SANGFOR

# Sangfor CoStrict:Enterprise-level AI-Native development  toolkit

**Sam Hu | AI Solution Manager**

sam.hu@sangfor.com

Sangfor Technologies

# Contents

# PART 1

# R&D's Opportunity and Challenge in AI era

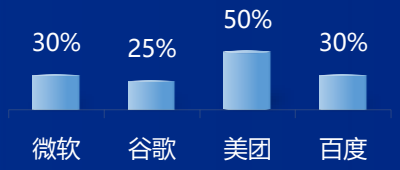
# Trends: Large-scale Production and Broad Prospects

**After Transformer and ChatGPT went viral, LLM starts revolutionizing various fields, particularly in coding.**

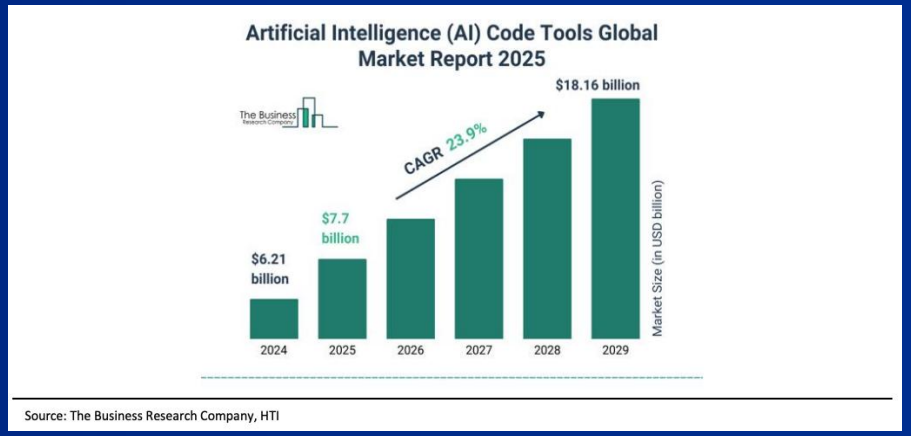
- 2021.6, Microsoft published Copilot
- 2023.3, Anysphere published Cursor
- 2025.5, Anthropic published Claude Code
- Various tools emerges: Kiro, Tabnine, Trae、Qoder

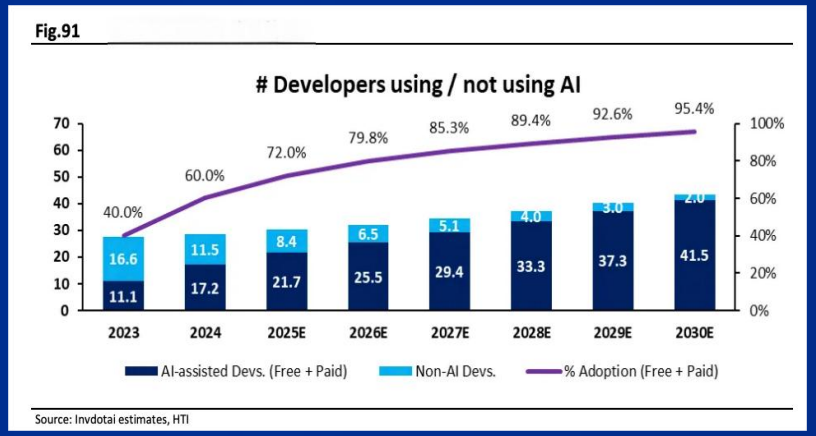
**Individual developers are using AI**
**Single person=Entire enterprise**

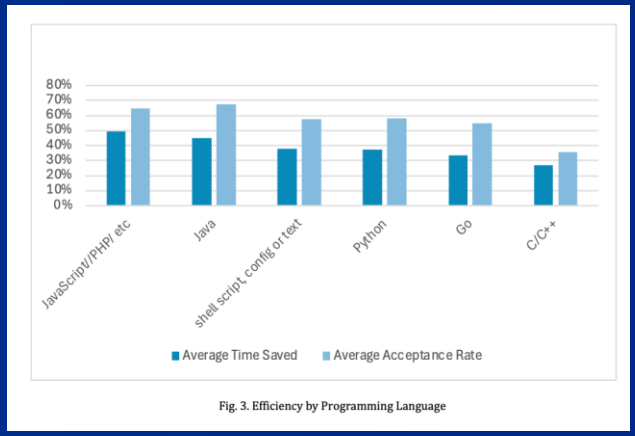
**More enterprise starts AI Coding**
**Code generated by AI increases**




## AI coding has broad application prospects and is widely used in production environments



# AI brings hope to traditional coding

## Traditional Coding Problems

Product complicated, Frequent staff turnover

### R&D team face tremendous challenges

Hard to balance quality and progress
No time for competitiveness and innovation

Slow in learning different technology stacks

Team lack of accumulation

## AI Coding Advantage

AI was trained with comprehensive high-quality code

### Equivalent to a full-stack engineer

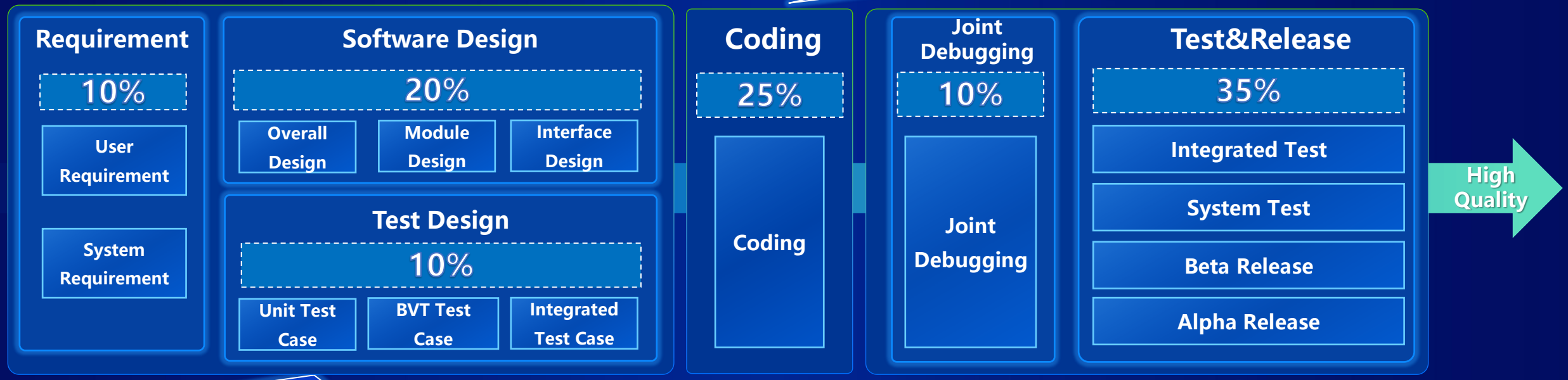
- Extensive experience in technology, architecture
- Supports all programming languages and can quickly generate code snippets


