SimpleQA	52.9%	13.8%	62.5%	—	43.6%	30.1%	—
Visual reasoning MMMU	single attempt (pass@1)	81.7%	no MM support	74.4%	75.0%	76.0%	no MM support
	multiple attempts	—	no MM support	—	—	78.0%	no MM support
Image understanding Vibe-Eval (Reka)	69.4%	no MM support	—	—	—	—	no MM support
Long context MRCR	128k (average)	94.5%	61.4%	64.0%	—	—	—
	1M (pointwise)	83.1%	—	—	—	—	—
Multilingual performance Global MMLU (Lite)	89.8%	—	—	—	—	—	—

Core features

Smart actions—code transformation

Assistive development

Developers can use natural language prompts to perform modifications on existing code, such as adding comments, bug fixing, or making code more readable

IDE Coverage



The screenshot shows a code editor interface with a dark theme. On the left is an Explorer sidebar displaying a project structure for a 'GO-GUESTBOOK' application. The 'src' folder contains 'Backend' and 'Frontend' subfolders. The 'Frontend' folder is expanded, showing 'kubernetes-manifests', 'static', 'templates', 'Dockerfile', 'go.mod', 'go.sum', 'main.go', 'Skaffold.yaml', '.dockerignore', '.gitignore', 'README.md', and 'Skaffold.yaml'. The main editor area shows a Go file named 'main.go' with several lines of code. A specific line, '148 // Create a new bucket', is highlighted with a yellow background. A tooltip window is open over this line, containing the text '/transform Add comments for selected code'. Below the tooltip are buttons for 'Accept' and 'Discard'. At the bottom right of the editor, it says '1 of 3 suggestions ↑ ↓'. The status bar at the bottom of the editor shows '20'.

```
// guestbookEntry represents the message object returned from the backend API.
type guestbookEntry struct {
    Author string `json:"author"`
    Message string `json:"message"`
    Date time.Time `json:"date"`
}

func CreateGCSBucket(projectID, bucketName string) error {
    // Create a new storage client.
    client, err := storage.NewClient(ctx)
    if err != nil {
        return err
    }

    bucket, err := client.Bucket(bucketName).Create(ctx)
    if err != nil {
        return err
    }
}

// Create a new bucket
bucket, err := client.Bucket(bucketName).Create(ctx)
if err != nil {
    return err
}

```

Explain code (Chat)

Assistive development

To improve test coverage for a code base a developer is not familiar with, they can ask Gemini Code Assist to **explain the code** to quickly and effectively understand how to start testing it.

IDE Coverage



The screenshot shows a code editor window with the file `internallogger.go` open. The file path is `src > frontend > internallogger.go`. The code defines a `Logger` interface and an `internalLogger` struct that implements it. It also includes a `NewLogger` function that opens a file named `log.txt`.

```
internallogger.go
...
src > frontend > internallogger.go > ...
1 package internallogger
2
3 import (
4     "fmt"
5     "os"
6 )
7
8 type Logger interface {
9     Info(msg string)
10    Error(msg string)
11    Close()
12 }
13
14 type internalLogger struct {
15     file *os.File
16 }
17
18 func NewLogger() (Logger, error) {
19     file, err := os.OpenFile("log.txt", os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0666)
20     if err != nil {
21         return nil, err
22     }
23     return &internalLogger{file}, nil
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
```

Test generation - Suggest a test plan (Chat) - 1/2

Assistive development

To define a test plan to ensure the code works as intended. Devs can ask Gemini Code Assist to **suggest a unit test plan** for the code, and iterate on it to ensure good coverage of edge cases.

IDE Coverage



The screenshot shows the DUET AI interface. On the left is a sidebar with icons for file operations, search, navigation, and settings. In the center is a code editor window titled "internallogger.go". The code is a Go program that defines a logger interface and a concrete implementation. A blue input field at the bottom says "Enter a prompt here".

```
internallogger.go
...
internallogger.go X
src > frontend > internallogger.go > ...
1 package internallogger
2
3 import (
4     "fmt"
5     "os"
6 )
7
8 type Logger interface {
9     Info(msg string)
10    Error(msg string)
11    Close()
12 }
13
14 type internalLogger struct {
15     file *os.File
16 }
17
18 func NewLogger() (Logger, error) {
19     file, err := os.OpenFile("log.txt", os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0666)
20     if err != nil {
21         return nil, err
22     }
23     return &internalLogger{file}, nil
24 }
25
26 func (l *internalLogger) Info(msg string) {
27     fmt.Fprintf(l.file, "[INFO] %s\n", msg)
28 }
29
30 func (l *internalLogger) Error(msg string) {
31     fmt.Fprintf(l.file, "[ERROR] %s\n", msg)
32 }
33
34 func (l *internalLogger) CymbalEvent(msg string) {
35     fmt.Fprintf(l.file, "[EVENT] %s\n", msg)
36     // call event counter service
37 }
```

