

章節	學習成效承諾	具體內容 v1.0	進度	內容上線時程	單元教學影片	圖文教學實戰	實作練習題	Github 專案& Prompt 實戰教材	共學與線上 FAQ	【加碼工具！即將推出！】網頁 AI 互動學習模組
Ch1-規格驅動開發的前提	理解課程架構與規格光譜	理解各門課程的架構與學習路徑。掌握規格光譜的核心概念(從 Vision/PRD → Markdown → DSL Feature → ISA Feature → Test Code 的完整翻譯鏈)。理解為何需要整合軟工方法論(TDD, BDD, DDD)來達成 60% →100% 自動化開發的漸進式路徑。	100%	已完成	v		v			
Ch1-規格驅動開發的前提	認識真正的規格驅動	理解真正的規格驅動定義：不是「把規格寫清楚」而是「建立完整的翻譯鏈」。從 Problem Domain 精準翻譯到 Solution Domain，並由 TDD、BDD、DOD 說明為什麼會到真正的規格驅動？因為它們仰賴 LLM 去「猜」需求或問題，而 LLM 是愚昧且妄想的。真正的規格驅動需要透過模型(Event Storming, Gherkin, ISA)確保資訊完整、用測試驗證正確的正確性(Auto-Feedback Loop)。	100%	已完成	v		v			
Ch1-規格驅動開發的前提	識別現有工具極限	識別現有工具極限的瓶頸。理解為何在 60% 自動化、問答核心在於(1)沒有結構化的規格模型、仰賴自然語言的模糊翻譯。(2)缺少精確的翻譯機制。LLM 需要「猜測」缺少資訊。(3)沒有 Auto-Feedback Loop 來驗證正確性。這些限制迫使 AI 无法持續、穩定地完成開發任務。需要人工介入修正。	100%	已完成	v		v			
Ch1-規格驅動開發的前提	破除規格清楚的迷思	理解為何「把規格寫清楚」是錯誤的。真正的重點是對模型。軟體危機的歷史教戰：1960-1970 年代，機器語言太複雜，沒辦法「讀」出來。解決方案：對模型的描述(KPI)，對 PM 的描述(PRDO)，對 QA 的描述(測試案例)，對工程師的描述(程式碼)。抽象讓我們協作。但抽象會損失資訊。因為需要「翻譯」來確認。	100%	已完成	v		v			
Ch1-規格驅動開發的前提	掌握 Discovery → Formulation 流程	掌握不同模態下規格的分析技巧。Discovery(探索)階段，透過 Event Storming 等技術發現問題領域的核心概念。Formulation(形式化)階段，運用規格化轉換化規格化規格(Gherkin Feature File)。這個流程會從來源資料到技術實現的規格轉換。避免「規格不一致」導致的錯誤與溝通誤工。	100%	已完成	v	v	v			
Ch2-規格光譜	理解抽象與翻譯的雙核心概念	深入理解軟工流程的大支柱：(1)抽象(Abstraction)：沒有抽象就無法協作。機器語言太複雜，企業無法運作，透過抽象，老闆能看懂 KPI, PM 能看懂測試結果，工程師能看懂程式碼。每個領域都能自己熟悉的語言能夠清晰地傳遞。(2)翻譯(Translation)：有了抽象就要簡單地翻譯到全自動化開發。每一層抽象都會擴充資訊(如支援庫包)，沒說紅字才能知道。是否只說規格？應該能體會能夠夠使資訊「翻譯」回來。規格驅動開發的建構：建立一條完整的翻譯鏈。從 Problem Domain 精準翻譯到 Solution Domain，不遺漏任何資訊。	100%	已完成						2025/11/4
Ch2-規格光譜	認識領域驅動設計(DDD)的核心概念	理解 DDD 的四大領域分：(1)Problem Domain(問題領域)：關注「要解決什麼問題」。參與者包含企管、業務、PM、QA、使用者、特性和範例等。DOD 有三個子領域：(2)Solution Domain(解法領域)：關注「如何解決問題」。SD 有三個子領域：Domain Model(問題領域的模型)、Domain API(問題領域的API)、Domain Rule(問題領域的規則)。(3)領域邏輯層(工作)：一機器人、終端裝置、終端應用。重要理解：Event Storming 產出的領域模型是如何映射到 Gherkin 的 Scenario。Gherkin 如何映射到 ISA(如後端需要 API Spec + Entity Spec)。ISA 如何產生測試用例。Gherkin 與操作程式碼。	100%	已完成			v			2025/11/4
Ch2-規格光譜	理解事件風暴(Event Storming)模型所呈現的系統本質樣貌	掌握 Event Storming 如何呈現系統的運作本質。系統行為本質：Command(指令)作為 Input, Event(事件)作為 Output, Command Handling(指令處理)作為事件行為。透過 Event Storming Workshop 團隊協作辨出：Domain Event(領域事件), Command(觸發事件的指令), Aggregate(聚合根, 處理指派的責任), Policy(政策, 業務邏輯的行為反應), Read Mode(查詢模式)。這種模態化方式讓業務邏輯輕量化、可重用，並能直接映射到 Gherkin 的 Given-When-Then 語句。	100%	已完成		v	v			2025/11/4
Ch2-規格光譜	認識規格光譜五層結構及其映射關係	理解規格光譜的完整結構與每一層的映射關係:Vision/PRD 最抽象，給老闆看(PM 看)→Markdown 規格(結構化的文字)→Domain Model(領域模型)→Domain API(領域API)→Domain Rule(領域規則)→Domain Logic(Instruction Set Architecture)最精確的可執行規範。使用 API(Protocol)→測試程式碼(可執行的映射邏輯編程)→程式碼(運算邏輯工作)→機器人、終端裝置、終端應用。重要理解：Event Storming 產出的領域模型是如何映射到 Gherkin 的 Scenario。Gherkin 如何映射到 ISA(如後端需要 API Spec + Entity Spec)。ISA 如何產生測試用例。Gherkin 與操作程式碼。	100%	已完成		v	v			2025/11/4
Ch2-規格光譜	理解從 DSL 翻譯到 ISA 的意義與世界趨勢	理解軟體語言演進的歷史趨勢與 DSL→ISA 翻譯的意義。軟體語言演進：(1)過程導向時代：機器語言→組合語言→高階語言(C/Java/Python)。工程需要明確告訴機器「怎麼做」。(2)結果導向時代：DSL(Compiler, 編譯器)→ISA(Assembler, ISA 裝載)→Codegen(直接生成)。工程師只需描述「要什麼」。Systematica(聚合根, 處理指派的責任), Policy(政策, 業務邏輯的行為反應), Read Mode(查詢模式)。這種模態化方式讓業務邏輯輕量化、可重用，並能直接映射到 Gherkin 的 Given-When-Then 語句。	100%	已完成		v	v			2025/11/4
Ch3-TDD 實戰(70%)	掌握 TDD 基本流程與單元測試	以實作案例(水冷壓縮機課程平臺)會到 TDD 的基本流程: Red-Green-Refactor cycle。(1)紅(Red)：先寫 fixture, test, 定義預期行為。(2)綠燈(Green)：寫出程式碼讓測試通過。(3)重構(Red): 在測試保持不改改善品質。重視學習單元測試的撰寫。從 Service Layer 的業務邏輯測試開始，理解測試的最大價值。作為規格(Specification)、作為品質保護(Protection)、作為設計工具(Design Tool)。掌握從零開始建立測試基础设施(test runner, assertion library)。	95%	2025/11/3-2025/11/18 影片 2025/11/25	v	v	v	v		2025/11/4
Ch3-TDD 實戰(70%)	理解 TDD 與 Gherkin 思維的結合	學習如何用 Given-When-Then 構造來思考撰寫單元測試。取代傳統的 A(Arrange-Act-Assert)模式。Given(前置條件)：系統的初始狀態。When(執行動作)：測試要測驗的行為。Then(後置條件)：驗證結果。驗證結果：測試的結果是否符合預期。不同條件下的測試規劃：如何針對不同的 context(如嵌入 vs 嵌入、有權限 vs 無權限)設立不同的測試。這裡說明訓練後續的 BDD(Behavior-Driven Development)建立基礎。	95%	2025/11/3-2025/11/18	v	v	v			2025/11/4
Ch3-TDD 實戰(70%)	掌握 AI 幫助 TDD 的三階段實戰	學會如何讓 AI 協助執行完整的 TDD 工程。(1)驗證階段: 提供專門的 prompt 讓 AI 提供 testing test。根據規則是前或後置條件使用不同的 prompt(因為驗證與後置的驗證重點不同)。(2)驗證階段: 下一個 prompt AI 將繼續工作，直到所有測試都通過為止。體驗測試驅動的優點：自動開發測試。(3)重構階段: 提供專門的 prompt 要求 AI 在回圈測試進行重構。確保程式品質提升但行不通。重點強調：將大任務拆成小任務(紅→綠→重構)的效益，以及如何透過 prompt engineering 控制 AI 的行為。	80%	2025/11/3-2025/11/18	v	v	v			2025/11/4