# Current AI Coding tools are not suitable for enterprise

## Existing AI coding tools tend to favor "Vibe Coding"

- Focus on the developer experience, quickly deliver from requirements to coding
- Lack of process constraints and is not suitable for high-quality enterprise development scenarios

① Although accelerated the code generation, but in standardized process, coding is only a proportion of 25%, even the coding efficiency doubled, it only increases 10% overall




② Under most conditions, new development is based on existing project, AI lack of the understanding of the original business, it can’ t generate complete and usable code, engineers have to adjust over and over

③ Coding without rigorous requirements and design, there are many hidden defects, results in longer test time, the efficiency is not increased

# Enterprise requires high product quality and data privacy

For enterprises with high quality standards, quality issues not only incur costs during product development but also lead to after-sales expenses and brand damage.

| Stage | Time of fixing 1 bug | Cost Structure |
|---|---|---|
| Coding | < 60min | Fix the code directly and complete self-testing |
| Test | 300min<br>5 times of coding stage | a) Testing discovery (1 person 18 per day = one 26min)<br>b) Test regression(1 person 8 per day = one 60min)<br>c) Developer fixing (1 person 2 per day = one 240min) |
| Operation | 15 times of coding stage | a) Direct cost（TAC/FAE+Reproduction+Fixing+Testing+Upgrading）<br>b) Indirect loss（Customer / Public praise） |

# Enterprise requires high product quality and data privacy

**Data is sensitive asset, requires privatization plan**

**Mainstream tools are mainly for online individual developers**

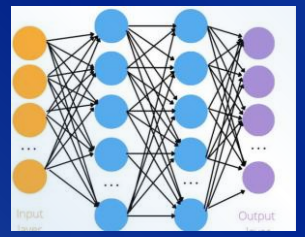

Claude Code
Copilot
CodeGeeX
Cline
通义灵码
Cursor
Trae
Gemini Cli

**If code was transfered via public network, it would probably be used in LLM training or other purpose**

# AI Coding Trends

"Rule+Template" based Code Completion

"Neural Network" based Code Generation



"Multiagent Collaboration" based Software Development

"Word Count" based Code Prediction


"LLM" based Code Generation


"AI Native" based Deep Integration

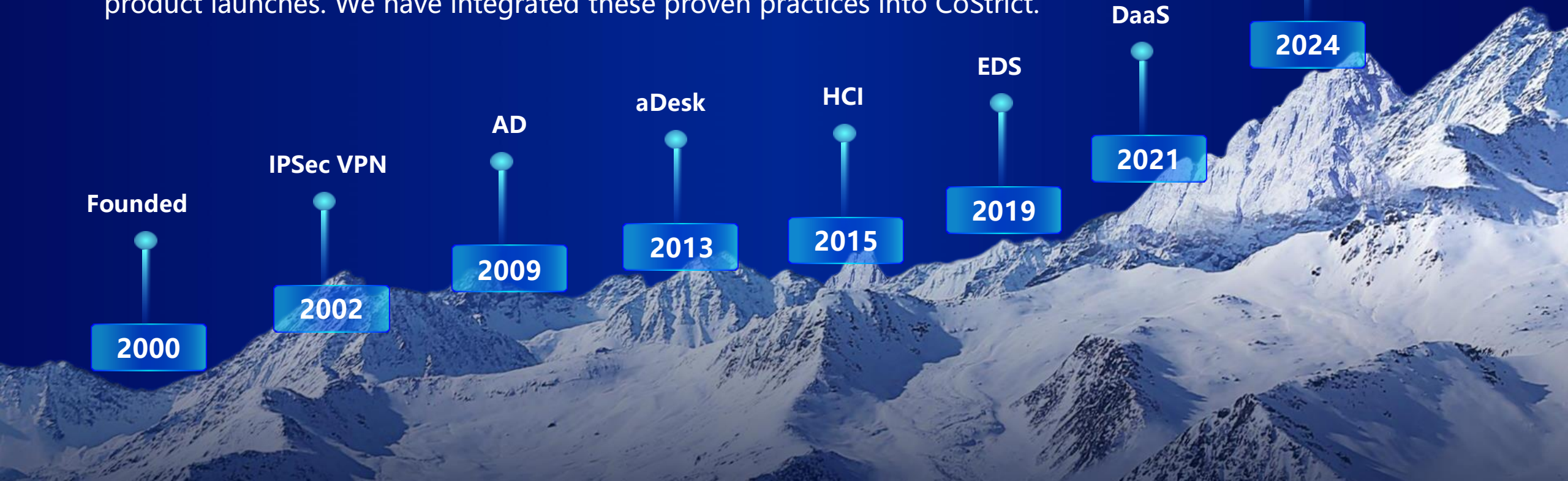
# PART 2

# CoStrict Core Value

# Sangfor committed to delivering high-quality

With 26 years of enterprise product development expertise, Sangfor has developed multiple data center level products including AD, HCI, EDS, and AICP.

In 2021, Sangfor introduced the IPD process to enhance the product quality, resulting in improved product quality, more efficient R&D, reduced manpower, and continuous new product launches. We have integrated these proven practices into CoStrict.



# CoStrict Positioning

CoStrict is Sangfor's open-source AI coding solution, developed for enterprise-grade scenarios requiring **“Private Deployment + High-Quality Standards”** .

It incorporates Sangfor's proven expertise in high-quality coding processes.


