

大富翁全链路排查机器人学习文档

测试开发视角的 Hermes Agent 速通与技术沉淀

版本：2026-06-25 | 适用对象：测试开发实习生

文档定位

- 本文面向软件工程专业研究生、测试开发实习生，目标是在约 3 天内理解团队内部“大富翁”全链路排查机器人的使用方式、技术原理和能力边界。
- 已知背景：大富翁基于 Hermes Agent 构建；实习工作重点是使用大富翁辅助问题排查，不需要开发 Hermes Agent，不参与 Agent Framework 源码建设。因此本文不展开框架源码，而是站在“测开如何高质量使用 Agent 排障”的视角组织内容。
- 说明：Hermes Agent 属于团队内部能力时，具体实现、平台入口、权限模型、工具名称、日志字段和灰度策略应以内部文档和线上配置为准。本文将 Hermes Agent 抽象为一个企业级 Agent 编排层，用来解释其通用工作机制。

第一部分：必须掌握的基础知识

优先级	知识点	为什么要学	学到什么程度即可	可跳过内容
1	LLM	Agent 的“脑”。大富翁能理解自然语言问题、总结日志、归纳异常链路，底层依赖大模型的语言理解、推理和生成能力。	知道 LLM 输入/输出、上下文窗口、token、system prompt、temperature、幻觉、上下文污染、模型不保证事实正确。能解释为什么排障结论必须基于工具返回证据。	模型训练细节、Transformer 数学推导、微调算法、推理加速、显存优化。
2	Function Calling	让模型把“我要查订单日志”转成结构化函数参数，而不是只生成一句自然语言。它是 Agent 调工具的基础能力。	理解函数 schema、name、description、parameters、required、enum、返回值。能看懂一次模型为什么选择某个函数、参数是否合理。	模型侧 function-call 模板实现、解析器源码、不同模型服务的协议细节。
3	Tool Calling	企业排障不是聊天，必须调用日志平台、数据库、监控、工单、知识库。Tool Calling 是 Agent 接入外部系统的执行层。	能区分只读工具和写工具；知道工具要做鉴权、参数校验、超时、重试、幂等、脱敏、审计。能判断工具返回是否足够支持结论。	工具运行时框架源码、插件打包机制、复杂沙箱实现。除非后续要开发工具，否则了解即可。
4	Agent	Agent = LLM + 目标/角色 + 工具 + 记忆/上下文 + 规划/反思 + 执行循环。大富翁就是排障场景下的 Agent 应用。	理解一次 Agent Loop：接收问题 -> 识别意图 -> 补上下文 -> 选择工具 -> 观察结果 -> 继续查询或生成报告。能讲清楚 Agent 与普通 ChatBot 的区别。	Agent Framework 内部调度源码、复杂插件生命周期、多模型路由实现。
5	ReAct	ReAct 是理解排障 Agent 的关键范式：边推理边行动。模型先提出排查思路，再调用工具拿证据，再根据观察结果调整方向。	掌握 Thought/Action/Observation 的思想即可。能把一次排障拆成“假设 -> 查询 -> 观察 -> 修正假设 -> 结论”。	论文实验细节、基准数据集、prompt 复现代码。

优先级	知识点	为什么要学	学到什么程度即可	可跳过内容
6	MCP	MCP 是把外部系统以标准协议暴露给 AI 应用的一种方式。即使 Hermes Agent 未必直接使用 MCP，也能帮助理解工具接入边界。	知道 MCP 的核心是 Tools、Resources、Prompts；Tools 可让模型查询数据库、调用 API、执行计算；Resources 可暴露上下文资源。	MCP Server 开发、传输协议实现、OAuth 细节。除非团队工具链采用 MCP，否则不深入。
7	Multi-Agent	复杂排障可能拆成多个角色：日志分析、SQL 查询、监控分析、知识库检索、报告生成。但初期使用大富翁不需要自己设计多 Agent。	了解多 Agent 是多个专长 Agent 协作；知道优点是分工明确，风险是成本、延迟和一致性更难控。	多 Agent 调度算法、角色博弈、复杂协作框架源码。使用阶段先忽略。

第二部分：Hermes Agent 3 天速路线

Day 1: 建立 Agent 基础认知，能看懂一次问答

学什么：

- LLM、Function Calling、Tool Calling、Agent Loop 的基本概念。
- 大富翁入口、常见问题类型、一次对话的输入输出形式。
- 系统提示词、工具列表、工具返回证据与最终回答之间的关系。

看什么资料：

- 团队内部：大富翁使用手册、Hermes Agent 接入说明、常见排障案例。
- 公开资料：Qwen-Agent README/Features，了解 Agent = 模型 + 工具 + 规划 + 记忆。
- 公开资料：MCP Tools 文档，理解工具如何被模型调用外部系统。

