



DeepLearning.AI

Agentic AI

M5: Patterns for highly
autonomous agents



Patterns for highly autonomous agents

Planning workflows

Planning example: Customer service agent

Inventory Database

id	name	description	price	stock
1001	Aviator	Timeless pilot style for any occasion, metal frame	80	12
1002	Catseye	Glamorous 1950s profile, plastic frame	60	28
1003	Moon	Oversized round style, plastic frame	120	15
1004	Classic	Classic round profile, gold frame	60	9

Customer query:

Do you have any **round** sunglasses in stock that are under \$100?

Yes, we have our **Classic** sunglasses, which are a classic round metal frame and cost \$60

Planning example: Customer service agent

system
prompt

You have access to the following
tools:

{description of tools}

Return a step-by-step plan to carry
out the user's request.

Do you have any
round sunglasses
in stock that are
under \$100?

LLM

1. Use **get_item_descriptions** tool to find round sunglasses
2. Use **check_inventory** to see if results are in stock
3. Use **get_item_price** to see if in-stock results are <\$100

Tools

get_item_
descriptions

check_
inventory

process_
item_return

get_item_
price

check_past_
transactions

process_
item_sale

Step 1 text

LLM

get_item_
descriptions

Step 1 output,
Step 2 text

LLM

check_
inventory

Step 2 output,
Step 3 text

LLM

get_item_
price

Planning example: Customer service agent

system
prompt

You have access to the following
tools:

{description of tools}

Return a step-by-step plan to carry
out the user's request.

I would like to
return the gold
frame glasses I
purchased, but not
the metal frame
ones.

LLM

1. Use **check_past_transactions** to find which glasses they bought
2. Use **get_item_descriptions** to find the gold frame glasses
3. Use **process_item_return** to return the gold-framed glasses

Tools

get_item_
descriptions

check_
inventory

process_
item_return

get_item_
price

check_past_
transactions

process_
item_sale

Step 1 text

LLM

check_past_
transactions

Step 1 output,
Step 2 text

LLM

get_item_
descriptions

Step 2 output,
Step 3 text

LLM

process_
item_return

Planning example: Email assistant

system
prompt

You have access to the following
tools:

{description of tools}

Return a step-by-step plan to carry
out the user's request.

Reply to that email
invitation from
Bob about dinner
in New York and
tell him I'll attend.
Then archive his
email.

LLM

1. Use **search_email** to find emails from "Bob" that mention "dinner" and "New York"
2. Use **send_email** tool to reply and confirm attendance
3. Use **move_email** tool to move email to "archive" folder

Tools

search_email

move_email

delete_email

send_email

Step 1 text

LLM

search_email

Step 1 output,
Step 2 text

LLM

send_email

Step 2 output,
Step 3 text

LLM

move_email



Patterns for highly autonomous agents

Creating and executing
LLM plans

Planning example: Customer service agent

system
prompt

You have access to the following tools:

{description of tools}

Return a step-by-step plan to carry out the user's request.

Do you have any round sunglasses in stock that are under \$100?

LLM

1. Use **get_item_descriptions** tool to find round sunglasses
2. Use **check_inventory** to see if results are in stock
3. Use **get_item_price** to see if in-stock results are <\$100

Tools

get_item_descriptions

check_inventory

process_item_return

get_item_price

check_past_transactions

process_item_sale

Step 1 text

LLM

get_item_descriptions

Step 1 output,
Step 2 text

LLM

check_inventory

Step 2 output,
Step 3 text

LLM

get_item_price

Formatting plan as JSON

Updated system prompt

You have access to the following tools:

{description of tools}

Create a step-by-step plan in JSON format.

Each step should have the following items:
step number, description, tool name, and
args.

Do you have any
round sunglasses
in stock that are
under \$100?

LLM

```
{  
  "plan": [  
    {  
      "step": 1,  
      "description": "Find round sunglasses",  
      "tool": "get_item_descriptions",  
      "args": {"query": "round sunglasses"}  
    },  
    {  
      "step": 2,  
      "description": "Check available stock",  
      "tool": "check_inventory",  
      "args": {"items": "results from step 1"}  
    },  
    ...  
  ]  
}
```



Patterns for highly autonomous agents

Planning with code execution

The challenge of planning with tools



Which month had the highest sales of hot chocolate?