## Deeply Understand The Project

Understand the project, and provide evidences for code generation

## Multiple Coding Mode

Various AI staff and coding mode for different scenarios, such as code snippet, 0-1 and 1-N project

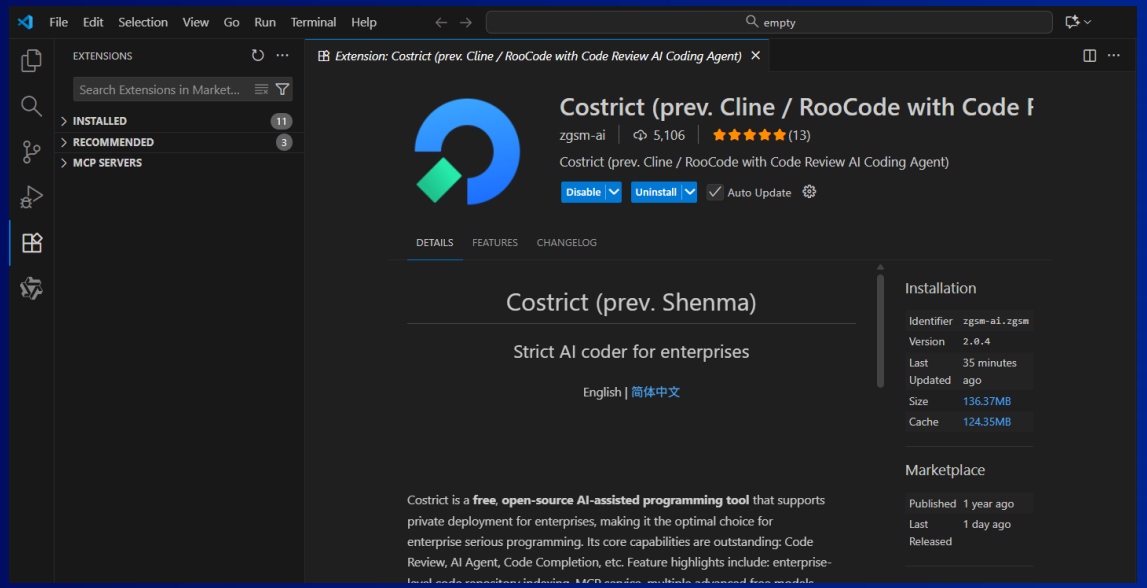
## Private Deployment

Data is not transmitted to the external network, ensuring the security of code and document assets.

# CoStrict Product form

## IDE plugin, currently supporting

- VS Code：1.86.3 and above
- JetBrains series IDEs：2023.1 and above




### Jetbrains IDEs

RunVSAgent currently supports the following JetBrains IDE series:

- **IntelliJ IDEA** (Ultimate & Community)
- **WebStorm** - JavaScript and TypeScript development
- **PyCharm** (Professional & Community) - Python development
- **PhpStorm** - PHP development
- **RubyMine** - Ruby development
- **CLion** - C/C++ development
- **GoLand** - Go development
- **DataGrip** - Database development
- **Rider** - .NET development
- **Android Studio** - Android development

> **Note:** Requires JetBrains IDE version 2023.1 or later for optimal compatibility.

Python Go Java JavaScript TypeScript C/C++ ...

PART 2

# CoStrict Core Value 1

## Deeply Understand The Project

# Core Value 1: Deeply Understand The Project

**Traditional Develpment Problems**

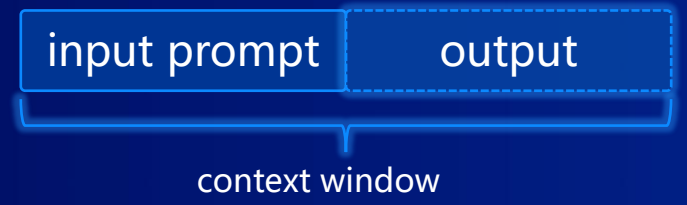
- Code is too complicated under years of development, hard to learn, hard to know the whole project
- Frequent staff turnover and a shortage of business experts can lead to single points of failure

**AI Coding Still Not Doing Well**

- AI is too general to understand vertical business domains. The generated code does not meet business requirements.
- LLM is not able to understand large scale project due to limited context, hard to generate complete code
- AI may be reinventing the wheel, not using existing codebases, or creating illusions by imagining variables and interface calls. The generated code is unusable.

# Core Value 1: Deeply Understand The Project

## LLM has limited context window, long context may cause memory loss, resulting in lower generation accuracy.




## Mainstream model context window

| Model | Max context window(Token) |
|---|---|
| DeepSeek-v3.1 | 128K tokens |
| GLM-4.7 | 200K tokens |
| Kimi K2 | 256K tokens |
| Qwen3-Next | 256K tokens |
| GPT-5.2 | 256K tokens |
| Claude 4.5 Sonnet | 200K/1000K tokens |
| Gemini 3 Pro | 1000K tokens |

**Fiction.LiveBench for Long Context Deep Comprehension**

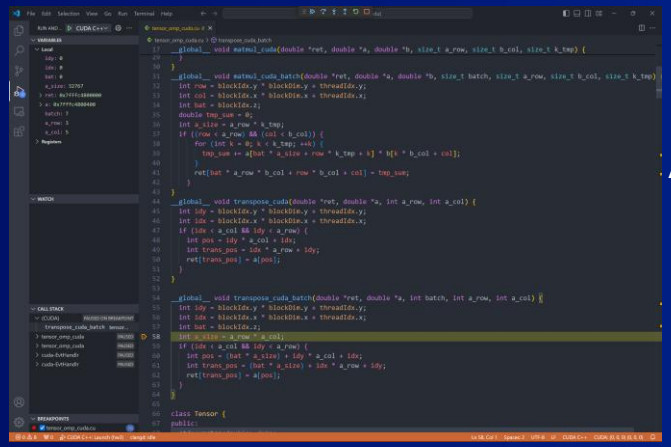
https://fiction.live/stories/Fiction-liveBench-August-21-2025/oQdzQvKHw8JyXbN87