产出什么笔记：

- 一页笔记：Agent、Function Calling、Tool Calling 的区别。
- 记录 3 个真实/模拟问题：用户怎么问、大富翁查了什么、最后怎么回答。
- 列出大富翁常见工具清单：日志、DB、监控、工单、知识库。

达到什么效果：达到效果

Day 2: 理解全链路排障流程，能分析工具调用是否合理

学什么：

- ReAct：假设、行动、观察、修正假设。
- 日志平台查询：traceId、requestId、userId、orderId、serviceName、errorCode、时间窗口。
- 数据库查询：只读查询、主键/业务键、状态字段、更新时间、数据一致性。
- 监控系统：接口成功率、耗时 P95/P99、错误率、QPS、告警、依赖服务状态。
- 工单系统与知识库：历史相似问题、已知故障、SOP、发布记录。

看什么资料：

- 团队内部：日志平台字段说明、监控平台指标说明、DB 查询规范、线上只读权限规范。
- 历史故障复盘：挑 2 个故障，看大富翁是否能复现排查路径。
- ReAct 论文摘要或博客：重点看 reasoning 与 acting 交替思想。

产出什么笔记：

- 一张流程图：用户问题 -> 意图识别 -> 工具选择 -> 多轮查询 -> 报告生成。
- 一份排障字段表：不同问题需要哪些关键字段，例如 traceId、订单号、服务名、时间范围。
- 一份工具观察表：工具调用是否必要、参数是否准确、结果是否足够支撑结论。

达到什么效果：达到效果

Day 3: 形成技术沉淀，能讲清架构和边界

学什么：

- Hermes Agent 在大富翁中的位置：入口层、编排层、模型层、工具层、数据源层、报告层。
- 能力边界：模型幻觉、权限限制、数据延迟、工具失败、知识库过期、跨系统字段不一致。
- 测开使用准则：提问模板、证据校验、报告复核、问题归因、反馈闭环。

看什么资料：

- 团队内部：Hermes Agent 平台架构图、工具注册文档、权限/审计规范、Prompt 变更流程。
- 公开资料：MCP Introduction、Qwen-Agent、AgentScope，用于补充通用概念。
- 历史 Case：选 1 个典型故障，写成标准排查报告。

产出什么笔记：

- 技术沉淀文档初稿：基础知识、3 天学习路线、大富翁流程、测开关注点、典型 Case。
- 一张架构图：从用户问题到报告输出。
- 一页 Mentor 汇报提纲：我理解的架构、我会怎么用、我知道哪些边界。

达到什么效果：达到效果

第三部分：大富翁机器人分析框架

假设大富翁是一个故障排查机器人，它的核心是把自然语言问题转成可执行的排查计划，并基于多数据源证据生成报告。

用户输入后 Agent 如何思考

- 识别问题类型。
- 抽取环境、时间范围、traceld、订单号、服务名、错误码等关键字段。
- 判断信息是否足够，不足时追问。
- 形成排查假设，并在工具观察后修正。

各数据源如何查询

数据源	常见查询输入	Agent 关注点	测开复核点
日志平台	traceld、requestId、orderId、服务名、错误码、时间范围	异常堆栈、调用链、上下游错误、耗时	时间范围、样本数量、是否误判日志级别
数据库	业务主键、用户 ID、订单号、状态字段	业务状态、幂等记录、数据一致性	只读权限、SQL 条件、环境区分
监控系统	服务名、接口名、依赖名、指标、时间窗口	错误率、耗时、QPS、告警	基线对比、发布/流量变化、指标粒度
工单系统	错误码、服务名、业务关键词	历史相似问题、处理状态	历史结论是否仍适用
知识库	业务名、错误码、SOP、场景关键词	排障步骤、修复建议	版本是否过期、是否有引用来源

完整流程图 Mermaid 源码

```
flowchart TD
    A[用户输入故障问题] --> B[Hermes Agent 接收请求]
    B --> C[解析意图与关键信息]
    C --> D{信息是否足够}
    D -- 不足 --> E[追问: 时间范围/traceld/订单号/服务名/环境]
    E --> C
    D -- 足够 --> F[制定排查计划]
    F --> G[选择工具]
    G --> H[日志平台工具]
    G --> I[数据库只读查询工具]
    G --> J[监控系统工具]
    G --> K[工单系统工具]
    G --> L[知识库/RAG 工具]
    H --> M[观察日志异常、调用链、错误码]
    I --> N[观察业务状态、数据一致性]
    J --> O[观察指标、告警、依赖服务]
    K --> P[观察历史工单、处理进展]
    L --> Q[观察 SOP、历史复盘、已知问题]
    M --> R[汇总证据]
    N --> R
    O --> R
    P --> R
    Q --> R
    R --> S{证据是否支持结论}
    S -- 否 --> F
    S -- 是 --> T[生成排查报告]
    T --> U[输出: 现象/影响/证据/可能原因/建议动作/下一步]
```

