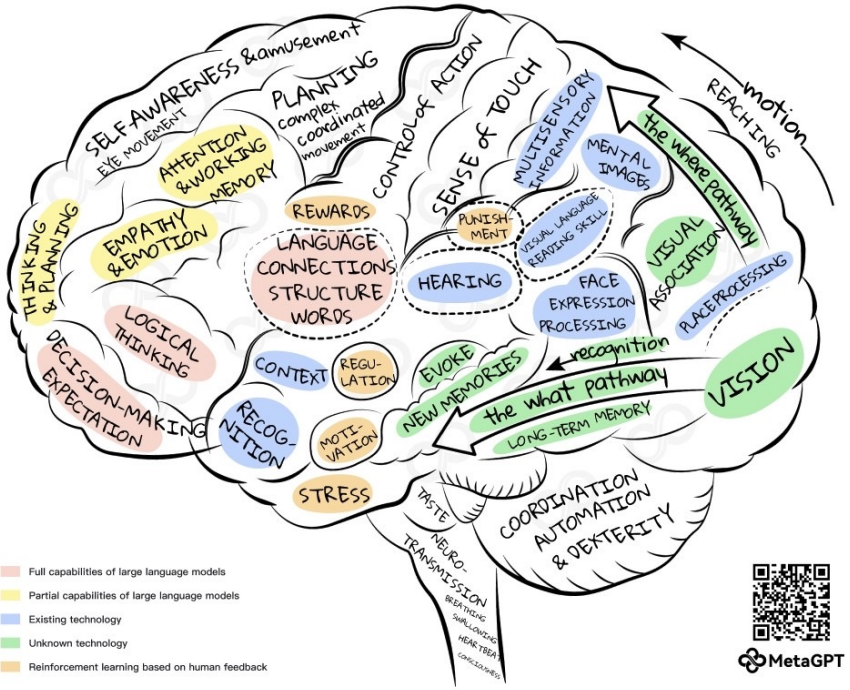
MetaGPT: Meta Programming For A Multi-Agent Collaborative Framework

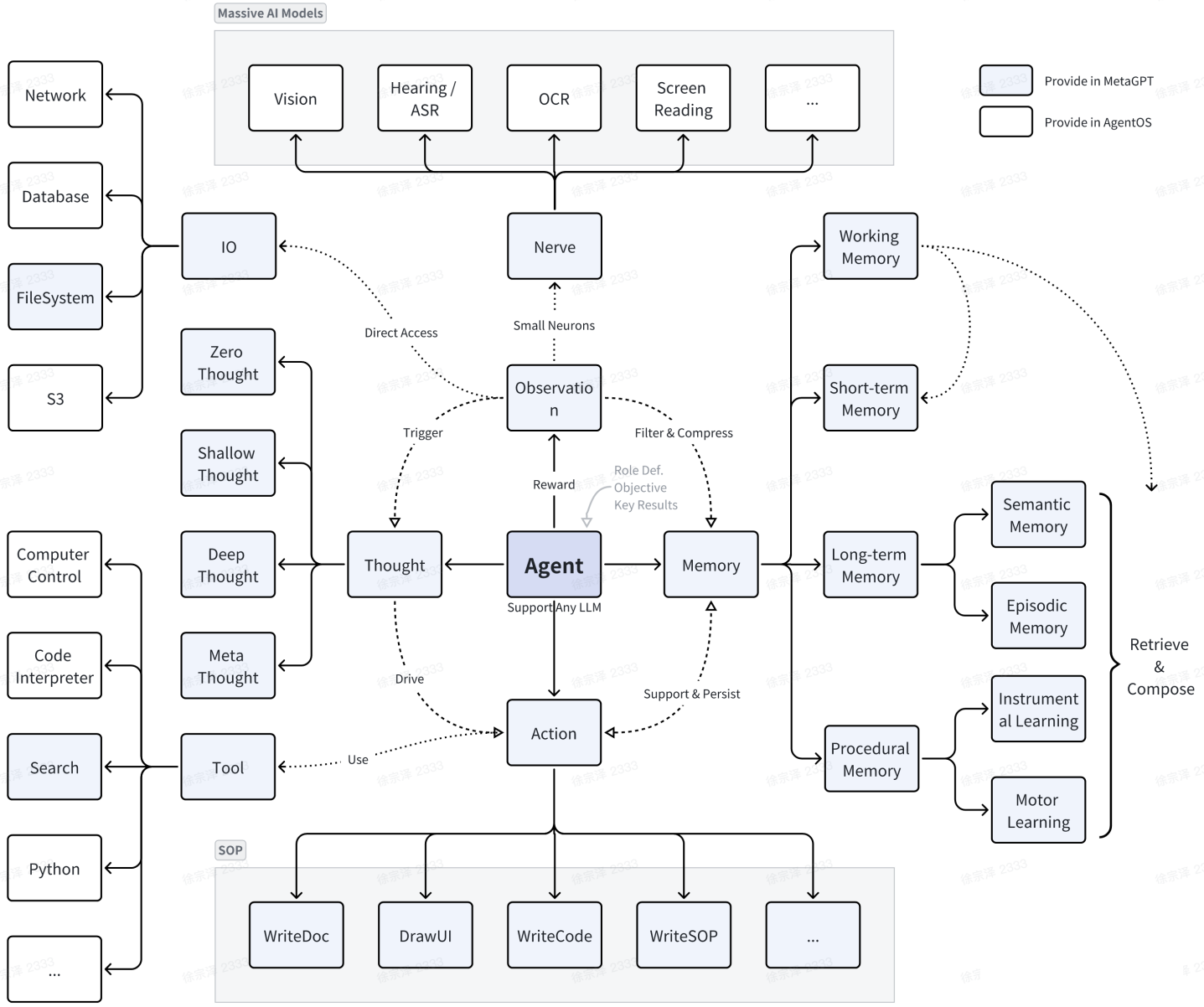
Sirui Hong, Mingchen Zhuge*, (equal contributions)*
Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang,
Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao,
Chenglin Wu†, Jürgen Schmidhuber (email: alexanderwu@deepwisdom.ai)

