

# ???

- [\[REDACTED\]](#)
- [\[REDACTED\]-Harness12\[REDACTED\]](#)
- [\[REDACTED\]](#)
- [Agent Harness \[REDACTED\] 12 \[REDACTED\]](#)
- [Agentic Engineering \[REDACTED\]](#)
- [\[REDACTED\] Agent\[REDACTED\]](#)



# 1.3 ????????

- <https://agix.host/books/013d4/page/75fc0>
- Transformer [Transformer](#) /[Transformer](#)
  - Meta [Free Transformer](#) Transformer
- <https://zhuanlan.zhihu.com/p/1909206010096259088> **CoALA** [LLM](#) [LLM](#)

# ??????????

## 2.1 ??????????????????????

- [Transformer](#)
- [Transformer](#)
- [Transformer](#)
- [Transformer](#)

## 2.2 ??????????????

- [Transformer](#)
- [Transformer](#)
- [Transformer](#)

```
“ Transformer WeaveAgent  
Transformer — Transformer Frame  
Transformer  
Agent/WeaveAgentDesign.md §13
```

## 2.3 ??????????????????

- [Transformer](#)
- [Transformer](#)
- [Transformer](#)
- [Transformer](#)

## 2.4 ??????????????????????





```

MergeKeywords [ ]
[ ]
[ ]
CRUD
" [ ] "
Agent/WeaveAgentDesign.md $8 [ ]

```

- [ ] [ ]
  - [ ]
  - [ ] skills / tools [ ]
    - [ ]
    - [ ]
    - GSD [ ]
    - [ ]

## 4.4 ?? / Frame ??????????????????

```

“ [ ] Frame [ ] ---
[ ]
[ ]
handle

```

### 4.4.1 ??/Frame ????

- [ ] ID [ ]
- [ ] / [ ] / [ ]
- [ ]
- [ ] [ ] [ ]
- [ ] [ ] [ ]
- [ ] [ ] [ ]
- [ ]

### 4.4.2 ??/Frame ???

- [ ] [ ] [ ] / [ ]
- [ ] [ ] [ ]
- [ ]
- [ ]
- [ ]

### 4.4.3 ??????????????

- [ ] [ ] + [ ] [ ]
- LLM [ ]
- SubFrame List nested to tree [ ]
- [ ] = [ ]

- o [ ]
- o [ ] skill [ ] prompt [ ]
- o LLM [ ]
- o [ ] " [ ] "
- o [ ] token
- o [ ]
- o [ ] skill [ ] prompt [ ]
- o [ ] / [ ]

#### 4.4.4 ?????

- [ ]
- [ ]

#### 4.4.5 ??????LLM ? Frame ??????

[ ] " [ ] TodoWrite" — LLM [ ]  
 [ ] + [ ] = **prompt** [ ] [ ]

- [ ]
- [ ] xxx [ ] xxx [ ]
- [ ]
- [ ] 5 [ ]
- [ ] xxx [ ]
- [ ] skill [ ]
- [ ]
- [ ]
- [ ] / [ ] / [ ]
- [ ] [ ]

#### 4.4.6 Frame ???

- [ ] Frame [ ] — [ ]
- [ ] SUBFRAME\_DONE [ ]
- [ ] Frame [ ]
- [ ] Frame [ ] Session [ ]
  - o task [ ] / [ ] + [ ]
  - o task [ ]
- [ ] Frame [ ] **session** [ ] [ ] **Frame** [ ] [ ]

#### 4.4.7 Frame ?????

- [ ] Frame [ ] Agent [ ] ToolRegistry [ ] Frame [ ]
- [ ] [ ] [ ] [ ] memory/\* [ ] frame/\* [ ] file/\* [ ] skill/\* [ ]



[[[ " " ]]

[[[	[[[	[[[
[[[ / [[[ / [[[	TOP [[[ System [[[	[[[ [[[
[[[	TOP [[[ \$4.4.7 [[[	[[[ [[[
[[[ / [[[	BOT [[[	[[[ [[[
Frame [[[	Frame [[[ [[[ \$4.4.2 [[[	[[[ [[[ prompt [[[
[[[	MID	[[[ [[[ --[[[ [[[ <b>Frame</b> [[[ /[[[ /[[[ [[[