LLM

Tools

get_column_max

get_column_mean

filter_rows

get_column_min

get_column_median

sum_rows

date	price	coffee_name	Size
2024-01-28	3.87	Hot Chocolate	M
2024-03-01	2.89	Cappuccino	S
2024-03-04	3.87	Latte	M
...
2025-03-23	4.57	Latte	L

coffee_sales.csv

1. Use the **filter_rows** tool to extract transactions in January with coffee_name "Hot Chocolate"
2. Use the **get_column_mean** to find the average amount
3. Use the **filter_rows** tool to extract transactions in February with coffee_name "Hot Chocolate"
4. Use the **get_column_mean** to find the average amount
5. Repeat for March, April, May, ..., December
6. Determine the month with highest average using results of previous steps

The challenge of planning with tools



Which month had the highest sales of hot chocolate?

LLM

Tools

get_column_max

get_column_mean

filter_rows

get_column_min

get_column_median

sum_rows

date	price	coffee_name	Size
2024-01-28	3.87	Hot Chocolate	M
2024-03-01	2.89	Cappuccino	S
2024-03-04	3.87	Latte	M
...
2025-03-23	4.57	Latte	L

How many unique transactions last week?

get_unique_entries

What were the amounts of the last 5 transactions?

get_last_N_values

...

...

- Brittle
- Inefficient
- Continuously dealing with edge cases

Planning with code execution

system
prompt

Write code to solve the user's query. Return your answer as python code delimited with `<execute_python>` and `</execute_python>` tags.

What were the amounts of the last 5 transactions?

LLM

```
<execute_python>
```

```
import pandas as pd
```

```
# Load CSV
```

```
df = pd.read_csv("transactions.csv")
```

```
# Ensure date column is parsed as datetime
```

```
df["date"] = pd.to_datetime(df["date"])
```

```
# Sort by date to get most recent transactions
```

```
df_sorted = df.sort_values(by="date",  
                           ascending=False)
```

```
# Select the last 5 transactions
```

```
last_5 = df_sorted.head(5)
```

```
# Show just the price column (amounts)
```

```
print("Last 5 transaction amounts:")
```

```
print(last_5["price"].to_list())
```

```
</execute_python>
```

Planning with code execution

system
prompt

Write code to solve the user's query. Return your answer as python code delimited with `<execute_python>` and `</execute_python>` tags.

How many unique transactions last week?

LLM

```
<execute_python>
```

```
import pandas as pd
```

```
# Read CSV and parse the "date" column as datetime
df = pd.read_csv("transactions.csv",
                 parse_dates=["date"])
```

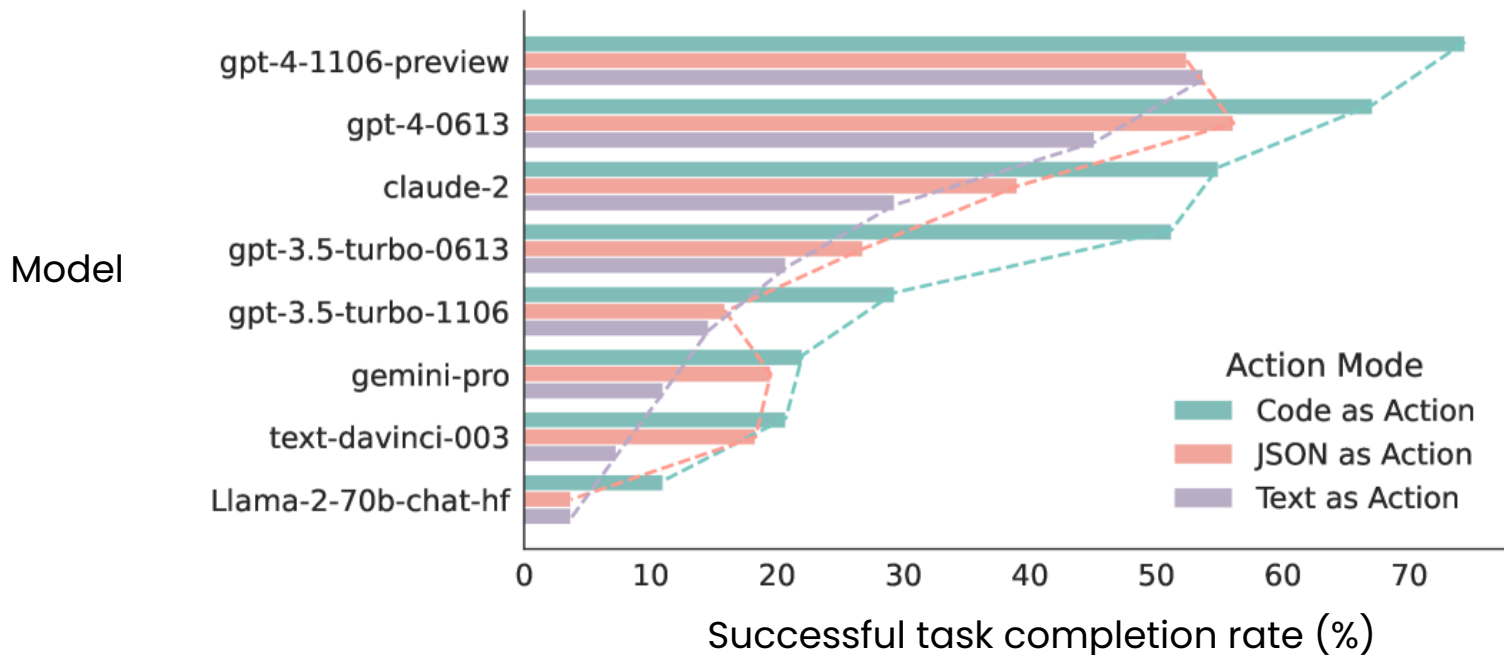
```
# Define time window
today = pd.Timestamp.today()
week_ago = today - pd.Timedelta(days=7)
```

```
# Filter rows where date is within last week
last_week = df[df["date"].between(week_ago, today)]
```