1.Introduction

What is an agent?

What is multi-agents?





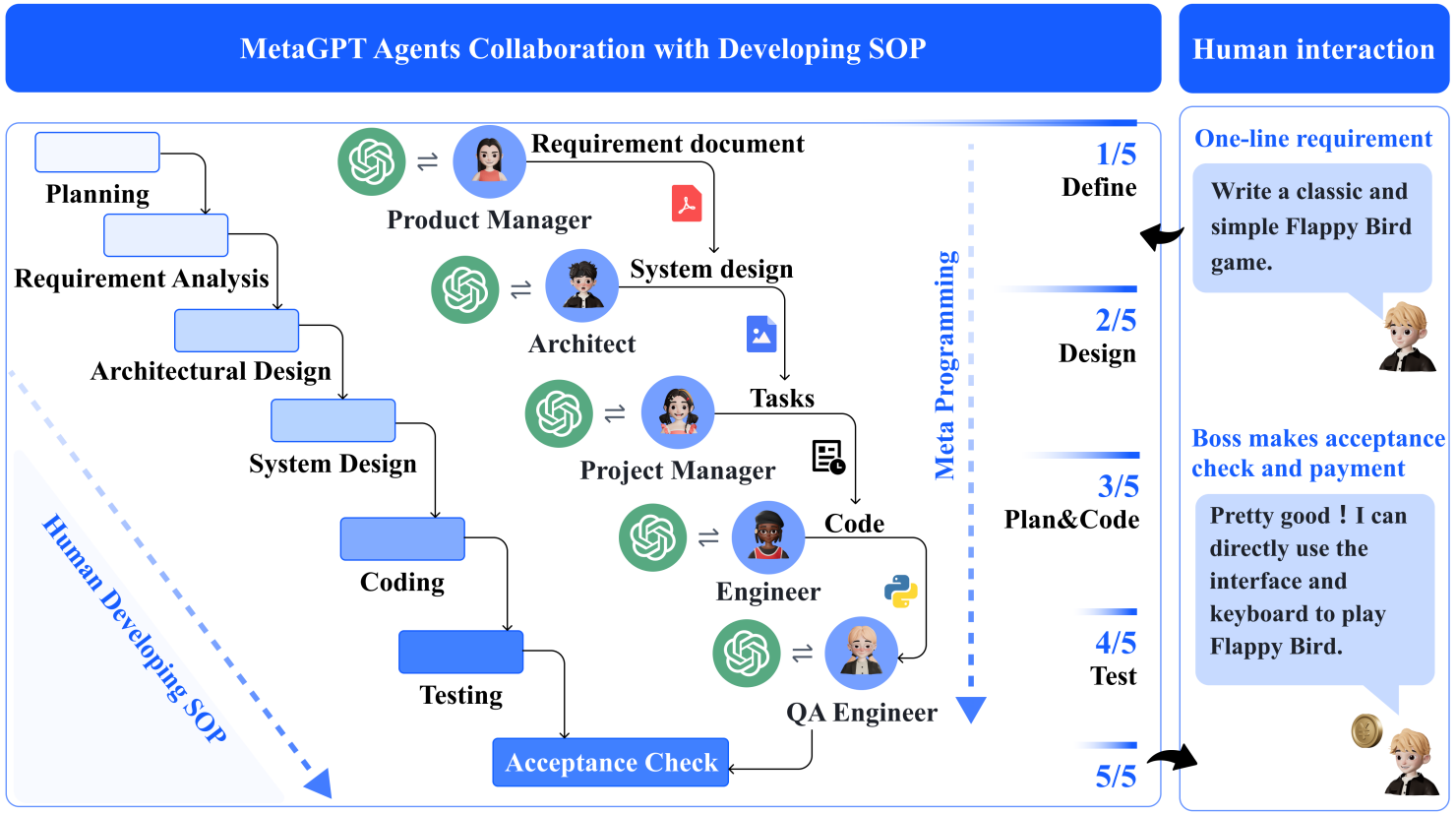
2. Challenges

Hallucinations and Inconsistency,

especially generations with long context

3. Standard Operating Procedure (SOP) in human society

4. Standard Operating Procedure (SOP) in MetaGPT



The software development SOPs between MetaGPT and real-world human teams.

In software engineering, SOPs promote collaboration among various roles. MetaGPT showcases its ability to decompose complex tasks into specific actionable procedures assigned to various roles (e.g., Product Manager, Architect, Engineer, etc.).

5. Use Case

```
"""Run a startup. Be a boss."""
from metagpt.roles import ProductManager, Architect, ProjectManager, Engineer, QaEngineer
from metagpt.team import Team

company = Team()
company.hire([ProductManager(), Architect(), ProjectManager()])