[[[ MID [[[ " " " " \$4.4.5 [[[ " / [[[ / [[[ " ]]

- **Frame** [[[ [[[ Frame [[[
- [[[ [[[ Session [[[
- [[[ / [[[ [[[

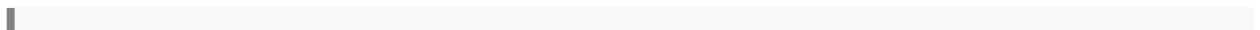
MID [[[ agent [[[ " " " --LLM ]]

### Anthropic Compaction [[[ ]]

[[[	Anthropic Claude Code	[[[
[[[	[[[	[[[ 80%[[[
[[[	LLM [[[ [[[ <b>bug</b>	[[[ [[[ LLM [[[
[[[	[[[ LLM call	[[[ LLM [[[ O(1) [[[
[[[	[[[	[[[ Frame [[[ /[[[
[[[	[[[	MID [[[ BOT [[[

[[[ ]]

1. BOT [[[ ]] ≤ 8K token Frame [[[ ]] ≤ 2K token  
[[[ MID [[[ ]]
2. [[[ ]] **Frame** [[[ ]] MID [[[ ]] MID [[[ ]]
3. [[[ ]] **Frame** [[[ ]] tool\_call/tool\_result [[[ ]]
4. [[[ ]] **LLM**[[[ ]] TOP + BOT  
[[[ ]] MID ]]





# ??????

## 5.1 ??????????

- [ ] [ ] Frame [ ]
- [ ] [ ]
  - [ ]
  - [ ]
  - [ ]
- [ ] [ ]
- [ ] [ ]

## 5.2 Frame ????? §4.4?

- [ ] handle [ ]
- [ ]
  - SubFrame List nested to tree [ ]
  - [ ]
  - [ ]
  - Tool [ ]
  - [ ] Agent [ ]
  - Frame [ ] §4.4.6 [ ]
  - Session [ ] Frame [ ]

## 5.3 Prompt ??

- [ ] **prompt** [ ] [ ]
  - [ ]
  - [ ]
  - [ ]
  - [ ]
  - [ ]
    - [ ]
    - [ ] git [ ]
    - [ ]
- **Prompt** [ ] [ ]
  - [ ] system prompt [ ]
  - [ ]
  - **Prompt** [ ] [ ]
- **Frame** [ ] [ ] prompt [ ]







#	Harness			
7	State Management		\$4.4.2 \$4.4.6 \$5.1	Frame + SUBFRAME_DONE git checkpoint
8	Error Handling	△	\$4.4.3 \$4.4.5	" " "+" "
9	Guardrails & Safety	△	\$4.1 \$6.6 \$3.1	" " + " + " " tripwire /
10	Verification Loops	△	\$4.4.4 \$4.5.2	" " " LLM-as-judge rules-based / visual
11	Subagent Orchestration		\$4.4.3 \$4.4.6 \$6.3 sub-agent	SubFrame nested-to-tree Fork/Worktree
12	Long-Horizon Execution		\$4.5.4 \$6.4	" " + " " + " " = " Ralph Loop

Guardrails & Safety 12 7 3 2 Output Parsing

????????

# 1. Orchestration Loop?????— ? ??

**Harness** TAO Thought-Action-Observation —assemble prompt → call LLM → parse → execute tools → loop Anthropic "dumb loop"

- \$4.5 / "dumb loop" —
- \$4.5.4 " " " " " " ↔

- §2.4 在 "..." / ... / ... "—" TAO

loop ... + ... "harness ... " ... Anthropic ... dumb

## 2. Tools????— ???

**Harness** ... Schema ... name/description/parameters ... Tool  
 Claude Code ... 6 ...

...

