

作品集

吴佳睿

东南大学 · 本硕

[jrwdev.site](#) 个人主页 ↗

邮箱 19913823585@163.com

微信 wjr19913823585

手机 19913823585

QQ 471464213



教育经历 东南大学 · 硕士 · 2024.09—2027.06 东南大学 · 本科 · 2020.09—2024.06

华为技术有限公司

2026.05—2026.08

开发者技术支持场景 · 实习实践

参与知识生产与管理平台建设，将工单处理、知识生成、质量评估与审核反馈串成可追踪的闭环。

Agent 工作流

判定 → 生成 → 评估 → 修复 → 保存，拆分阶段并控制上下文。

质量评估

异构检查并行执行，工具、语义与规则路径相互隔离。

Skill 自演进

Badcase → Proposal → Diff → 版本化，支持回滚和人工确认。

可靠性设计

超时、重试、降级与 Fail-Closed 共同守住交付质量。

约 60%

Token 消耗下降

10—25s

单条评估耗时区间

约 8%

触发实例切换

55%→70%

整体审核通过率变化

EventGuard

01

订单事件溯源与异常治理

Java

Kafka CDC

Vue

从不可变事件、异步读模型到告警与人工补偿，覆盖发现、定位、处置。

[eventguard.jrwdev.site](#) ↗

Synapse

02

可观测 Code Agent Harness

Python

MCP

Git

治理上下文、记忆、工具执行与结果验证，让长链路任务可以恢复和评测。

[jrwdev.site/synapse](#) ↗

个人网站工程化

静态页面与本地数据驱动，统一设计 token、响应式布局、可访问链接与站点发布流程。

HTML · CSS · JavaScript
Git · Nginx · HTTPS

华为技术有限公司 · 开发者技术支持场景

Java 业务编排 + Python Agent 工程

平台服务于开发者技术支持场景：把分散在工单、规范和审核意见里的信息，整理成可检索、可评估、可追踪的知识资产。我的工作集中在流程编排、质量控制、Skill 版本化与运行可靠性。

平台闭环

每一步都有输入、结果与可回溯状态



01 · Agent 阶段拆分

把单阶段硬编排改为“判定—生成—评估—修复—保存”的多阶段流程。判定不通过时提前结束，减少无效上下文和重复生成。

约降低 60% Token 消耗

02 · 三路径质量评估

将 LLM Dimensions、Programmatic 与 Hybrid 三条路径并行化；程序化规则小于 10ms，任一路径失败都能独立降级。

18—45s → 10—25s

03 · Skill 自演进

审核意见聚合成 Proposal，经人工 Diff 后以 Copy-on-Write 生成新版本；每次变更可比较、可回滚、可继续观察。

Badcase → Proposal → Version

04 · 容错与边界

为超时、限额和异常结果设置重试、LLM fallback 与 Sentinel Finding。高风险质检采用 Fail-Closed，避免“无结果”被误判为通过。

切换后最终成功约 95%

工程取舍

让自动化有速度，也有可解释的边界

先判定，再生成 把昂贵的生成动作放到有效工单之后，节省上下文预算。	并行，但不互相拖累 检查路径独立超时和降级，单点异常不阻塞整条流水线。	版本化，而不是覆盖 审核意见生成候选版本，人工确认后再逐步放量。
---	---	--

质量评估矩阵

不同检查负责不同类型的确定性

LLM Dimensions 语义与完整性 检查回答是否覆盖问题、证据与关键结论，适合处理开放文本。	Programmatic 结构与硬规则 字段、链接、格式和代码规则快速校验，程序化路径小于 10ms。	Hybrid 高风险组合 规则先筛选，再交给模型解释；异常使用 Sentinel Finding 并保持 Fail-Closed。
--	---	---

运行结果

以公开安全视图呈现抽样区间，指标用于说明设计取舍而非承诺固定产出。

120—180 / 日

日均工单处理量区间

约 60%

评估通过后可跳过修复

让每一次状态变化都能被追溯和处置

从写入、投影到告警，围绕“发生了什么”建立证据链。

传统订单系统常只保留最终状态，问题发生后很难回答“哪一次变更造成了现在的结果”。EventGuard 以不可变领域事件为事实源，异步构建读模型，并把异常检测、告警分析与人工补偿接到同一条操作路径上。