if implement or code_review:
    company.hire([Engineer(n_borg=5, use_code_review=code_review)])

if run_tests:
    company.hire([QaEngineer()])

company.invest(investment)
company.run_project(idea, project_name=project_name, inc=inc)
asyncio.run(company.run(n_round=n_round))
```

6. Agent Collaboration in Software Development

Specialization of Roles

❖ Role Playing Mechanism

Each agent in MetaGPT is designed with a specific role and set of responsibilities, which allows for a division of labor that mirrors real-world software development teams

Each agent in MetaGPT is initialized with the specific context and skills such as web search, diagram tools, etc.

❖ React-style behavior

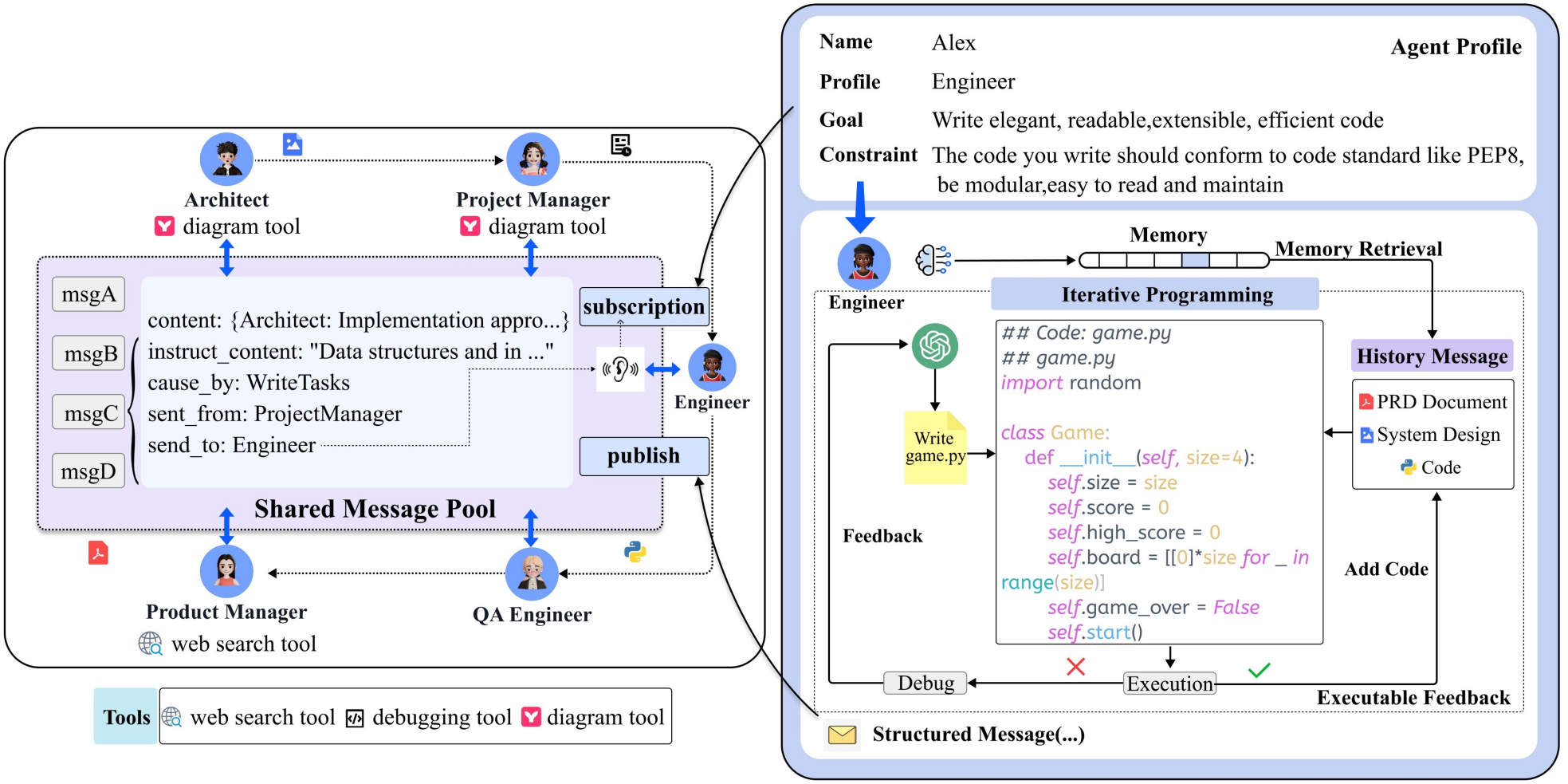
Each agent in MetaGPT adheres to think-act behavior, we extend their observations by providing an environment for communication and execution.