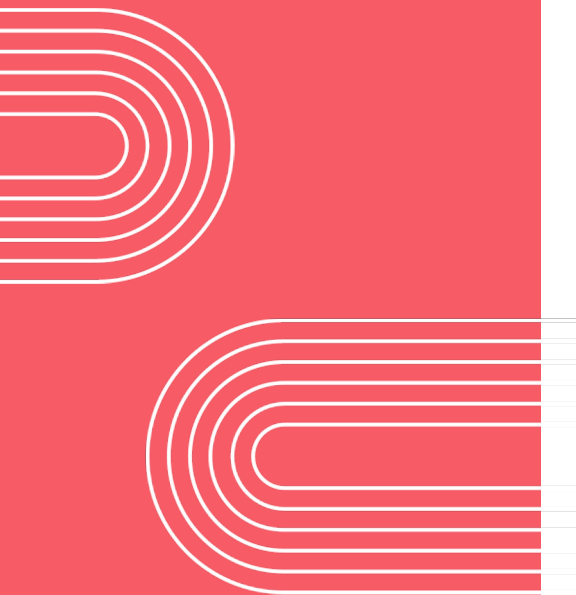
```
# Drop duplicate rows and count
print(last_week.drop_duplicates().shape[0])
```

```
</execute_python>
```

Planning with code improves performance



[Adapted from "Executable Code actions Elicit Better LLM Agents", Wang et al. 2024]



Patterns for highly autonomous agents

Multi-agentic workflows

Some tasks require more than 1 person!

Task	Team
Create marketing assets	Researcher Graphic Designer Writer
Writing a research article	Researcher Statistician Lead writer Editor
Preparing a legal case	Associate Paralegal Investigator

