

運用大型語言模型之軟體錯誤回報自動重現應用

Automatically Reproducing Bug Reports based on LLM

指導教授：李信杰

專題成員：李沛宸

專題背景與目的

- Bug Report 資訊經常不完整**：研究分析 180 個真實 Bug Reports，發現重現步驟與環境資訊經常缺失，增加 Bug 重現難度
 - Bug重現是修復必要前置步驟**：錯誤重現是錯誤定位與修復的基礎，但人工重現流程耗時且高度依賴開發者經驗
 - LLM 具備強大理解與推理能力**：近年研究開始探討利用大型語言模型自動理解 Bug Report 並執行錯誤重現流程
- 建構 AI Agent 驅動之 Bug 自動重現系統，使 AI 能根據 Bug Report 自動理解、規劃並執行操作流程，降低人工重現成本並提升軟體測試效率

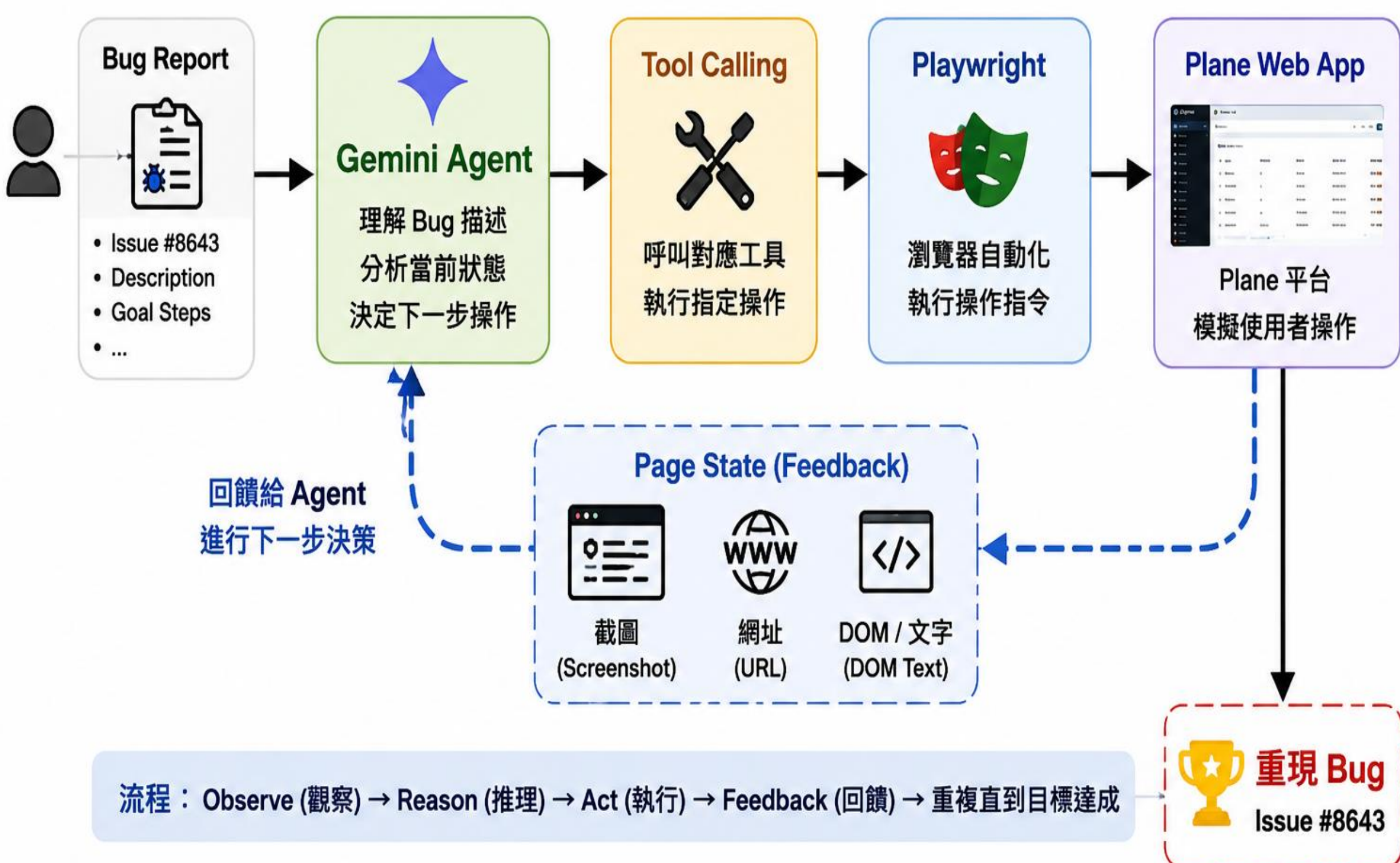
專題介紹

本研究結合大型語言模型（Gemini CLI）與 Playwright 瀏覽器自動化技術，建構一套 AI Agent 驅動之 Bug 自動重現系統。系統透過 Tool-based Agent Architecture 將系統操作封裝為可呼叫工具，並結合頁面狀態擷取機制，讓 Agent 能根據當前介面資訊動態規劃與執行操作流程。執行過程中，AI Agent 持續觀察頁面狀態、分析操作結果並決定下一步行動，形成 Observe - Reason - Act 的決策循環。本研究以 Plane 平台真實案例 Issue #8643 作為驗證對象，成功重現工作項於特定分組條件下消失與重新出現之錯誤現象，驗證 AI Agent 應用於 Bug Reproduction 與軟體測試自動化之可行性