Each agent has independent memory and shared memory, enabling them efficient and reliable in the task process

Workflow across Agents

Agents perform collaboration by utilizing messages in MetaGPT, which either directly trigger actions or assist in finishing a job.

7. Agent Collaboration in Software Development (a)



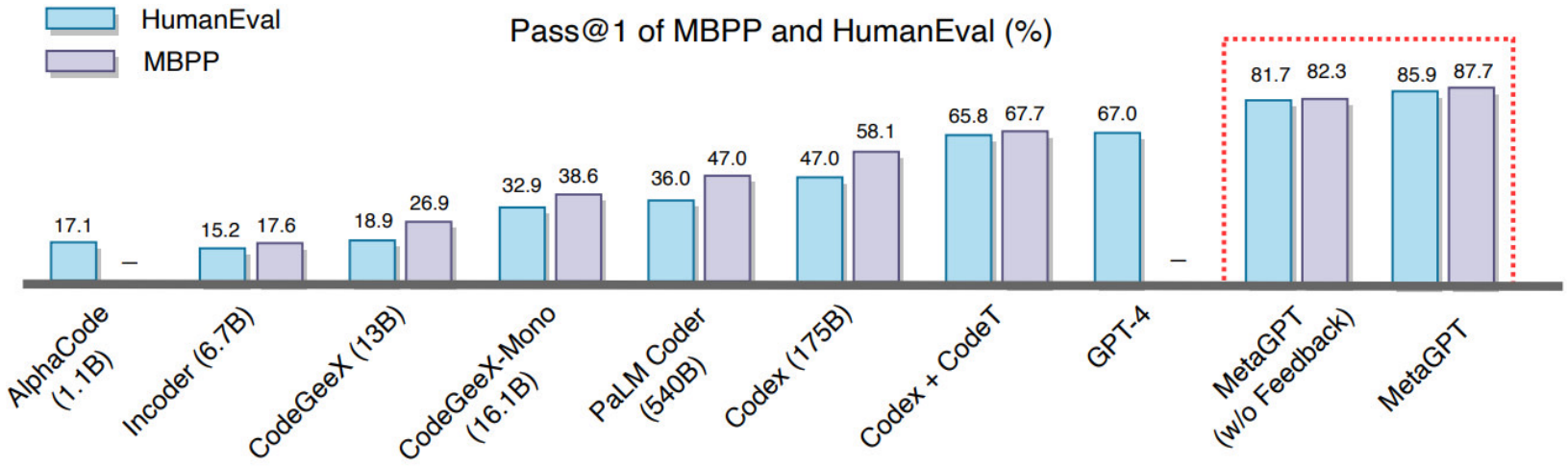
(1) Structured Communication Interfaces

(2) Publish-Subscribe Mechanism

(3) Executable Feedback

(4) Iterative Programming

8. Experiments (a)



MetaGPT achieves **85.9%** and **87.7%** in HumanEval and MBPP respectively

MetaGPT achieves a **28.2%** relative improvement on HumanEval compared to GPT-4. Executable feedback led to a **4.2%** and **5.4%** improvement on HumanEval and MBPP respectively.