Example: Marketing team

Researcher

Tasks

- Analyze market trends
- Research competitors

Tools

- Web search

researcher

Graphic designer

Tasks

- Create data visualizations
- Create artwork

Tools

- Image generation, manipulation
- Code execution for chart generation

graphic
designer

Writer

Tasks

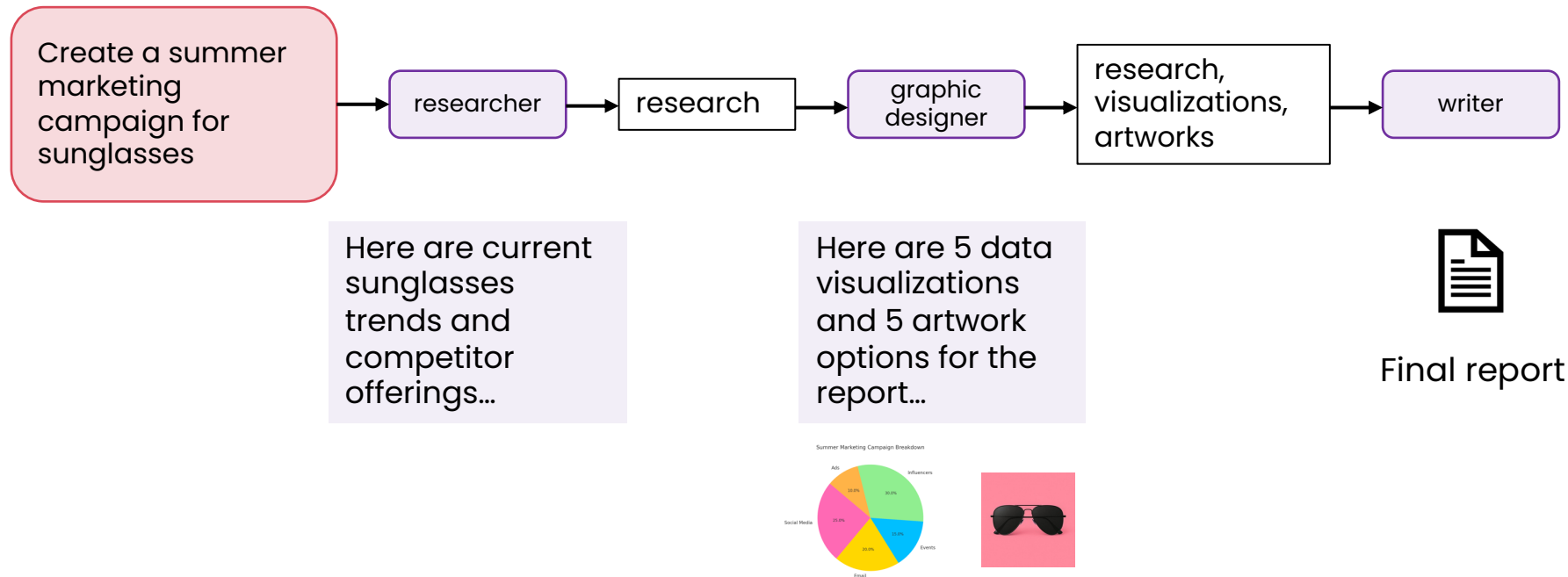
- Transform research into report text and marketing copy

Tools

- (None)

writer

Example: Marketing team with linear plan



Example: Planning with multiple agents

system
prompt

You have access to the following tools:

{description of tools}

Return a step-by-step plan to carry out the user's request.

Tools

get_item_
descriptions

check_
inventory

process_
item_return

get_item_
price

check_past_
transactions

process_
item_sale

Create a summer
marketing
campaign for
sunglasses

LLM

Example: Planning with multiple agents

system
prompt

You have access to the following
tools:

{description of tools}

Return a step-by-step plan to carry
out the user's request.

Agents

researcher

graphic
designer

writer

Create a summer
marketing
campaign for
sunglasses

LLM

Example: Planning with multiple agents

system
prompt

You are a marketing manager and
have the following team of agents to
work with:

{description of agents}

Return a step-by-step plan to carry
out the user's request.

Agents

researcher

graphic
designer

writer

Create a summer
marketing
campaign for
sunglasses

LLM

Example: Planning with multiple agents

system
prompt

You are a marketing manager and
have the following team of agents to
work with:

{description of agents}

Return a step-by-step plan to carry
out the user's request.

Create a summer
marketing
campaign for
sunglasses

LLM

marketing
manager

1. Ask **researcher** to research
current sunglasses trends
2. Ask **graphic designer** to
create ad images
3. Ask **writer** to create report
4. Review report