Java 17

Spring Boot 3

PostgreSQL

Kafka

Vue 3

在线体验

体验指南



核心链路 · 写入与治理分离，故障有明确落点

Command
幂等指纹 / 版本校验

Event Store
不可变事件 + 快照

CDC
Debezium → Kafka

Read Model
去重 / 乱序守卫

Detect & Act
告警 / 查询 / 补偿

事件事实源

用 commandId、请求指纹与唯一约束处理重复提交；每 100 个事件生成快照，重建时只需回放增量。

重复提交不新增事件

异步读模型

消费去重表、版本守卫、提交通知与兜底轮询共同处理重复、乱序和短暂延迟，读写解耦但状态可追平。

527 次抽样均在 2s 内追平

CDC 与毒消息隔离

PostgreSQL WAL 通过 Debezium 进入 Kafka，消费端手动提交、三次重试并将无法处理的消息送入死信队列。

数据库或 Kafka 暂停后约 2s 恢复

异常到处置闭环

规则与模型分层检测，查询采用固定只读模板；高风险动作进入人工审批，支付重试和库存失败可回到补偿入口。

8 个查询场景验证通过

338
端到端测试通过

70.78
组 / 秒 · 50 写并发

925ms
P95 延迟

0
已提交事件丢失

幂等验证
重复 command、重试与并发冲突均有覆盖。

故障演练
数据库宕机、Kafka 暂停和毒消息路径可复现。

可观测
Prometheus、Grafana、Loki 贯穿写入、积压与补偿。

验证范围包含功能、负载和故障演练；指标来自本地可复现实验与抽样验收。

Synapse

可观测、可扩展、可评测的 Code Agent Harness

让长链路任务 有上下文，也有边界

把模型、规划器和真实开发环境之间的运行时控制层做清楚。

Synapse 面向代码仓库任务，统一治理上下文、分层记忆、工具执行、任务恢复与结果验证。设计重点不是“让模型多说一点”，而是让每一步都能被授权、被恢复、被评测。

[Python](#) [FastAPI](#) [MCP](#) [Git](#) [SWE-bench](#)[网页版 ↗](#)[命令行 ↗](#)[体验指南 ↗](#)

运行时结构 · 组件可替换，状态可恢复，结果可验证

Provider

3 家模型服务

Planner

4 类规划策略

Context

检索 / 预算 / 压缩

Tools

action-time 鉴权

Grader

gold patch / gate

上下文与记忆治理

Git-aware 检索、AST 符号提取、相关性排序和四区预算共同控制上下文；Session、Project、User、Semantic 四层记忆支持跨轮复用。

检索故障最多阻塞 10s

任务恢复与执行韧性

打转检测、逐工具超时、token 硬预算和中断配对让长任务可以 resume；同文件反复修改时回滚快照，不触碰用户工作区。

180 次评测无渠道误判

Action-time 安全

授权发生在动作执行前，结合工作区、命令链、重定向与 SSRF 风险判定；高风险操作拦截或请求确认，并留下签名审计。

5 级决策 · 14 类危险命令

冻结评测与外部判分

以隔离 workspace、gold patch、completion gate 和不可变登记判分，避免只看模型自报成功；评测结果由外部 grader 复核。

20 题冻结集 · success@3 约 40%—55%

3

Provider 接入

4

Planner 类型

20

冻结评测题目

40—55%

success@3 区间

可扩展

新增 Provider 约 40 行，主循环保持不变。

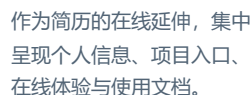
可恢复

跨进程 resume，补齐中断调用并回滚快照。

可判分

隔离 workspace + gold patch + 外部 grader。

评测页面、命令行工具和使用说明均可在个人网站直接打开，截图索引见下一页。



EventGuard 控制台的六个主要入口

订单中心	状态与时间线	异常看板	实时监控与根因	AI 查询	自然语言检索
运营周报	指标与趋势汇总	补偿执行	失败处置闭环	审批队列	高风险动作确认

通过多轮交互与规划执行复杂任务，并汇总测试结果、耗时和工具调用。

保存会话历史与检查点，支持选择既有任务并从中断位置继续。

以可视化界面呈现会话、工作区、工具调用和模型推理过程。

从安全、过程、质量和效率四个维度展示任务评价与改进建议。