- §4.4.7 **Frame** ...
  - ... ToolRegistry ... Frame ...
  - ... memory/\* ... frame/\* ... file/\* ... skill/\*
  - ... Frame.Next() ... prompt ... / ... **System** ...
- §5.4 ... **LLM** ... Frame/Action ...

... **PDF** ...

- "Registry + ... + ... " ... **Harness** ... **6** ... **Tool**  
**Scoping** ... Vercel ... 80% ...
- ... Claude Code ... " / Skills" ...

## 3. Memory????— ??? + ???

**Harness** ... session ... + ... session ... Anthropic ... **agent** ... **hint**  
 ~150 ... → ... → ... transcript ...

...

- §2.2 ... " ... "
- §4.2 ... Frame ...
- §4.3 ...
- §5.1 ...

... **Claude Code** ...

- Claude Code ... CLAUDE.md + MEMORY.md ...
- ... + **Info** ... + ... \$4.3 ... WeaveAgentDesign.md \$8

- MindStore Qdrant +

Anthropic "agent hint " " §4.3  
 " Frame "

## 4. Context Management??????— ? ?? + ??

**Harness** Context Rot Compaction / Observation Masking / Just-in-time Retrieval / Sub-agent Delegation Anthropic " token "

Harness	
Compaction	§4.4.8 + " " LLM O(1)
Observation Masking	§4.4.7
Just-in-time Retrieval	§4.4.7 + §6.3 "
Sub-agent Delegation	§4.4.3 SubFrame nested-to-tree + §6.3 sub-agent
Frame	§4.4.6 " Frame session Frame "

**Compaction** —

- §3.4 " LLM " "
- §6.1 " LLM " = Context Rot
- "Frame + " " PDF 3 "sub-agent delegation + structured note-taking"
- **§4.4.8** Anthropic Compaction
  - TOP System BOT  
 $\text{len(TOP)+len(BOT)} \ll \text{len(LLM\_max\_context)}$
  - = MID LLM O(1)
  - MID " " §4.4.5 " / /  
 " Frame / / —LLM

	Anthropic Claude Code	§4.4.8
		80%
	LLM	MID
	1 LLM call	0 LLM call
		Frame /



Harness — Frame §4.4.5 / / / / / / / / tool tool calling

# 7. State Management??????— ? ??

Harness LangGraph typed dict + checkpoint OpenAI Claude Code git commit checkpoint + progress

- §4.4.2 Frame /
- §4.4.6 Frame Frame SUBFRAME\_DONE
- §5.1

- Frame 7 LangGraph typed dict
- 

checkpoint / LangGraph super-step checkpoint Claude Code "git commit checkpoint" Frame git

# 8. Error Handling??????— ?? ??

Harness 10 × 99% = 90.4% LangGraph Transient retry LLM-recoverable ToolMessage User-fixable interrupt Unexpected bubble up Stripe 2

- §4.4.3 token LLM-recoverable
- §4.4.5 5

- " " bubble up
- retry/backoff interrupt bubble up
- Stripe 2
- 99%

# 9. Guardrails & Safety

Harness OpenAI input/output/tool guardrails + tripwire Anthropic Claude Code ~40

- §4.1
- §3.1
- §4.4.4
- §6.6

- +
- Harness input/output/tool guardrails + tripwire
- guardrail

§5

- Tool Anthropic ~40
- API tripwire
- /

# 10. Verification Loops

Harness verification—Rules-based /linter/ Visual Playwright LLM-as-judge subagent Boris Cherny 2-3x

- §4.4.4 LLM-as-judge
- §4.5.2
- §3.2

LLM-as-judge

- LLM

- **rules-based** linter **ground truth**
- **visual** UI /

§4.4.4 " "—

1. schema
2. / /
3. LLM

Martin Fowler/Thoughtworks **guides** **feedforward** **vs sensors** **feedback**