Agents

researcher

graphic
designer

writer

Step 1 text

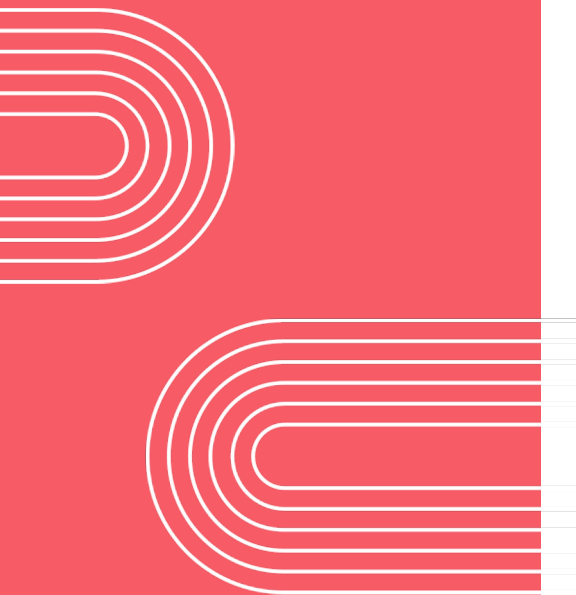
researcher

Step 1 output,
Step 2 text

graphic
designer

Step 2 output,
Step 3 text

writer



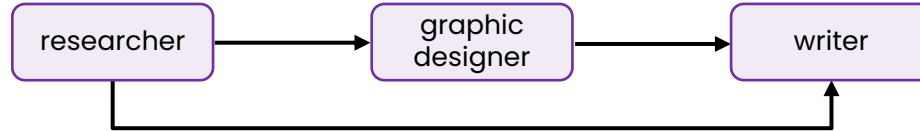
Patterns for highly autonomous agents

Communication patterns
for multi-agent systems

Example: Marketing team with linear plan

Create a summary
marketing
campaign for
sunglasses

Communication Pattern

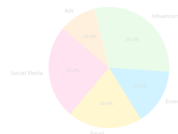


writer



Final report

Summer marketing campaign breakdown



Example: Planning with multiple agents

system
prompt

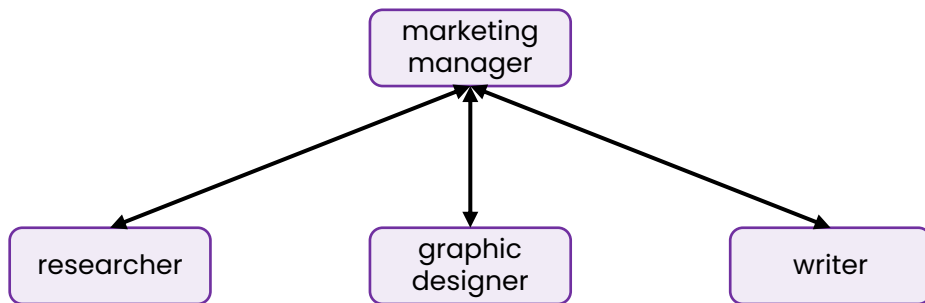
You are a marketing manager and
have the following team of agents to
work with:

Agents

graphic

writer

Communication Pattern



Create a summ
marketing
campaign for
sunglasses

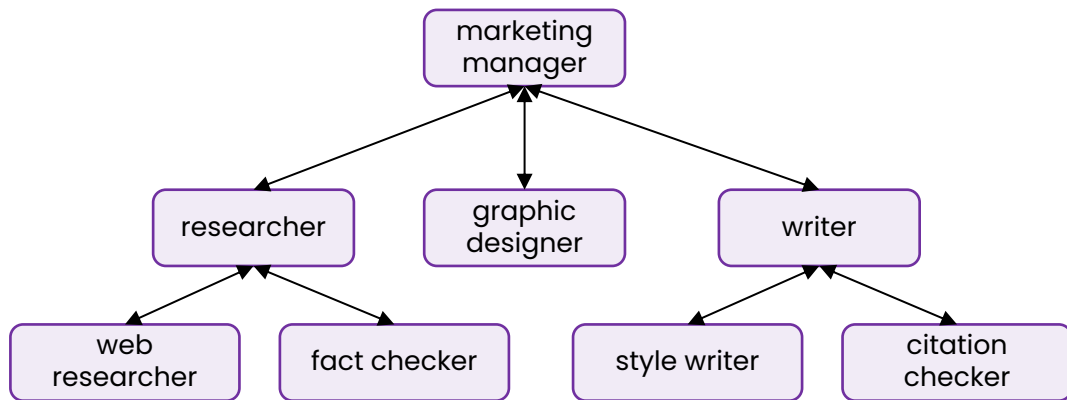
3. Ask **writer** to create report
4. Review report

Step 2 output,
Step 3 text

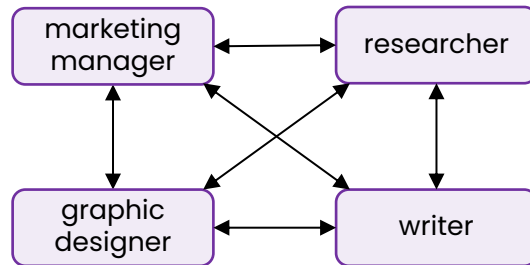
writer

Other communication patterns

Deeper hierarchy



All-to-all





Agentic AI

Conclusion

Summary

- Why Agentic AI
- Reflection design pattern
- Tool use (function calling)
- Evals, error analysis
- Planning, multi-agent systems



End of M5