MetaGPT achieves a score of **3.75**, which is very close to 4 (flawless).

MetaGPT takes only **503** seconds to finish a task on average.

MetaGPT demonstrates high productivity in code, which needs only **124.3** tokens for one line of code.

Table 1: The statistical analysis on SoftwareDev.

Statistical Index	ChatDev	MetaGPT w/o Feedback	MetaGPT
(A) Executability	2.25	3.67	**3.75**
(B) Cost#1: Running Times (s)	762	**503**	541
(B) Cost#2: Token Usage	**19,292**	24,613	31,255
(C) Code Statistic#1: Code Files	1.9	4.6	**5.1**
(C) Code Statistic#2: Lines of Code per File	40.8	42.3	**49.3**
(C) Code Statistic#3: Total Code Lines	77.5	194.6	**251.4**
(D) Productivity	248.9	126.5	**124.3**
(E) Human Revision Cost	2.5	2.25	**0.83**

8. Experiments (b)

Table 2: **Comparison of capabilities for MetaGPT and other approaches.** '✓' indicates the presence of a specific feature in the corresponding framework, '✗' its absence.

Framework Capabiliy	AutoGPT	LangChain	AgentVerse	ChatDev	MetaGPT
PRD generation	✗	✗	✗	✗	✓
Tenical design genenration	✗	✗	✗	✗	✓
API interface generation	✗	✗	✗	✗	✓
Code generation	✓	✓	✓	✓	✓
Precompilation execution	✗	✗	✗	✗	✓
Role-based task management	✗	✗	✗	✓	✓
Code review	✗	✗	✓	✓	✓

Table 3: **Ablation study on roles.** '#' denotes 'The number of', 'Product' denotes 'Product manager', and 'Project' denotes 'Project manager'. '✓' indicates the addition of a specific role. 'Revisions' refers to 'Human Revision Cost'.

Engineer	Product	Architect	Project	#Agents	#Lines	Expense	Revisions	Executability
✓	✗	✗	✗	1	83.0	**$ 0.915**	10	1.0
✓	✓	✗	✗	2	112.0	$ 1.059	6.5	2.0
✓	✓	✓	✗	3	143.0	$ 1.204	4.0	2.5
✓	✓	✗	✓	3	205.0	$ 1.251	3.5	2.0
✓	✓	✓	✓	**4**	**191.0**	$ 1.385	**2.5**	**4.0**

Table 4: **Executability comparison.** The executability scores are on a grading system ranging from '1' to '4'. A score of '1' signifies complete failure, '2' denotes executable code, '3' represents largely satisfying expected workflow, and '4' indicates a perfect match with expectations.

Task	AutoGPT	LangChain	AgentVerse	ChatDev	MetaGPT
Flappy bird	1	1	1	2	**3**
Tank battle game	1	1	1	2	**4**
2048 game	1	1	1	1	**4**
Snake game	1	1	1	3	**4**
Brick breaker game	1	1	1	1	**4**
Excel data process	1	1	1	**4**	**4**
CRUD manage	1	1	1	2	**4**
Average score	1.0	1.0	1.0	2.1	**3.9**