Ch3-TDD 實戰(70%)	掌握從單元測試到端到端測試	學習部署測試的技術與模式。(1)Test Double 概念: 理解為何需要測試替身(驗證依賴, 接口測試速度, 模擬難以重複的場景)。(2)Mock(模擬物件): 驗證依賴行為。Mock 透過預設的 prompt 請 AI 理解如何設定 Mock, 如何模擬互聯網連線: 測試資料要如何介面 API, 物料庫, 第三方服務的連接。	15%	2025/11/3-2025/11/20	v	v	v			2025/11/4
Ch3-TDD 實戰(70%)	掌握進階測試技巧	培養基於測試的技術與模式。(1)Test Double 概念: 理解為何需要測試替身(驗證依賴, 接口測試速度, 模擬難以重複的場景)。(2)Mock(模擬物件): 驗證依賴行為。(3)Stub(替身): 提供預設回應。讓測試聚焦在測試對象而非依賴。(4)Spy(間諜物件): 記錄動態資訊供後續檢證。(5)建立 API 標準並寫 Mock。透過預設的 prompt 請 AI 理解如何設定 Mock, 如何模擬互聯網連線: 測試資料要如何介面 API, 物料庫, 第三方服務的連接。	0%	2025/11/10-2025/11/23	v	v	v			2025/11/11
Ch3-TDD 實戰(70%)	提升測試判斷力能夠挑刷 AI 的測試程式碼	培養基於測試的技術與模式。(1)針對不對的培訓：不對的測試是完全、正確、有價值。學習如何：(1)針對不對的培訓：不同測試的結果是否完全、正確、有價值。測試資料要對測試對象而非依賴。(2)理清不同驗證點的差異(狀態正確 vs 行為正確, 程序正確 vs 約束正確)。(3)識別別扭的培訓：(過於精緻作業)。測試不穩定，測試結果不準。(4)評估複雜度的意義：不是程式複雜度，而是測試資料複雜度。核心目標：你能挑刷的測試程式碼，就代表你具備了設計自动化的能力。	95%	2025/11/3-2025/11/18	v	v	v			2025/11/11
Ch4-BDD 實戰(80%)	理解測試驅動開發規格的問題與 Gherkin 的必要性	理解測試驅動開發規格的基本問題：(1)本末倒置：一個簡單的業務邏輯可能需要十行測試程式碼。充份掌握技術術語(如 setup, teardown, 模块設定)。(2)難搞：非技術人員(PM, QA, 領域專家)無法理解測試在驗證什麼業務行為。(3)數改：業務邏輯變更，需要改動測試程式碼。維護成本高。理解為何需要 Gherkin 來做關注點切換：將「業務邏輯」與「技術操作」(How)分離。業務邏輯自然語言描述邏輯，技術層用 Step Definition 描述。DOD 有三個子領域：Domain Model(問題領域的模型)、Domain API(問題領域的API)、Domain Rule(問題領域的規則)。	30%	2025/11/15-2025/11/30	v	v	v			2025/11/11
Ch4-BDD 實戰(80%)	掌握 BDD 的 Test-First Workflow 與自動化開發	瞭解 BDD 要求的測試驅動程式碼寫到 DSL level 可執行的規格能力。BDD 的 Test-First Workflow。(1)先寫 Gherkin Feature File(當前業務行為)。(2)執行測試(Red Step Definition 尚未完成)。(3)建 AI 實作 Step Definition 與業務邏輯(Green 测试通过)。(4)重構與優化。透過兩兩 prompt 完成 BDD 的基本開發。Prompt 1 AI 請提供 Feature File 並生成 Step Definition。Prompt 2 請 AI 實作業務邏輯。感受結果導向的全自動化開發(你只需要寫 Gherkin, AI 自動產生工作)。放棄雙工，專注在規格丈量而無需程式撰寫。	30%	2025/11/15-2025/11/30	v	v	v			2025/11/11
Ch4-BDD 實戰(80%)	能夠 Review AI 產生的測試程式碼	運用第三章學到的測試評估能力，檢視 AI 在 BDD 流程中产出的 Step Definition 與測試程式碼品質，評估重點：(1)Step Definition 是否正確實行 Gherkin 的語彙。(2)測試資料的準確性是否完整(Given 步驟)。(3)驗證邏輯是否準確(Then 步驟)。(4)是否有不必要的複雜或重複。(5)錯誤處理是否恰當。學會如何給 AI 回饋，讓 AI 修正問題。(如若 Step Definition 驗證要驗證 HTTP status code 是 201，(Given 步驟缺少初始化資源狀態))。目標：找出並指出是哪個步驟有問題。	30%	2025/11/15-2025/11/30	v	v	v			2025/11/11
Ch4-BDD 實戰(80%)	掌握前端與後端的 BDD 落地實戰(結合 AI)	學習如何用 Gherkin 語言來寫 BDD，並結合 AI 自動化。(1)後端 BDD + AI: 請寫 API 層級的 Gherkin Feature File，並用 API 請求的 POST /api/users 請求，with body ->。讓 AI 產生 API 測試與 Controller/Service 實作。(2)前端 BDD + AI: 請寫 UI 層級的 Gherkin Feature File，用使用者行為的語彙(如 When 用户點擊登入按鈕)，讓 AI 生產 E2E 請求(Playwright/Cypress)與元件實作。掌握不同層級 BDD 的 Step Definition，撰寫模式，以如何設計 prompt 讓 AI 了解前端與後端的技術差異。	0%	2025/11/15-2025/12/20	v	v	v			2025/11/11
Ch4-BDD 實戰(80%)	掌握如何覆寫能產出 99% 正確率測試程式的 ISA feature file	理解 ISA level 與 DSL level 的核心差異。DSL-Level(Domain-Specific Language)：目標讀者是 PO, BA, 業務人員。語言風格是自然語言。業務差異(如當玩家 Alice 加入遊戲 G001)。細節程度是高階業務意涵。ISA-Level(Instruction Set Architecture)：目標讀者是開發者、測試工程師。語言風格是技術術語、API 叫喚(如 When 玩家加入遊戲, call table : gameid playerName)。細節程度是低階技術指標。感受效果寫完 ISA level feature file 後如何達到 99% 正確率。因為 ISA 消除了歧義。AI 不需要猜測而是「執行指令」。	0%	2025/11/20-2025/12/10	v	v	v			2025/11/11

章節	學習成效承諾	具體內容 v1.0	進度	內容上線時程	單元教學影片	圖文教學實戰	實作練習題	Github 專案& Prompt 實戰教材	共學與線上 FAQ	【加碼工具！即將推出！】網頁 AI 互動學習模組
Ch5-ISA可執行規格 (90%)	透過原子化思維將 AI 產出的測試式碼精準度調高至 99%	學習掌握原子化設計原則與實踐。原子化(Atomic)意義：每個測試步驟只做一件事、驗證一個概念，避免複合邏輯導致的誤義。如何原子化：(1) 將複雜的業務邏輯拆解成多個原步驟。(2) 每個 Step Definition 只叫一個 API，並賦予唯一名稱。(3) 線性步驟分離重點(如分別點擊「Create」step 和「Read」step)。學習如何在 ISA Feature File 中增加約束或拆分步驟... → 重新生成... → 驗證，這個過程訓練你的「規格設計能力」。	0%	2025/11/20-2025/12/10	v	v	v		2025/11/18	
Ch5-ISA可執行規格 (90%)	用 ISA-Feature File 來產出高精度的後端 E2E Test	掌握語寫後端 ISA-Level Feature File 的技術，以技術接 API(CL) 的方式描述前端行為。例如 Given 步驟用頁面資料存取(Given 資料庫存在, with table: name email password), When 步驟用 API 呼叫(When 發送 POST /api/users, call table: name email password), Then 步驟用 Response Database 查詢(Then response status is 201), 「And 資料庫中存在, with table: name email 」。學習如何設計 ISA 讓 AI 能產生完整的 E2E Test，包含 API client setup, HTTP request/response handling, Database assertion, Error case handling。	0%	2025/11/20-2025/12/10	v	v	v	v	2025/11/25	
Ch5-ISA可執行規格 (90%)	用 ISA-Feature File 來產出高精度的前端 E2E Test	掌握語寫前端 ISA-Level Feature File 的技術，以技術接 API(CL) 的方式描述前端行為。例如 Given 步驟用頁面資料存取(Given 資料庫存在, with table: name email password), When 步驟用 API 呼叫(When 發送 POST /api/users, call table: name email password), Then 步驟用 UI 自動化模擬(Then 頁面顯示 dashboard, 「And 顯示歡迎訊息: Welcome, test@example.com」), Element locator, User interaction simulation, Assertion on UI state。	0%	2025/12/22		v	v	v	2025/12/22	