| Model | 0 | 400 | 1k | 2k | 4k | 8k | 16k | 32k | 60k | 120k | 192k |
|---|---|---|---|---|---|---|---|---|---|---|---|
| gpt-oss-120b-high [chutes] | 100.0 | 75.0 | 61.1 | 55.6 | 52.8 | 47.2 | 44.4 | 44.4 | 38.9 | - | - |
| gpt-5 | 100.0 | 97.2 | 94.4 | 97.2 | 97.2 | 100.0 | 97.2 | 97.2 | 100.0 | 96.9 | 87.5 |
| gpt-5-mini | 100.0 | 97.2 | 88.9 | 80.6 | 77.8 | 66.7 | 69.4 | 63.9 | 61.1 | 62.5 | 59.4 |
| gpt-5-nano | 75.0 | 58.3 | 63.9 | 50.0 | 50.0 | 52.8 | 44.4 | 47.2 | 36.1 | 21.9 | 34.4 |
| o3 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 88.9 | 100.0 | 83.3 | 100.0 | 58.1 |
| o4-mini | 100.0 | 100.0 | 100.0 | 100.0 | 77.8 | 66.7 | 77.8 | 55.6 | 66.7 | 62.5 | 43.8 |
| claude-opus-4:thinking | 100.0 | 94.4 | 97.2 | 94.4 | 94.4 | 83.3 | 69.4 | 80.6 | 69.4 | 62.5 | - |
| claude-sonnet-4:thinking | 100.0 | 100.0 | 100.0 | 97.2 | 100.0 | 97.2 | 100.0 | 91.7 | 94.4 | 81.3 | - |
| deepseek-r1 | 100.0 | 82.2 | 80.6 | 76.7 | 77.8 | 83.3 | 69.4 | 63.9 | 66.7 | 33.3 | |
| deepseek-r1-0528:free | 100.0 | 91.7 | 83.3 | 82.9 | 88.9 | 86.1 | 75.0 | 69.4 | 58.3 | - | - |
| **deepseek-chat-v3.1 [reasoning: high] [deepseek]** | 100.0 | 100.0 | 94.4 | 86.1 | 91.7 | 80.6 | 72.2 | 63.9 | 58.3 | 62.5 | |
| gemini-2.5-flash-lite-preview-06-17:thinking | 100.0 | 66.7 | 52.8 | 52.8 | 55.6 | 47.2 | 47.2 | 44.4 | 27.8 | 21.9 | 37.5 |
| gemini-2.5-flash | 100.0 | 94.4 | 88.9 | 86.1 | 77.8 | 69.4 | 80.6 | 69.4 | 63.9 | 65.6 | 78.1 |
| gemini-2.5-pro-preview-06-05 | 100.0 | 100.0 | 100.0 | 97.2 | 94.4 | 80.6 | 91.7 | 91.7 | 83.3 | 87.5 | 90.6 |
| gemini-2.5-pro-exp-03-25:free | 100.0 | 100.0 | 100.0 | 100.0 | 97.2 | 91.7 | 66.7 | 86.1 | 83.3 | 90.6 | |
| minimax-m1 | 87.5 | 86.1 | 94.4 | 91.7 | 88.9 | 69.4 | 69.4 | 72.2 | 58.3 | 59.4 | 71.9 |
| qwen3-235b-a22b-thinking-2507 [chutes] | 100.0 | 100.0 | 91.7 | 86.1 | 91.7 | 97.2 | 75.0 | 80.6 | 77.8 | 68.8 | |
| qwq-32b:free | 100.0 | 91.7 | 94.4 | 88.9 | 94.4 | 86.1 | 83.3 | 80.6 | 61.1 | - | |
| qwen3-235b-a22b:free | 100.0 | 90.0 | 89.3 | 80.0 | 69.0 | 66.7 | 67.7 | - | - | - | |
| qwen3-32b:free | 80.0 | 90.9 | 93.8 | 76.7 | 86.7 | 80.0 | 74.2 | - | - | - | |
| qwen3-30b-a3b:free | 85.7 | 58.1 | 54.8 | 51.5 | 53.3 | 50.0 | 40.6 | - | - | - | |
| qwen3-14b:free | 83.3 | 64.5 | 61.8 | 59.4 | 64.7 | 51.6 | 62.5 | - | - | - | |

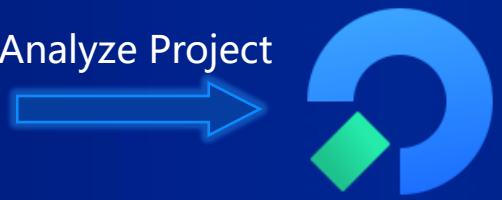
# Core Value 1: Deeply Understand The Project

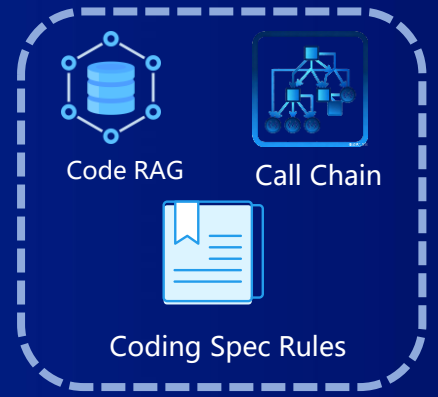
## Coding based on existing project: Analyze the project

- CoStrict understands the requirements, functions, designs, APIs, Rules
- In coding stage, CoStrict would refer relevant information to get more accurate context