## 11. Subagent Orchestration?? agent ???— ? ??

**Harness** Claude Code Fork Teammate +  
 Worktree git OpenAI agents-as-tools + handoffs LangGraph  
 state graph

- **§4.4.3** "SubFrame List nested to tree" = LangGraph state graph
- **§4.4.6 Frame** Frame `SUBFRAME_DONE` LangGraph reducer
- **§6.3** sub-agent

- "Frame + " = LangGraph " "
- **Frame ID +** + **Session/Frame** §4.4.6 = Anthropic Fork

Claude Code Worktree

## 12. Long-Horizon Execution????? / Ralph Loop ?— ? ?? + ??

**Harness** Anthropic Ralph Loop—Initializer Agent init script + progress  
 + feature + commit Coding Agent git log + progress  
 feature → → commit →

















# ??LLM ??????????????

???????????? LLM ??? ? ??? ?

## LLM ???

1. ??? --????????????
2. ??? --????????????????????????????????????
3. ??? ??? --???????????? "?????" "?????"

## LLM ????

1. ??? --????????????
2. ??? --? top-K ???
3. ? ID--????????
4. ??? / ? --LLM  
??  
????????????

???????? LLM ????????????????????????? ?

# ????????????????

??	?? / ???	?????	??
???	?????	?????	?? + ????
???	????????	????? + ?	????????
???	????????	????????	????????
?????	?? / ???	??	???
"???" "???"	???????? top-K?	??	????????
???	????????	????????	????????
???	?????	??	????????

???? ? ? ???? LLM ?????? LLM ?  
?? "???" /???????? "???????? wiki  
????????

???????



# Agent Harness ??????????????

## 12 ???

“ Akshay Pachaar The Anatomy of an Agent Harness 2026-04-06  
[https://x.com/akshay\\_pachaar/status/2041146899319971922](https://x.com/akshay_pachaar/status/2041146899319971922)  
Agent/design/harness.pdf

## ????? Agent Harness

chatbot ReAct —demo

“

LangChain TerminalBench 2.0 " " pass rate 76.4%

Agent Harness

### 1.1 ??

“ Harness LLM

- Anthropic Claude Code SDK "the agent harness that powers Claude Code"
- OpenAI Codex "agent" "harness"
- LangChain Vivek Trivedy



Anthropic OpenAI LangChain

# 1. Orchestration Loop??????

TAO Thought-Action-Observation ReAct loop

prompt → LLM → tool calls →

while

```
Anthropic runtime "dumb loop" harness
```

# 2. Tools????

agent Schema name description parameter types LLM Tool

- registration
- Schema validation
- argument extraction
- sandboxed execution
- result capture
- observation

- Claude Code 6 Web subagent
- OpenAI Agents SDK function tools @function\_tool + hosted tools WebSearch / CodeInterpreter / FileSearch + MCP server tools

# 3. Memory????

- Short-term memory

- **Long-term memory**

- Anthropic `CLAUDE.md` + `MEMORY.md`
- LangGraph namespace JSON Stores
- OpenAI SQLite/Redis Sessions

### Claude Code

1. ~150
- 2.
3. transcript

```
“ agent “
```

## 4. Context Management??????

agent Context Rot

- 30%+ Chroma Stanford "Lost in the Middle"
- token

<b>Compaction</b>	Claude Code bug
<b>Observation Masking</b>	JetBrains Junie
<b>Just-in-time Retrieval</b>	Claude Code grep / glob / head / tail
<b>Sub-agent Delegation</b>	agent 1k-2k token

```
“ Anthropic context engineering token “
```

## 5. Prompt Construction? Prompt ???









<b>Anthropic Claude Agent SDK</b>	<pre> query() </pre>	"dumb loop" Gather-Act-Verify gather context → take action → verify results → repeat
<b>OpenAI Agents SDK</b>	<pre> Runner async/sync/streamed </pre>	<b>Code-first</b> Python graph DSL
<b>OpenAI Codex Harness</b>		Codex Core agent + runtime + App Server JSON-RPC + CLI / VS Code / Web <b>Codex harness</b>
<b>LangGraph</b>	<pre> state graph llm_call tool_node + </pre>	AgentExecutor agent
<b>LangChain Deep Agents</b>	"agent harness"	+ write_todos + + subagent +
<b>CrewAI</b>	Agent / Task / Crew + Flows	role/goal/backstory/tools Flows "intelligence where it matters"
<b>AutoGen → Microsoft Agent Framework</b>	Core / AgentChat / Extensions	sequential / concurrent (fan-out/fan-in) / group chat / handoff / <b>magentic</b> manager agent