Test generation - Suggest a test plan (Chat) - 2/2

Assistive development

Once the test plan looks good, Devs can ask Gemini Code Assist to implement the test plan defined

IDE Coverage



The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar: Shows the project structure under **SHOPPING**, including `.idea`, `.readmes`, `img`, and `src` (which contains `Backend`, `Frontend`, `kubernetes-manifests`, `static`, `templates`, `Dockerfile`, `go.mod`, `go.sum`, `internallogger.go`, `Skaffold.yaml`, `.dockerignore`, `.gitignore`, `README.md`, and `Skaffold.yaml`).
- internallogger.go** file tab: The current file being edited.
- Code Editor Content:**

```
internallogger.go
-----
1 package internallogger
2
3 import (
4     "fmt"
5     "os"
6 )
7
8 type Logger interface {
9     Info(msg string)
10    Error(msg string)
11    Close()
12 }
13
14 type internalLogger struct {
15     file *os.File
16 }
17
18 func NewLogger() (Logger, error) {
19     file, err := os.OpenFile("log.txt", os.O_CREATE|os.O_WRONLY|os.O_APPEND, 0666)
20     if err != nil {
21         return nil, err
22     }
23     return &internalLogger{file}, nil
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
```

Code Assist Agent Mode

Reimagined chat experience in Code Assist
powered by the Gemini CLI. It expands the
capabilities of **command-response** interactions
inside Code Assist chat by providing support for
multi file edits, full project context, built-in tools
and integration with ecosystem tools (using MCP)

Evolution of Chat



Interactive Chat

Answer questions
Cite trusted sources
Generate code based on a given prompt



Agent Mode

Solve more complex problems through **multiple steps**
Implement **proposed solutions** on behalf of the developer
Amplify impact through **integrations with developer tools**
Extensibility to support a **rich tooling ecosystem**

Code Assist Agent Mode



Code Assist Agent Mode (Preview)

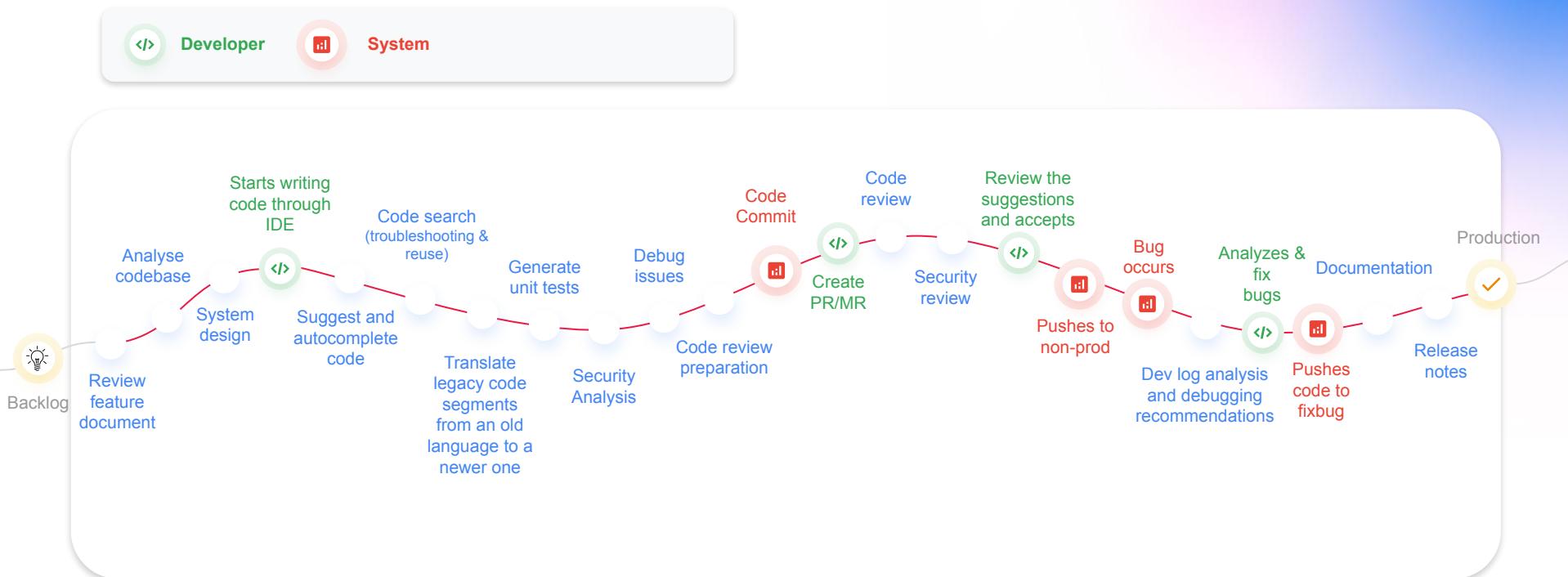
- Initial release for VSCode
- Powered by Gemini CLI (for similar experience)
- Use with Gemini Code Assist license (all tiers with existing [quotas and limits](#)).
- MCP Support (for local servers & remote servers [\[url\]](#))