第四部分：测开需要重点关注什么

重要程度	内容	原因
★★★★★	Agent 使用与排障链路	这是你的核心工作。要能把用户问题拆成排查目标，理解 Agent 为什么查这些系统，判断报告是否有证据。
★★★★★	Function/Tool Calling	大富翁的有效性取决于工具调用。测开要重点看工具选择、参数、返回结果、异常和权限边界。
★★★★★	日志、数据库、监控、工单、知识库的数据含义	排障结论来自这些数据源。不了解字段和指标，就无法判断 Agent 输出是否可信。
★★★★★	ReAct 排障思维	排障天然“提出假设 -> 查证 -> 修正”的循环。掌握 ReAct 能帮助你复盘 Agent 的推理路径。
★★★★★	证据链与报告校验	测开不能只复制机器人结论，需要校验时间范围、样本、日志、指标和业务状态是否一致。
★★★	LLM 基础	需要理解上下文、幻觉和 prompt，但不需要研究模型训练。
★★★	MCP 概念	有助于理解工具标准化接入，但未确认 Hermes Agent 是否使用 MCP，先了解 Tools/Resources/Prompts 即可。
★★★	Multi-Agent	复杂排障可能用到多 Agent 分工，但作为使用者先知道概念，不需要设计协作系统。
★★★	Agent 评测	建议了解用例集、回放、人工打分、准确率/召回率，但短期不用搭评测平台。
★	Agent Framework 源码	当前岗位目标是不开发 Hermes Agent。源码投入产出低，除非 Mentor 明确安排。
★	模型训练/微调/推理优化	与日常使用大富翁排障关系较远，几天学习时间内不值得深入。
★	复杂多 Agent 调度算法	会增加认知负担，且对使用大富翁帮助有限。

第五部分：技术沉淀学习文档

我的理解：大富翁是什么

- 大富翁是面向故障排查场景的 Agent 应用，不是单纯问答机器人。
- Hermes Agent 承担编排角色：理解用户问题、维护上下文、选择工具、执行多轮查询、组织报告。
- 其价值是把日志、DB、监控、工单和知识库串成一条排查链路。

使用大富翁的提问模板

- 请帮我排查【环境】中【服务/接口】在【时间范围】出现【错误码/现象】的问题，traceId 是【xxx】。
- 订单【orderId】在【时间】发生【业务状态异常】，请按日志、DB、监控、历史工单给出证据链。
- 请判断【服务名】在【时间范围】的错误率升高是否与【发布/依赖服务】有关，并输出排查报告。

使用边界与风险

- 模型可能幻觉，没有工具证据的结论只能当假设。
- 工具可能查错范围，时间窗口、环境、服务名、业务 ID 错误会导致错误归因。
- 知识库可能过期，历史 SOP 必须结合当前版本和实际证据复核。
- 不要让 Agent 自动做高风险修复，补偿、重试、改数据、回滚应走人工确认。

Mentor 汇报提纲

- 我理解的大富翁：基于 Hermes Agent 的全链路排障助手，核心是模型编排工具。
- 工作机制：识别意图 -> 补齐上下文 -> 制定计划 -> 调工具 -> 汇总证据 -> 生成报告。
- 测开使用重点：提供准确上下文、检查工具调用、验证证据链、识别能力边界、沉淀典型 Case。

参考资料

资料	链接	用途
Qwen-Agent GitHub	https://github.com/QwenLM/Qwen-Agent	Qwen-Agent 是基于 Qwen 的指令遵循、工具使用、规划和记忆能力构建的 Agent 框架。
Qwen-Agent Features	https://qwenlm.github.io/Qwen-Agent/en/guide/get_started/features/	介绍统一 Agent 接口、多步/多轮工具调用、工具调用解析等能力。
Qwen Function Calling	https://qwen.readthedocs.io/en/latest/framework/function_call.html	介绍 Qwen3 如何支持 function calling。
MCP Introduction	https://modelcontextprotocol.io/docs/getting-started/intro	MCP 是连接 AI 应用与外部系统的开放标准。
MCP Tools	https://modelcontextprotocol.io/specification/2025-06-18/server/tools	MCP Tools 允许模型与外部系统交互，如查询数据库、调用 API、执行计算。
ReAct Paper	https://arxiv.org/abs/2210.03629	ReAct 将推理轨迹与任务行动交替结合，让模型与知识库或环境交互获取信息。
AgentScope GitHub	https://github.com/agentscope-ai/agentscope	AgentScope 是生产取向的 Agent 框架，可作为企业级 Agent 能力参考。