ABLATION STUDY

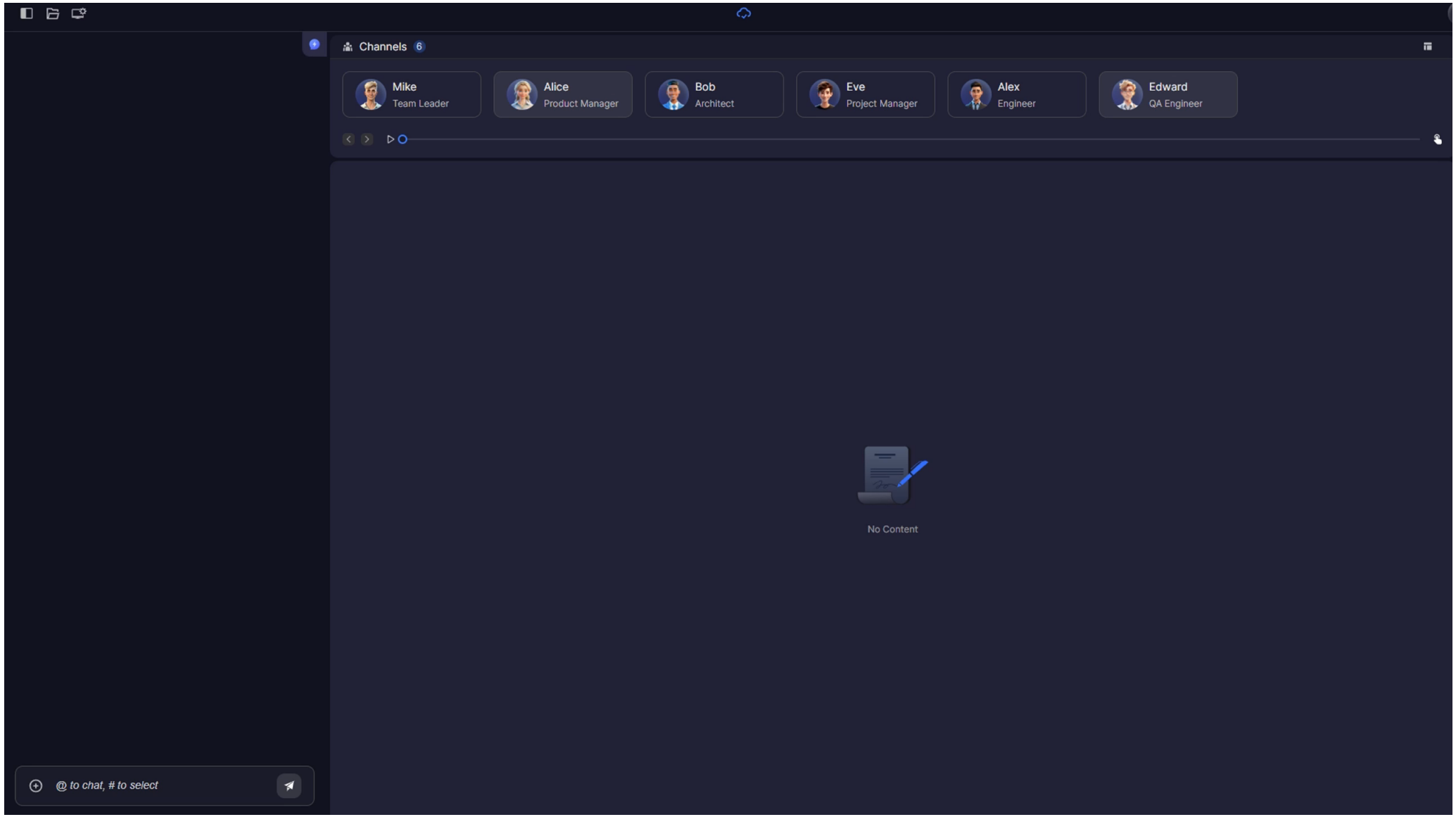
The Effectiveness of Roles

- ✓ Adding roles different from just the Engineer consistently improves revisions and executability.

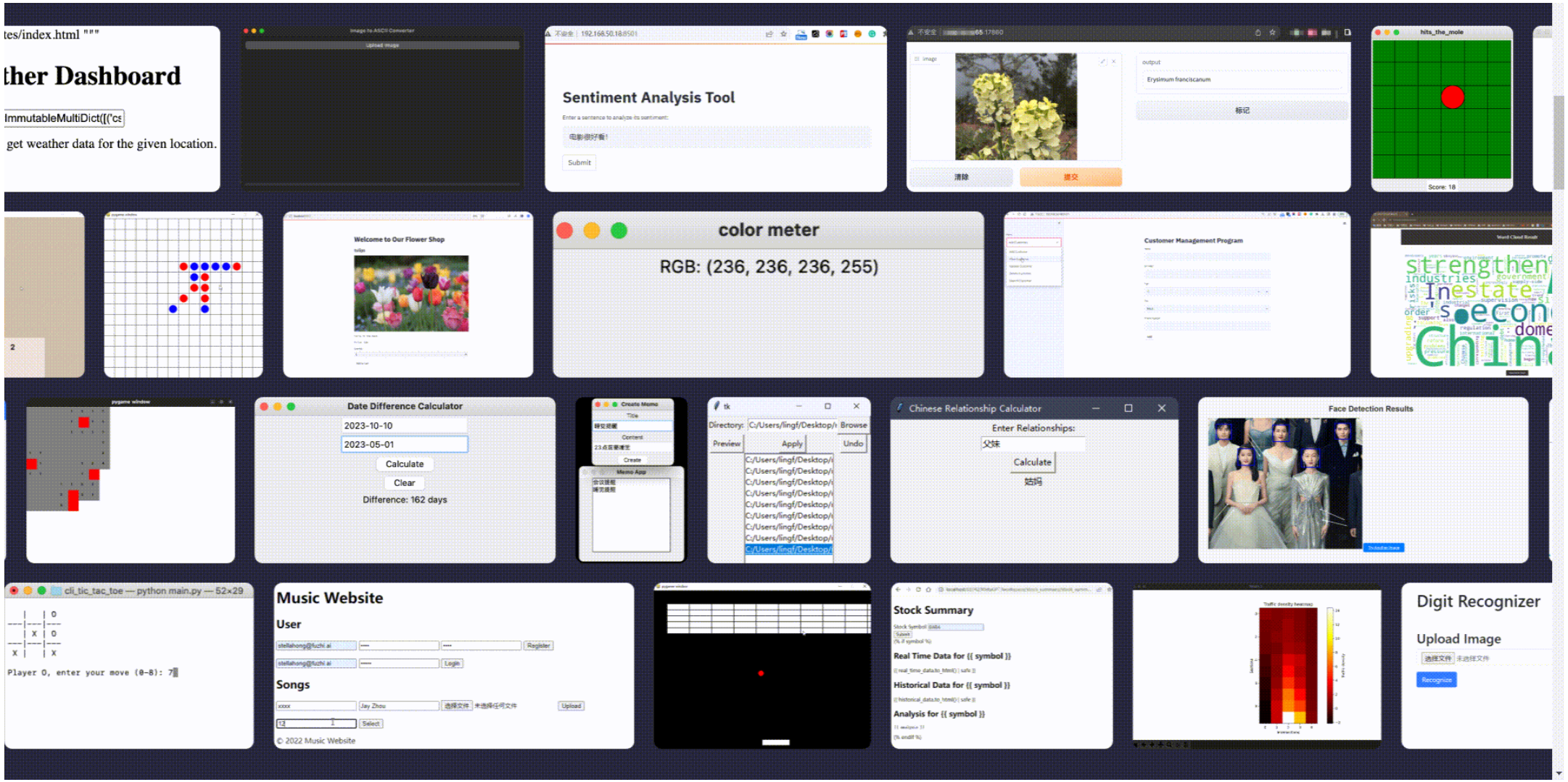
The Effectiveness of Executable Feedback Mechanism