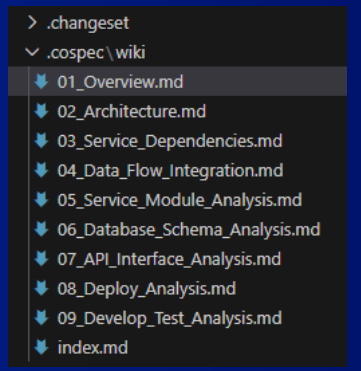


Analyze Project





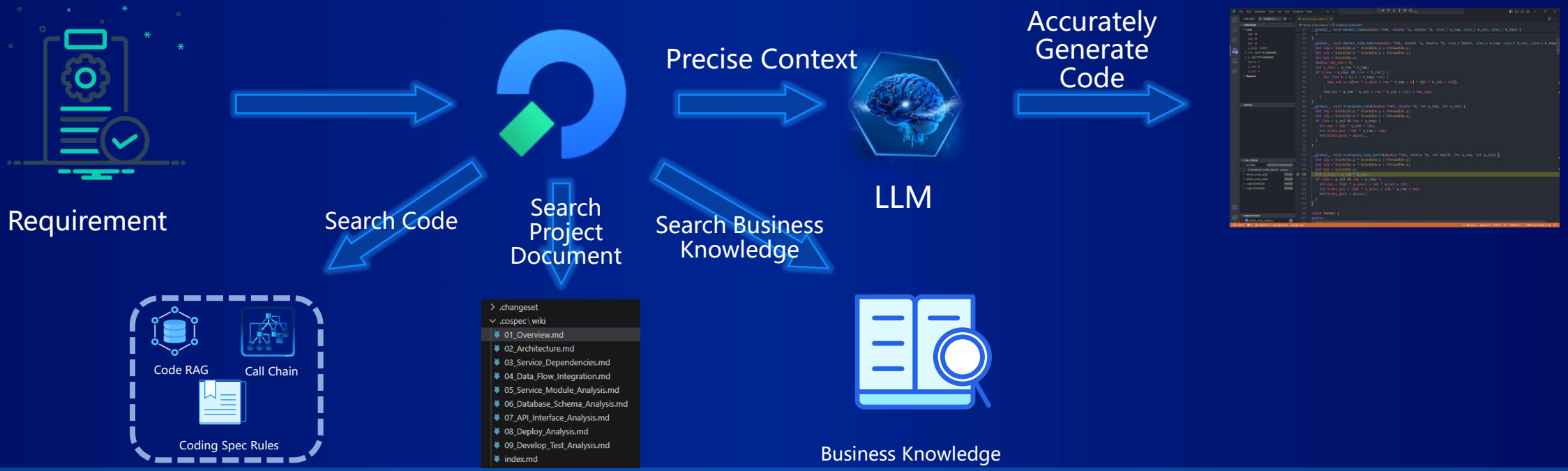

Project Document



# Core Value 1: Deeply Understand The Project

## Coding based on existing project: Implementing Requirement

- During requirement and design stage, refer to the project document
- During the coding stage, implement precise API indexing and reuse historical assets to avoid reinventing the wheel.
- Retrieve business knowledge base in real time to enhance business understanding (planning)
- Provide precise context to the LLM, reduce token consumption by 50%, and double inference performance
- Supports 5W file context, ideal for legacy projects with complex business processes, large codebase, multiple dependencies



# Core Value 1: Deeply Understand The Project

## Demo Video: CoStrict analyze existing project

PART 2

# CoStrict Core Value 2

## Multiple Coding Mode

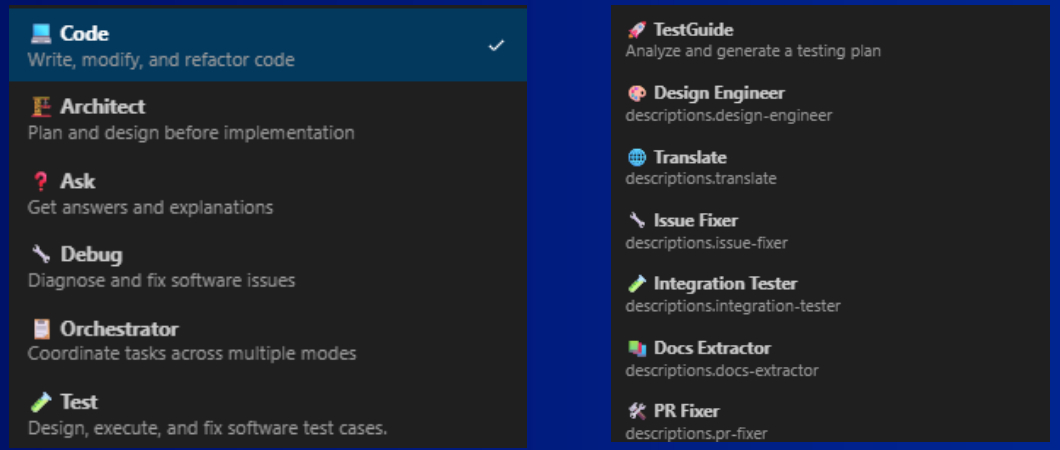
# Core Value 2: Multiple Coding Mode

LLM is popular, but hard to master, it requires experience and skills.

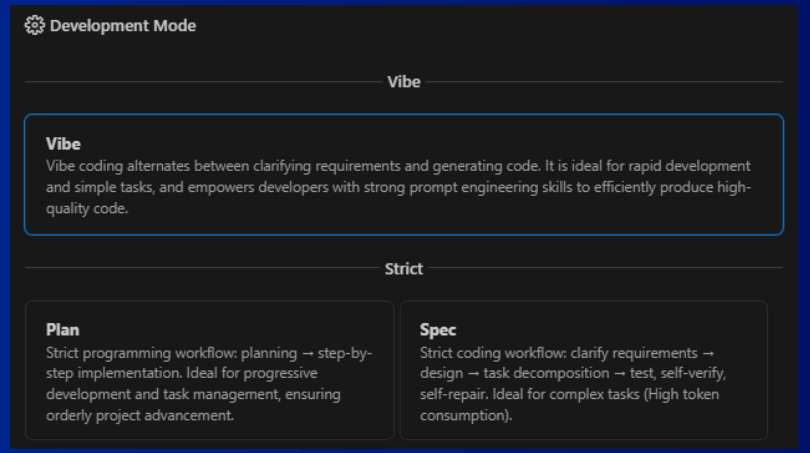
Not all engineers have rich experience and are skillful

So CoStrict designed various **AI Staff** embedded in different **Mode** for typical developing scenarios, make it easier to use

## AI Staffs




## Different Modes



## AI Staff

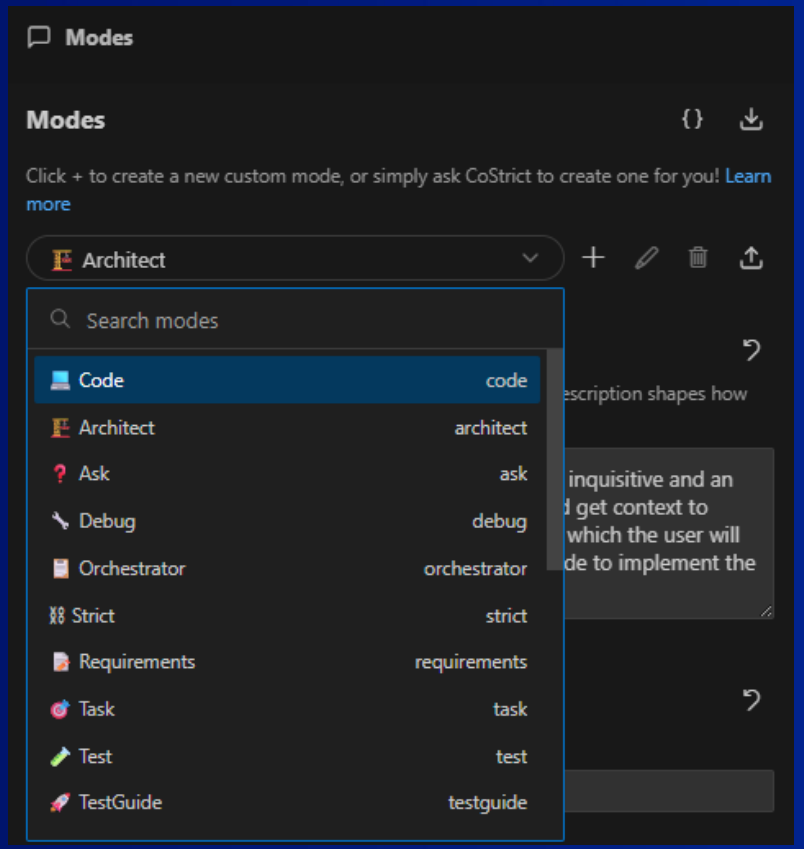
For enterprises, it requires significant time and effort to cultivate employees. Frequent turnover of staff，particularly experienced personnel，can adversely affect production.