Ch5-ISA可執行規格 (90%)	掌握 ISA-Level 變數系統(Var-system)	深入理解 ISA-Level Var-system 如何處理變數產生的值。三種符號：(1) >contextKey(大於符號)：定義要存入 scenario context 的變數名稱，出現在 data table 的 header row, (2) <executionKey(小於符號)：指定要從 execution variable 取得變數的值，出現在 data table 的 data row, (3) \$variableName(等於號)：從 scenario context 中引用之前要保存的值，可出現在任何需引用的地方。實際應用場景：處理系統自動生成的 UUID, JWT Token 等無法直接賦值的值。實現變數的變數擴充與傳遞，並在後續步驟中使用。如 <user> 引用 \$user_id，變數產生的值會被存入 \$variableName，再由 <age> 引用 \$age，以此類推。	0%	2025/11/10-2025/11/20		v	v	v	2025/12/2	
Ch5-ISA可執行規格 (90%)	Design by Contract - 掌握 ISA-Level 約束語言系統(CAS-system)	學習 Gherkin Constraint Assertion System(CAS) 的實現範例。及各約束語彙涵括的八大範例：(1) 數值比較(&eq, <, >, &le, &ge)、(2) 類型檢查(&isNum, &isBool, &isNull, &isDef)、(3) 字串操作(&contains, &startsWith, &matches, &length)、(4) 集合操作(&hasItem, &oneOf)、(7) 時間相關(&before, &after, &withinDays, &sameDay)、(6) 函數調用(&called, &calledWith)、核心價值：驗證結果不需編碼具體性。如 <age> &between(16,65) &age < 25 。讓測試更真實、更穩定、更易維護。	0%	2025/11/10-2025/11/20		v	v	v	2025/12/2	
Ch5-ISA可執行規格 (90%)	Contract Testing - 透過合約的測試做到分散式系統的整合測試驅動開發	理解合約測試(Contract Testing)在微服務架構中的重要性與實作方式，問題背後：在分散式系統中，服務之間透過 API 相互依賴，如何確保 Provider(提供者)的變更不會破壞 Consumer(消費者)？傳統的 E2E 測試成本高、速度慢、難重複。合約測試：(1) Consumer 定義期的 API 約訂(request/response 格式)。(2) 從 feature file 生產合約範例。(3) Provider 驅動是否滿足所有 Consumer 的約訂。(4) 在 CI/CD 中執行執行合約驗證，學習如何從 ISA Feature File 映射到 Pact 合約定義，以及如何讓 Provider 與 Consumer 的約訂維繫，達到分散式系統的測試驅動開發。	0%	2026/1/20		v	v	v	2025/12/9	
Ch5-ISA可執行規格 (90%)	掌握 DSL level 翻譯成 ISA level 的實戰	以水球軟體學院課程平臺為實戰案例，學習完整的 DSL —> ISA 翻譯流程。翻譯步驟：(1) 讀取 DSL-Level Feature File(業務邏輯的 Gherkin)。(2) 讀取業務實體與動作(如:用戶註冊, 「課程購買」)。(3) 調整到技術實體與 API(如POST /api/users, [POST /api/orders])。(4) 調整 ISA-Level Feature File 技術面向的 Gherkin，包含 API endpoint, request/response 格式, database schema)。學習 DSL-to-ISA Translator 工具自動化翻譯 DSL 語彙，常見問題與最佳實務。	0%	2026/1/1		v	v	v	2025/12/9	
Ch6-自幹Agent (99%)	學會自幹一個可以自動化執行 BDD ISA Feature File to Auto Feedback Loop 的 workflow	學習如何用 Python 腳本控制整個 BDD 流程，達到 99% 自動化。核心步驟：(1) 從 ISA Feature File 讀取規格(parsing Gherkin syntax)。(2) 自動產生自動回歸測試(透過 LLM 或 Codegen)。(3) 執行測試並取得結果(test runner integration)。(4) 測試結果自動回歸修正(如果測試失敗，讓 AI 讀取錯誤訊息並修改程式碼)。(5) 持續循環直至所有測試通過(Auto-Feedback Loop)。Auto-Feedback Loop：實作技術：使用 CLI/SDK 工具(如 pytest, behave)，腳本化控制 Python(Shell)。藉助訊息分析與 AI prompt engineering，測試結果追蹤與報告。此工作 flow 是企業級 BDD 自動化的基礎。	0%	2025/12/8	v	v	v	v	2025/12/16	
Ch6-自幹Agent (99%)	學習如何使用 SDD.os 框架來做到全自動開發	掌握 SDD.os(Specification-Driven Development Operating System)的核心架構與運作機制。SDD.os 的五大核心：DSL Reader, 語意與解析 DSL 資料(ISA-level), Instruction Architecture (ISA), 定義 Step By Executor 的映射規則(使用正面表達式記號 Gherkin Steps, 調整對應的 Executor)。(3) Scheduler: 管理測試執行順序與依賴關係。(4) Assembler: 將 Feature File 翻譯成可執行的測試程式碼(透過 Step 的委託方法)。(5) Executors: 執行具體的測試動作(API call, Database query, UI interaction)。學習如何配置和使用 SDD.os 進行規格驅動開發，達到 99% 的全自動化開發能力(僅需人工撰寫 Feature File 與 Review 要求)。	0%	2025/12/8		v	v	v	2025/12/16	
Ch7-超AI化 (100%)	學會如何透過 SDD.os 來做到 100% 錄影不可能寫錯的測試程式碼生成達到 100% 全自動化開發	理解約 Codegen(不依賴 LLM)的運作原理與優勢。為什麼需要 100% Codegen? LLM 有機率產生錯誤的程式碼(即使用 99% 也代表每 100 次只有一次錯)，在測試系統中不能接受。AI 亂寫的作法是不可取的。定義 ISA 被約束的範例，並配合 SDD.os 進行自動化測試。Gherkin 有機率會因為錯誤而無法執行，但 AI 亂寫的程式碼卻能正確執行。Step 的委託方法：(1) 用正面表達式記號 Gherkin Steps, 調整對應的 Executor。(2) 用負面表達式記號 Gherkin Steps, 調整對應的 Executor。(3) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(4) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(5) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(6) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(7) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(8) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(9) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(10) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(11) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(12) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(13) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(14) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(15) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(16) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(17) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(18) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(19) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(20) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(21) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(22) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(23) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(24) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(25) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(26) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(27) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(28) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(29) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(30) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(31) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(32) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(33) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(34) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(35) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(36) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(37) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(38) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(39) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(40) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(41) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(42) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(43) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(44) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(45) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(46) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(47) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(48) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(49) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(50) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(51) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(52) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(53) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(54) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(55) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(56) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(57) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(58) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(59) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(60) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(61) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(62) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(63) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(64) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(65) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(66) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(67) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(68) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(69) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(70) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(71) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(72) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(73) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(74) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(75) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(76) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(77) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(78) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(79) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(80) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(81) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(82) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(83) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(84) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(85) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(86) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(87) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(88) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(89) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(90) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(91) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(92) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(93) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(94) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(95) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(96) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(97) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(98) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(99) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(100) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(101) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(102) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(103) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(104) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(105) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(106) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(107) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(108) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(109) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(110) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(111) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(112) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(113) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(114) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(115) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(116) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(117) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(118) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(119) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(120) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(121) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(122) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(123) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(124) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(125) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(126) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(127) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(128) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(129) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(130) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(131) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(132) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(133) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(134) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(135) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(136) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(137) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(138) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(139) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(140) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(141) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(142) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(143) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(144) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(145) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(146) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(147) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(148) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(149) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(150) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(151) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(152) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(153) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(154) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(155) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(156) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(157) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(158) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(159) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(160) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(161) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