- ✓ adding executable feedback into MetaGPT leads to a significant improvement of 4.2% and 5.4% in Pass @1
- ✓ The feedback mechanism improves feasibility (3.67 to 3.75) and reduces the cost of human revisions (2.25 to 0.83).

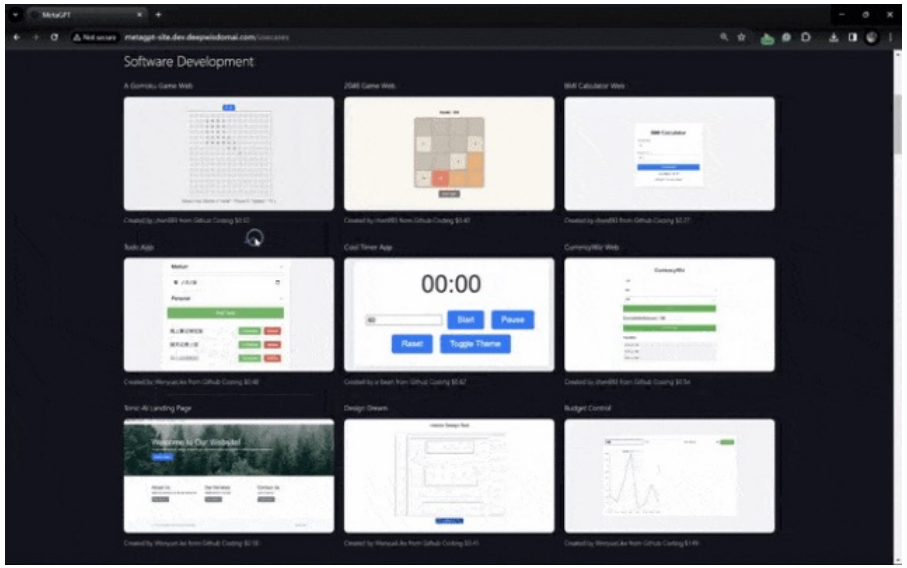
9. Demos: Development Procedures in MetaGPT