Therefore, leveraging AI to systematically consolidate employees' best practices will be a sound strategy to consistently elevate their core competencies.

# Core Value 2: Multiple Coding Mode

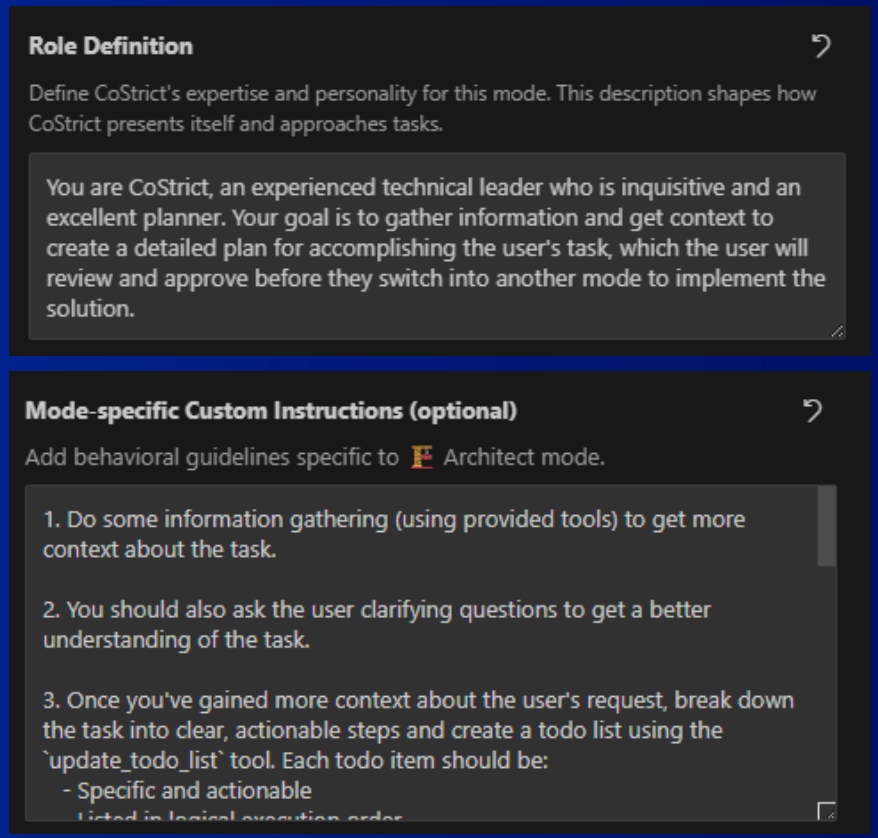
## AI Staff

Various AI Staffs Playing Different Roles




Customizable

Professional Prompt



# Core Value 2: Multiple Coding Mode

## Coding Modes

In daily development, there are three typical scenarios, and CoStrict has designed three patterns to address varying levels of complexity.

### Vibe

**Realize your idea immediately**

Suitable for code snippet generation or small feature implementation


### Plan

**Implement the requirement by specified plan**

Suitable for increment development based on existing project

### Spec

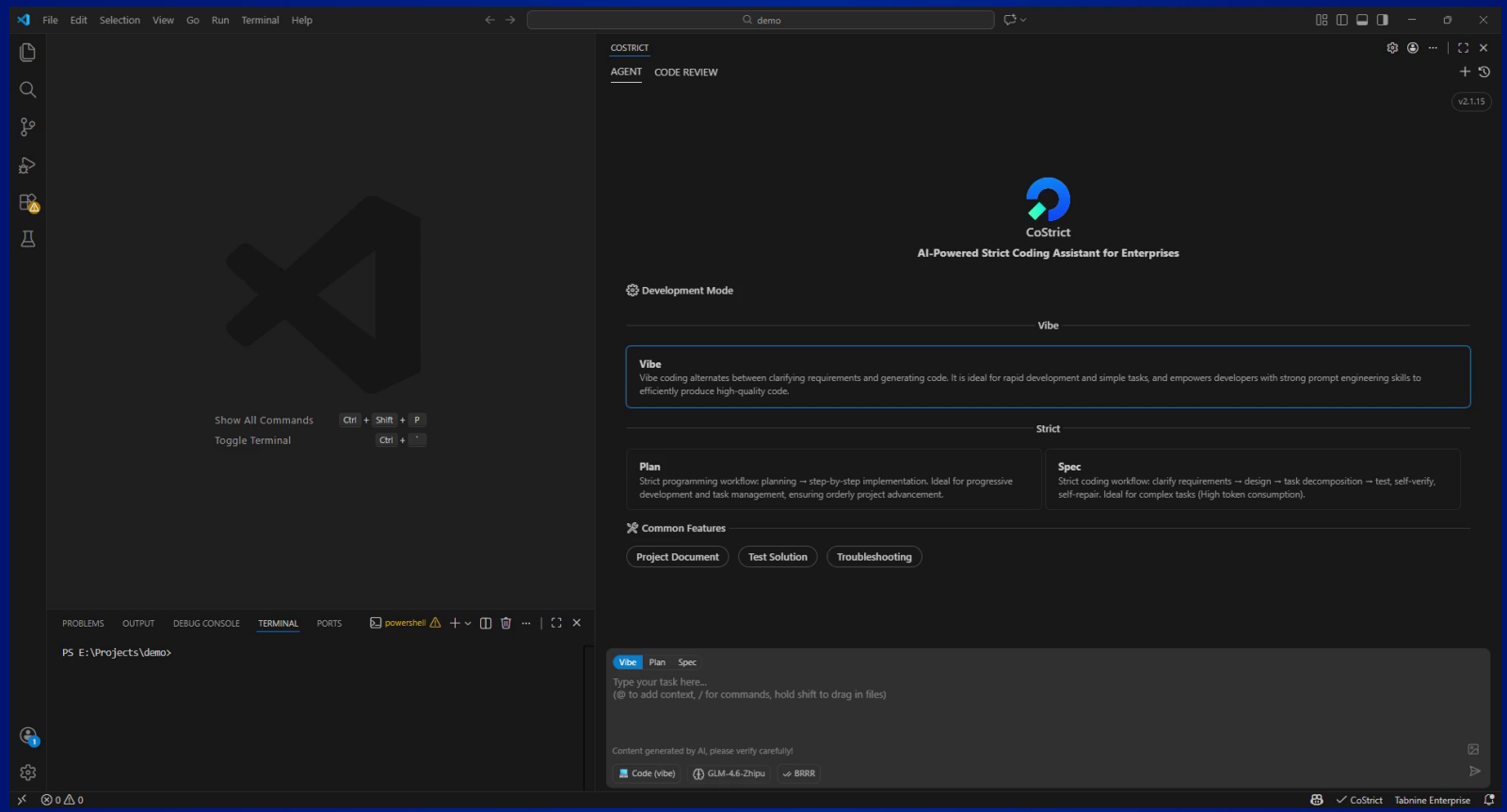
**Implement the requirement according to standard process**