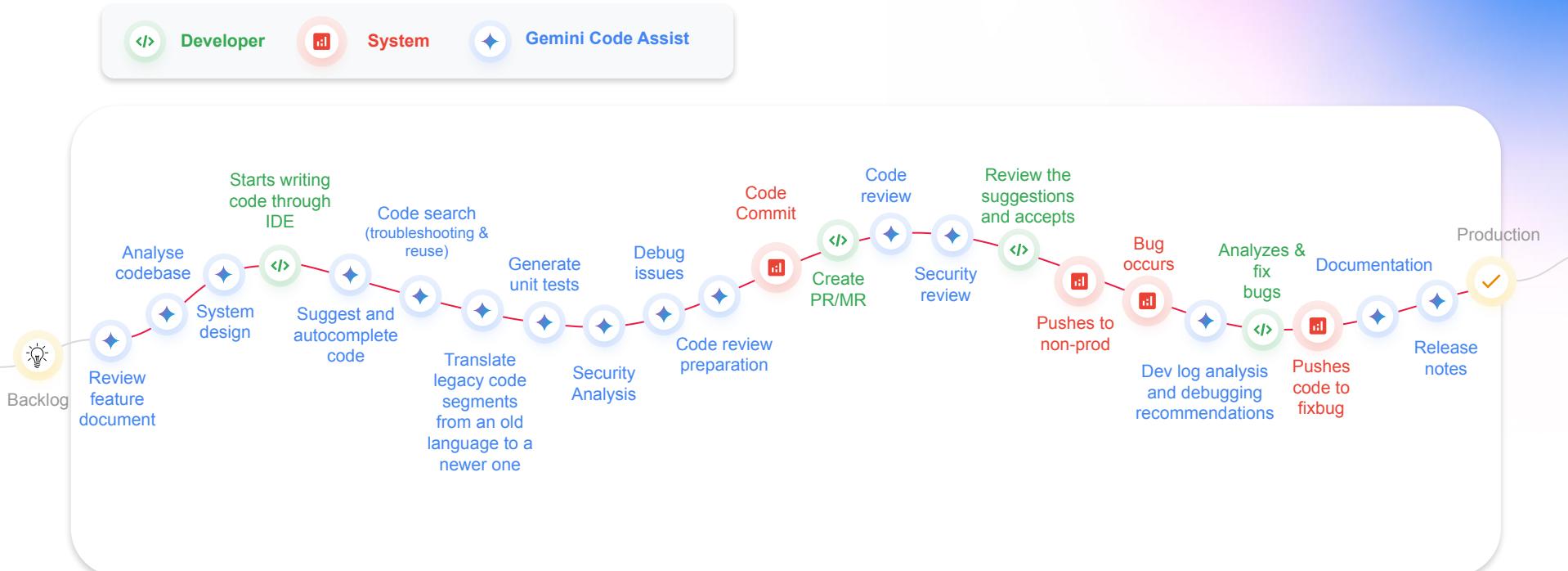
Gemini CLI (Preview)

- Open source CLI tool
- Accessible anywhere
 - Use with Gemini Code Assist license (all tiers with existing [quotas and limits](#))
 - OR Bring your own Gemini API key
- MCP Support (local servers, remote servers [\[url\]](#))