# ?????? Harness ? 7 ??????

#		trade-off
1	agent vs agent	Anthropic & OpenAI agent agent agent LLM handoff <b>10</b>
2	ReAct vs Plan-and-Execute	ReAct Plan-and-Execute <b>LLMCompiler</b> ReAct 3.6x
3		time-based clearing / summarization / observation masking / structured note-taking / sub-agent delegation <b>ACON 26-54% token</b> + 95%+ reasoning trace raw tool output

#		trade-off
4		Computational linter = ground truth Inferential LLM-as-judge = Martin Fowler / Thoughtworks guides feedforward vs sensors feedback
5		Permissive vs Restrictive
6		Vercel v0 80% Claude Code 95%
7	Harness	harness Anthropic harness Anthropic Claude Code harness

????????????

## 7.1 ??????????????????

- =
- 
- 

“ → harness

Manus 6 5

- → shell
- "Management agents" → handoff





# Agentic Engineering ??????

## ????????-Agent???

???????? Code[] Cowork [] [][] OpenClaw [][] Agent[]

1. []
  1. []
2. [] Task[] SubAgent[] Skills[] Mcp[]
  1. []
  2. []
  3. MCP []

## ?????

1. [] [] [] []
  1. []
  2. LLM[]
  3. [] 1M[]
  4. [] LLM[]
2. []
  1. []
3. []
  1. LLM[]
  2. []
4. []
5. [] todo list[]
  1. []
6. []
  1. []
  2. []

## ????

1. [] xml[] PCB[] xml
2. []
  1. []
  2. []
  3. []
  4. []
3. []
  1. []
4. []

1. 代码 “\_” 和 “\_” 的 token 数量 /
5. 代码 json 的 json 数量
6. 代码 CLAUDE.md 的 LLM 数量  
tool 数量

????????

1. LLM 数量 Agent 数量
2. 代码数量
3. LLM 数量
4. TodoWrite 数量 LLM 数量
5. 代码数量
6. AI
7. 代码数量

???

代码数量

代码

1. AI Agent 数量
2. 代码数量 50 代码数量
3. AI Coding  
代码数量  
Prompt 数量
4. 代码数量
5. 代码数量
6. 代码数量
7. 代码数量
8. 代码数量
9. 代码数量

代码 AI 数量

代码数量

代码

代码数量

代码数量

代码数量



1. [ OpenClaw ] Claude Code

[ ]

[ ]

2. [ ] harness [ ] skills [ ]

3. OpenClaw [ ] Agent

[ ]

AGI

[ ]

4. [ ]

[ ]

[ ]

# ??Agent???

## Agent?????

1. [ ] AI [ ]
2. [ ] AI  
[ ]  
[ ] AI  
[ ] AI [ ]  
[ ]  
[ ]
3. AI [ ] Facebook [ ] Google [ ] GPU
4. [ ] Gemini 3 [ ] OpenAI  
[ ]  
[ ] AI [ ]  
[ ]
5. [ ] AI  
[ ] 2026 [ ] OpenAI  
[ ]

## ??????

1. MemRL [ ]  
Lifelong Learning [ ]
2. [ ]
  1. [ ] agent [ ]
  2. [ ] fast rcnn [ ] LLM [ ]
  3. [ ] agent [ ] [ ] [ ]
  4. [ ] [ ]
3. [ ]

## [ ]

1. [ ]
2. [ ] RAG
3. [ ] RLM [ ] token [ ] Python REPL [ ]

<https://mp.weixin.qq.com/s/Kg5oiN4LUWPDuW6ngTIP5A>