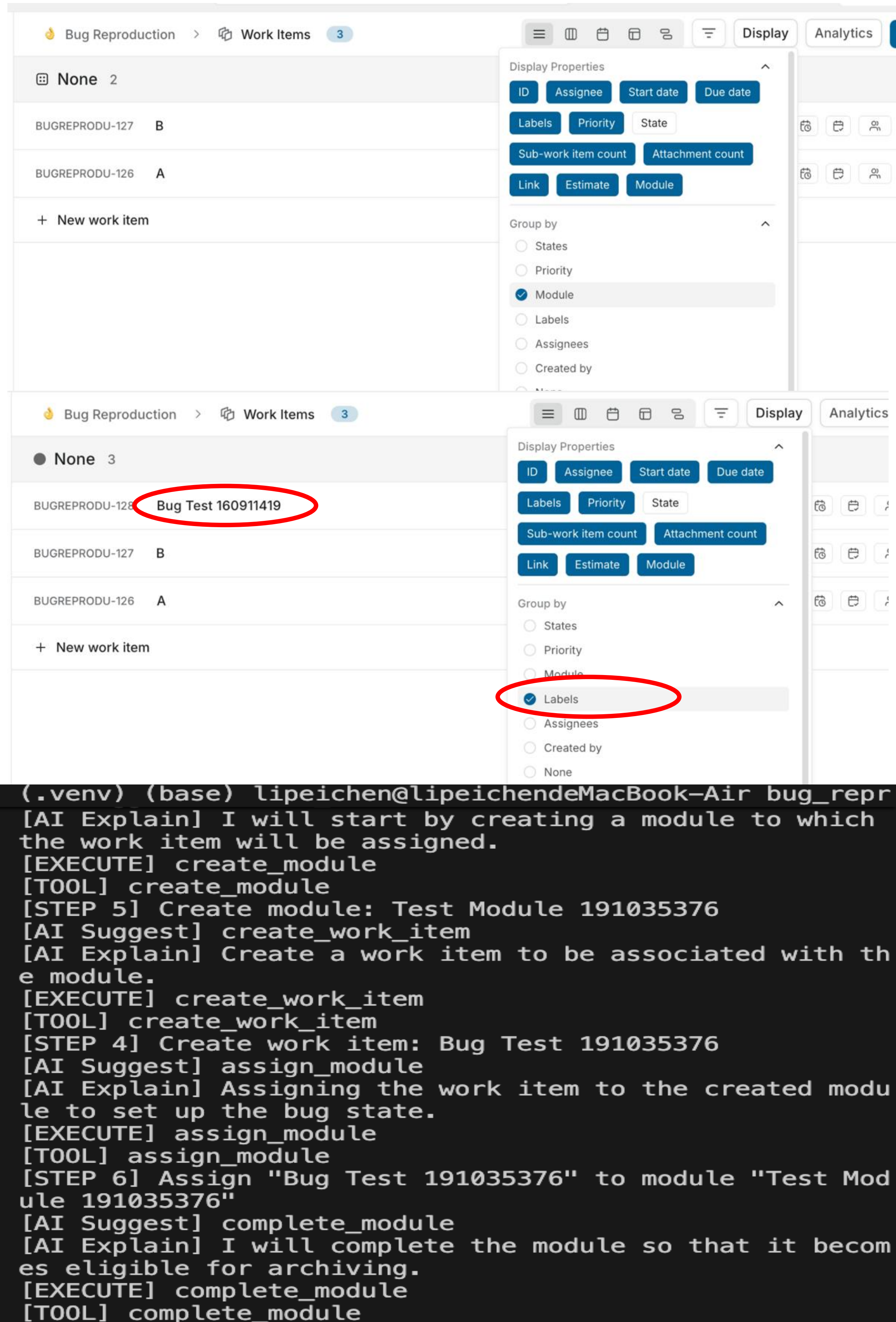
核心技術

- 動態決策式 Agent Loop**：採用 Observe - Reason - Act 架構，根據頁面狀態動態規劃下一步操作，而非固定腳本執行。
- Tool-based Action Planning**：將系統操作封裝為高階工具，透過 Tool Calling 完成自然語言到操作流程的轉換。
- 閉環式狀態回饋機制**：持續蒐集 Screenshot、URL 與 DOM 資訊作為決策依據，提升重現流程穩定性

系統架構圖



成果展示



專題亮點

- 真實 Web Application 驗證**：以真實 SaaS平台 Plane 實際作為驗證案例，非模擬情境或簡化測試案例，驗證 AI Agent 於真實軟體環境中的可行性
- 整合 LLM 與 Browser Automation**：結合 Gemini、Tool Calling 與 Playwright，建立從 Bug Report 到實際操作的完整流程，實現自然語言到系統操作的轉換
- 複合型 Workflow Bug 重現**：涉及 Module、Work Item、Archive 等多項功能，需經過跨頁面、多步驟操作才能觸發，不同於單一步驟 UI 測試案例

結論與未來展望

研究成功整合 Gemini、Tool Calling 與 Playwright，建構 AI Agent 驅動之軟體錯誤回報自動重現系統。未來或許能進一步支援 GitHub Issues、Jira 與 Bugzilla 等不同錯誤回報平台，並強化 Agent 在自主決策與錯誤恢復能力，提升面對複雜真實情境時的錯誤重現效率與穩定性