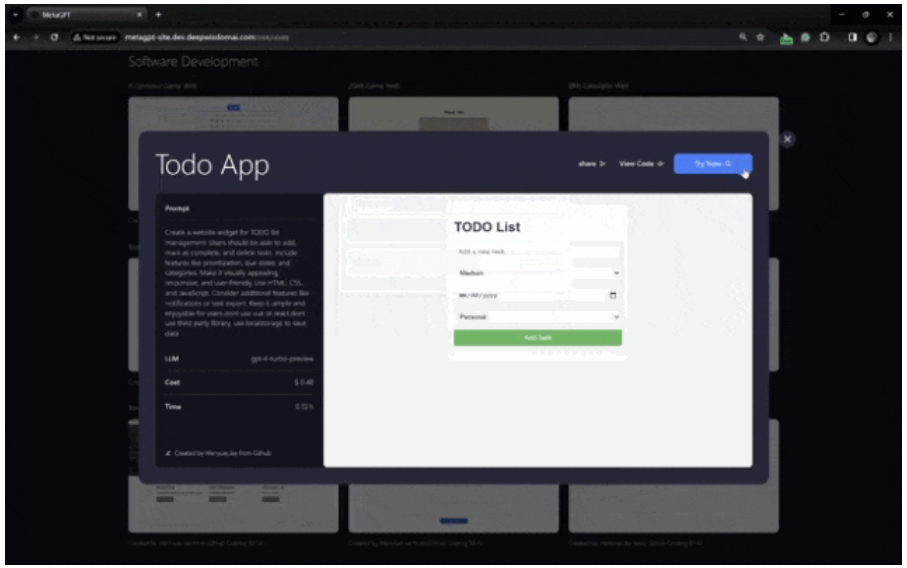


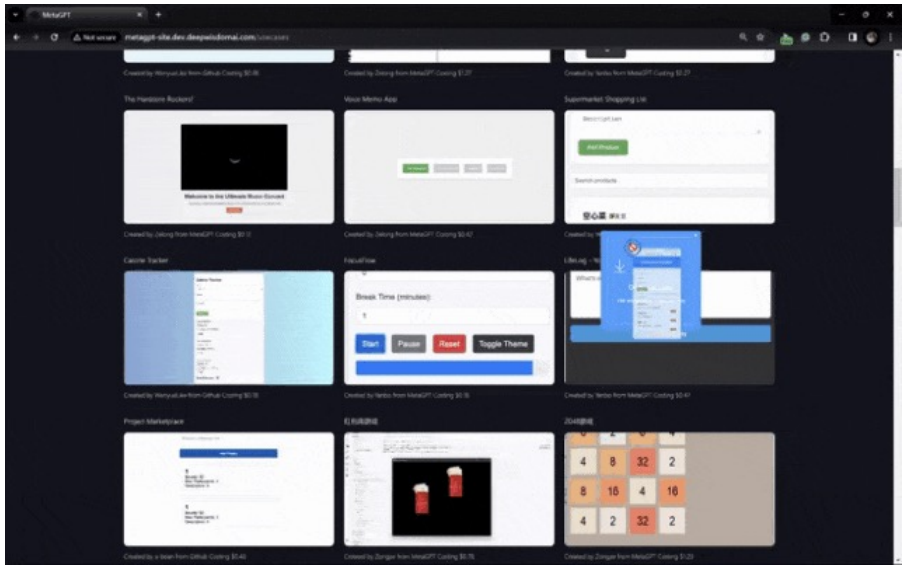
9. Demos: Softwares generated by MetaGPT

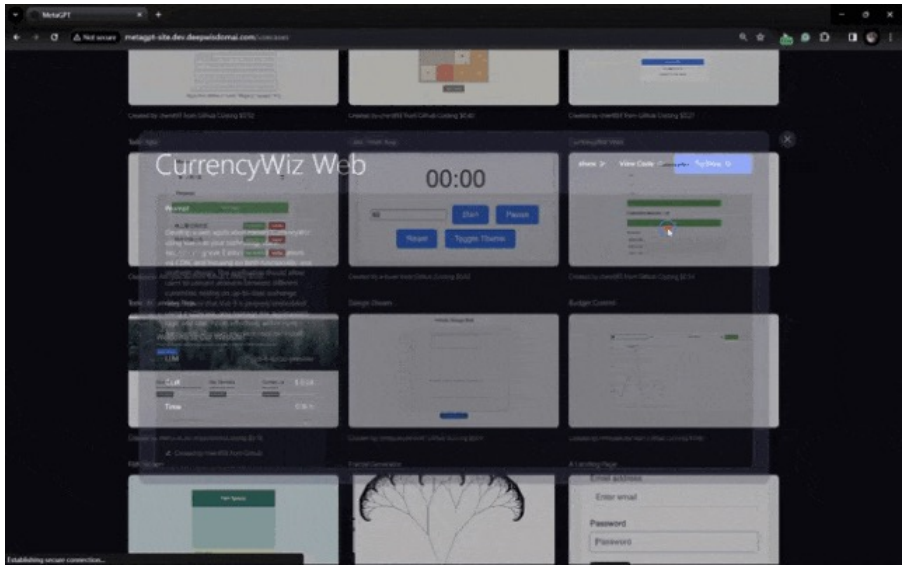


9. Demos: Executions of Generated Softwares









Thanks!