Suitable for new project

# Core Value 2: Multiple Coding Mode

## Vibe Mode

Like most AI Coding assistant, CoStrict supports vibe mode, engineers can chat with CoStrict, and get feedback immediately.Such as generate code snippet, or independent modules



# Core Value 2: Multiple Coding Mode

## Plan Mode

1. Although engineers can get feedback from Vibe Coding immediately, but the code generated directly from the rough description may not be accurate, making the engineers modify the description and generate code again and again.
2. The context window of LLM is limited, and with the context being accumulated during developing, the LLM may forget tasks, generates incomplete code.

CoStrict designed Plan Mode, the engineers clarify all requirements and demands with CoStrict before it starts coding, so that less efforts would be wasted

Insert Plan mode demo video

# Core Value 2: Multiple Coding Mode

## Spec Mode

AI doesn’ t have process，it’ s hard to control the output，so CoStrict integrated enterprise standardised process to control the quality

### SDD

**Specification Driven Development**

CoStrict would follow a standard process and specificcation to generate high-quality code, the process including requirement, design, task breakdown

### TDD

**Test Driven Development**

Define the accept condition in advance, make AI generate code under A/C, when not satisfied, auto fix issues

### CodeReview

**Multi-Expert Review**

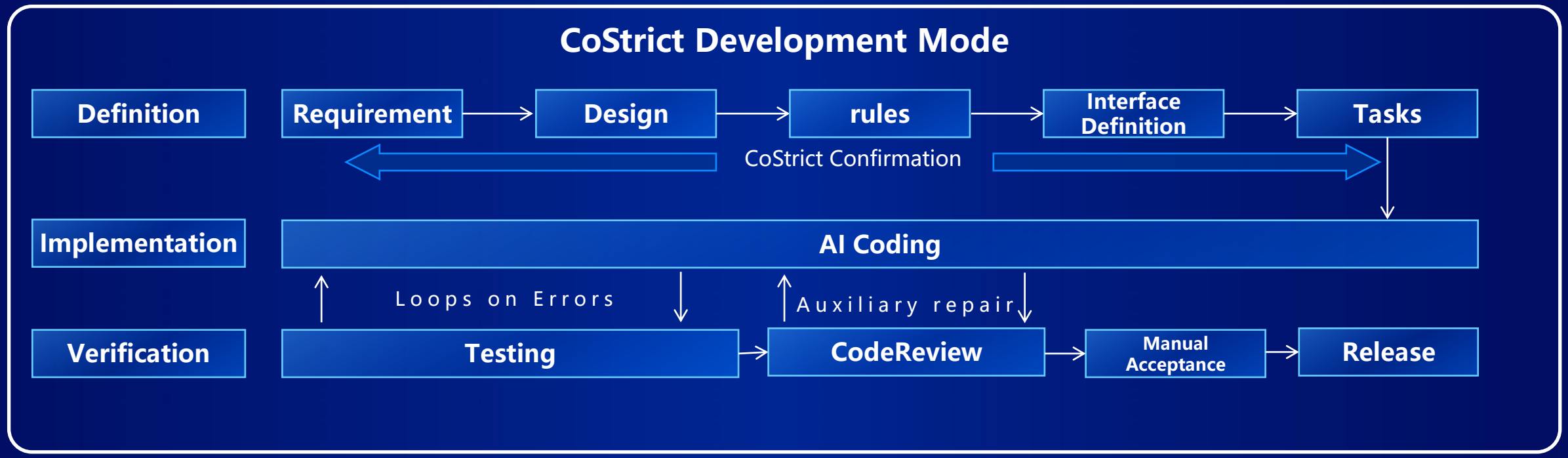
Identify logical errors, security coding errors, performance issues, and memory leaks across the project to further improve code quality

# Core Value 2: Multiple Coding Mode

## New Paradigm of Man-machine Collaboration

To accelerate engineer transition and ensure product quality, Sangfor has integrated its proven enterprise product development process into CoStrict.

- By controlling quality across end-to-end processes and standardizing the conduct of engineers and AI, we reduce costs associated with delayed defect resolution.
- Shift focus from 'implementation' to 'definition' to enable AI to automatically complete encoding and verify definition compliance



# Core Value 2: Multiple Coding Mode

## SDD Demo Video

**Traditional Development**

- Differences in individual capabilities lead to unpredictable code quality and lack of a quality baseline
- The technical solution relies on expert review, but experts are frequently interrupted, which hinders project progress.
- Not everyone follows standardized development procedures, resulting in uncontrolled progress and quality.