(162) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(163) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(164) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(165) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(166) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(167) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(168) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(169) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(170) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(171) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(172) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(173) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(174) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(175) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(176) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(177) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(178) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(179) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(180) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(181) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(182) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(183) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(184) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(185) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(186) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(187) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(188) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(189) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(190) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(191) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(192) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(193) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(194) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(195) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(196) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(197) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(198) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(199) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(200) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(201) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(202) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(203) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(204) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(205) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(206) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(207) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(208) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(209) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(210) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(211) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(212) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(213) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(214) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(215) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(216) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(217) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(218) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(219) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(220) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(221) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(222) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(223) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(224) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(225) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(226) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(227) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(228) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(229) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(230) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(231) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(232) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(233) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(234) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(235) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(236) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(237) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(238) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(239) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(240) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(241) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(242) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(243) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(244) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(245) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(246) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(247) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(248) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(249) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(250) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(251) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(252) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(253) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(254) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(255) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(256) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(257) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(258) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(259) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(260) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(261) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(262) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(263) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(264) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(265) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(266) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(267) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(268) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(269) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(270) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(271) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(272) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(273) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(274) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(275) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(276) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(277) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(278) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(279) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(280) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(281) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(282) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(283) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(284) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(285) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(286) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(287) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(288) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(289) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(290) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(291) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(292) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(293) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(294) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(295) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(296) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(297) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(298) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(299) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(300) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(301) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(302) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(303) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(304) 用子步驟記號 Gherkin Steps, 調整對應的 Executor。(305) 用子步驟記號 Gherkin Steps, 調整對應的								