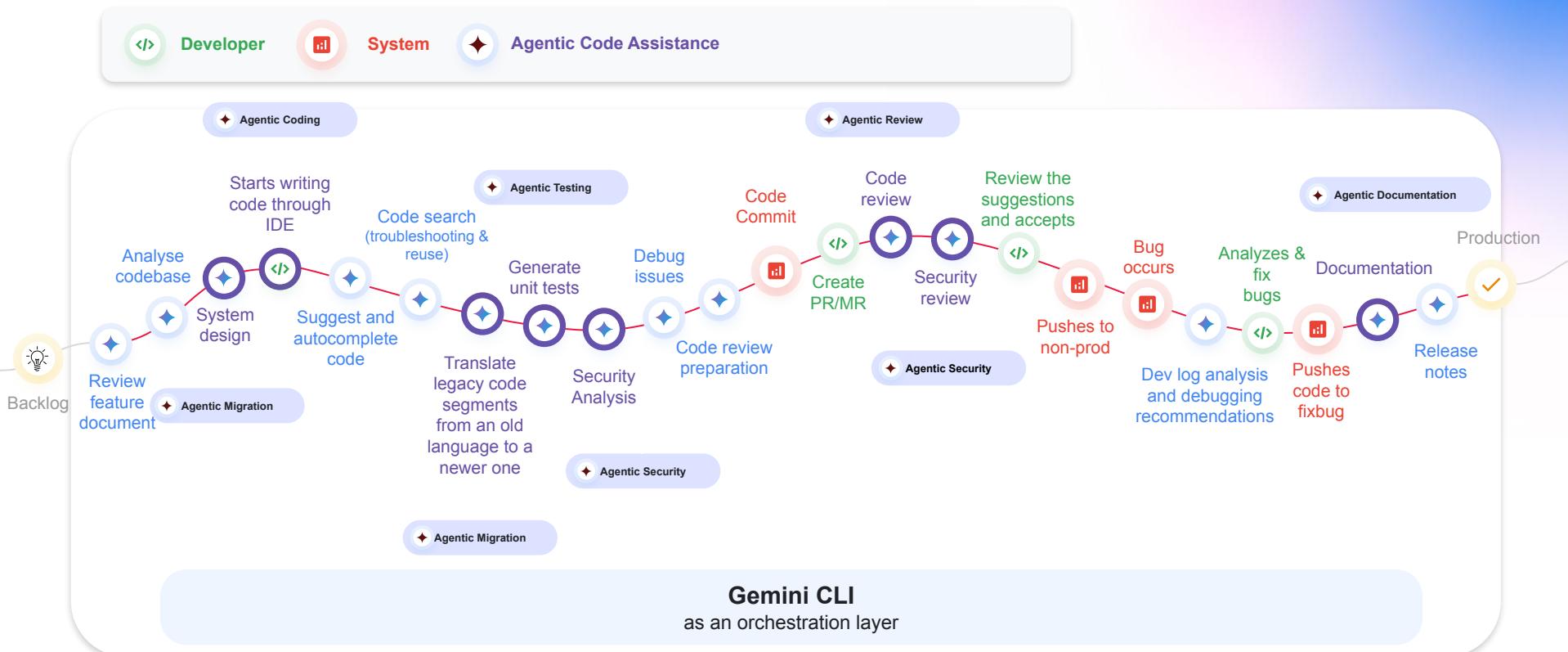
Typical Workflow



Typical Workflow with AI Assistance



Agent Mode and SDLC



From Prompting to Coaching



Techniques

The Prompting Spectrum

Basic Prompting

Asking the LLM to perform simple tasks like “**write a function to compute the sum of two numbers in Python**”.

These are not complex, have explicit instructions (“in Python”) and do not assume much from the LLM.

Context-Aware Prompting

Give the LLM some knowledge of your project by giving it specific context such as: “**modify @[app.py](#) so that it uses functions from @./my/[library.py](#) instead of writing them from scratch and use documentation from [@<https://m3.material.io/> for any CSS”](https://m3.material.io/)**

Meta Prompting

Instead of writing the prompts by hand, coach the agent on how to write the prompts for the tasks it needs to execute. “**You need to write a basic web app to serve a guestbook with a scalable database.**” Context is inferred and a plan is created.

Prompting Demo!

[Google Guide](#)



Working with Tools + Context

Supercharge your Agentic Peer Programmer



Tools

Agents can use predefined tools (`readFile`, `writeFile`, `bash`, etc) and user-defined tools (custom code).



MCP

Connect agents to other systems to get information and/or take action.



Local Context

Repo awareness plus specific references to local files and folders.

Controlling your Agent

Gemini Code Assist Agentic Architecture

Understanding how Agents Behave

[RTFM!](#)

GEMINI.md

The primary orchestrator for all agentic behavior. Better understood as the agent's "memory". This is where everything stems from.

+ settings.json

DESIGN.md

- Which design libraries (public and private) to use
- Design philosophy
- Guidelines, templates, images

DATABASES.md

- Preferences and design approach
- Existing data landscape, schemas, data dictionaries, etc.
- Platforms and choices

DEPLOYMENT.md

- Deployment pipeline guidance
- Security requirements
- Infrastructure / Platform information

Agent Setup Workshop!

[Google Guide](#)



Planning + Executing

[PLAN.md](#)

Edit the plan

New and existing code

Planning + Executing Workshop!



Thank you!