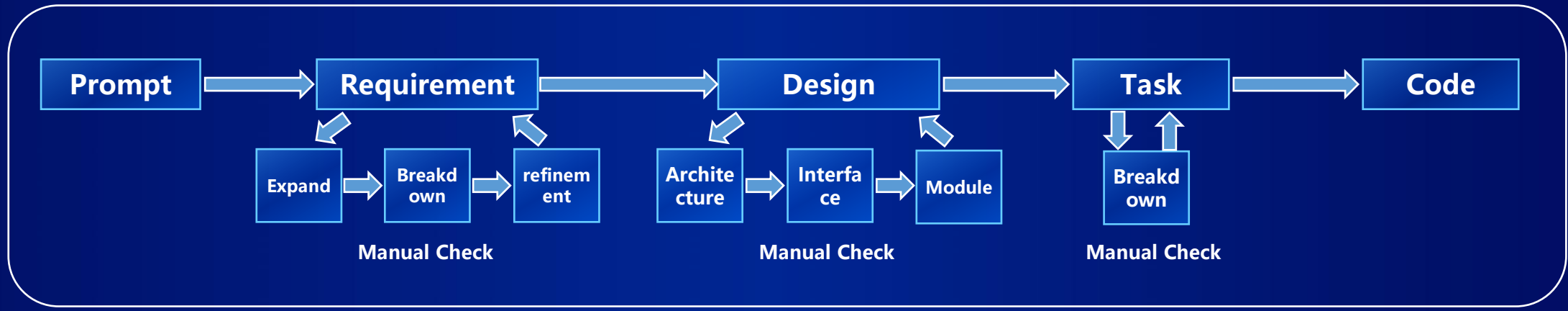
**AI Coding Not doing well**

- The Coding tool relies on prompts, and most users can only reach a 30-point level, with limited efficiency improvement.
- Lacking procedural controls, AI often moves directly from requirements to code generation, resulting in significant deviations from the final implementation.

# Core Value 2: Multiple Coding Mode

## SDD-Specification Driven Development

- **Requirement stage**: Expand, breakdown and refine the one-sentence spec
- **Design stage**: Generate architecture, interface, module design according to the requirement
- **Task stage**: Generate tasks to be done according to requirement and design
- Generate high-quality code by following standardized workflows, reduce reliance on personnel



# Core Value 2: Multiple Coding Mode

## SDD Demo Video

# Core Value 2: Multiple Coding Mode

**Traditional Test**

- The repetitive work of extensive testing and verification
- Report issues to the development team. If developers are unable to troubleshoot promptly, the optimal troubleshooting window may be missed.
- The issue fix lacks correlation analysis, changes may cause other problems, and requires repeated modifications and testing

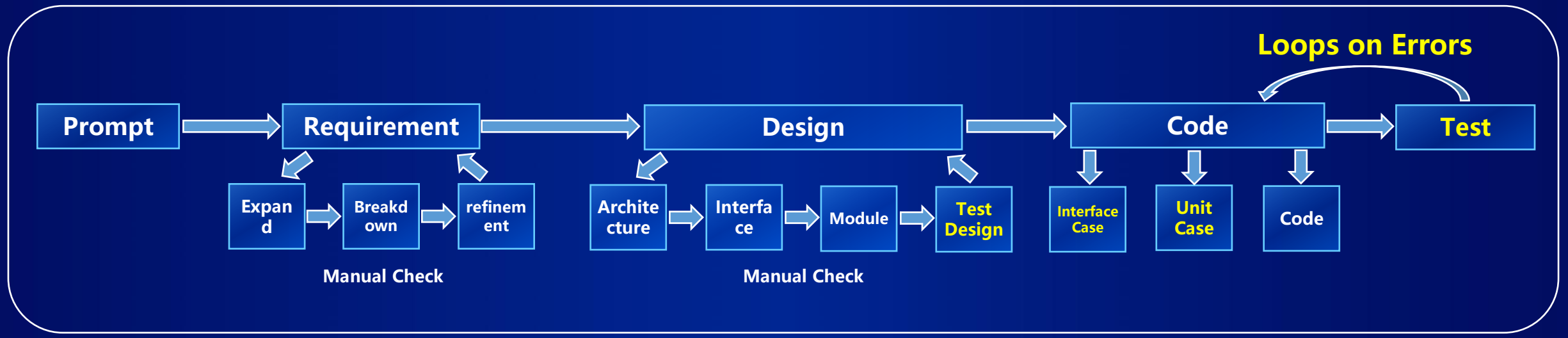
**AI Coding Not doing well**

- The AI Coding tool can automatically generate unit tests and interface tests, but lacks full-process test control
- The issue still requires manual intervention for troubleshooting, resulting in low efficiency.

# Core Value 2: Multiple Coding Mode

## TDD-Test Driven Development

- From “Human do everything” to “Human do definition, AI do coding and self-verification”
- Test become the “constraint condition”, control the implementation direction
- Test is embedded in full process
- Loops on Errors, when found errors, auto-fix and auto-test again on loop



# Core Value 2: Multiple Coding Mode

## TDD Demo Video

# Core Value 2: Multiple Coding Mode

## SDD Demo Video

# Core Value 2: Multiple Coding Mode

## CodeReview

**Traditional CodeReview**

- Relies on tech experts, but each expert is only familiar with certain field, can’ t relveal systematic risks
- When review code, the experts may not be in good condition, leading to uncontrollable code quality
- Lack of SDL experts may leave the code vulnerable, exposing it to hacking risks

**AI Coding Not doing well**

- AI is OK with code snippet review, but not good at complex logical problems due to limited context window
- AI is good at normal issue review rather than security vulnerabilities

# Core Value 2: Multiple Coding Mode

## CodeReview

CoStrict has assembled CodeReview experts across diverse problem domains to conduct multi-angle, comprehensive code diagnostics.

- Supports logical relationship review, associating the entire project context rather than code snippets. Among customer issues, 18% are logical errors.
- Supports security vulnerability review. Sangfor has deep expertise in the security field.
- Supports general issue reviews, such as performance issues and memory leaks, which account for 30% of all issues. It can intercept over 80% of these issues.

# Core Value 2: Multiple Coding Mode

## CodeReview Demo Video

PART 2

# CoStrict Core Value 3

## Private Deployment

SANGFOR

# THANK YOU!

**Harris Zhang | Cloud Solution Manager**

harris.zhang@sangfor.com

Sangfor Technologies