AIT 1.2.0 / 協作指南

Agent Teams：協作與審查

以 1.2.0 實例說明角色分工、預設 agent 審查、並行 Task、修正交接與遠端故障續行。

角色分工、獨立審查、並行 Task 與故障續行。範例依已發布的 1.2.0；本指南不是完整 CLI 命令手冊。

https://ait-native.dev/zh-tw/development/1.2.0/agent-teams/
2026-11-02

目錄

1.2.0 · 參考 3

從需求到獨立審查完成的工作 3

預設 agent 審查與仍須符合的條件 3

初始化與保留既有對話 3

範例：一個需求、一個 Task、一份經審查的結果 3

範例：審查發現問題後如何修正 4

範例：兩位 Developer 同時開發 4

本機完成、遠端封存與重試 4

唯讀聯絡資料 5

準備既有成員 5

角色投遞與繼續工作 6

權威與版本邊界 6

1.2.0 • 參考

本 1.2.0 參考保留原本的 /development/ 網址，讓既有連結持續可用。安裝請看快速上手，各管道實際可用性與資格範圍以發布狀態為準。1.1.1 技術手冊與 PDF 仍為封存參考。

從需求到獨立審查完成的工作

Agent Teams 將需求、指定對話、隔離的 Task worktree、確切審查版本與原生完成結果連在一起。你決定目標與驗收條件；AIT 記錄並檢查工作流程，由 agent 執行規劃、實作與審查。

角色	職責	交接
Product Manager	釐清使用情境、範圍、優先順序與可衡量的驗收條件。	將已確定需求與必要決策交給 Project Manager。
Project Manager	準備可執行工單、相依工作、本機或已授權遠端範圍，指定 Developer 與獨立 Reviewer。	完整派工，協調範圍變更與阻礙。
Developer	在回傳的 edit_root 實作、執行相關檢查並凍結 Snapshot。	直接向指定 Reviewer 傳送 READY；在同一 Task 修正問題。
Reviewer	獨立檢查凍結版本與證據、查核必要條件，執行適用的 finish。	直接向 Developer 傳送 CHANGES_REQUIRED；原生完成後向 Project Manager 回報 COMPLETED。

Reviewer 不撰寫自己審查的候選版本。檢查失敗時，Reviewer 應查證到足以提出可執行的問題：受影響位置、觀察結果、要求的修正與驗證方法。只轉交 CI 失敗訊息不足以形成審查結論。

預設 agent 審查與仍須符合的條件

新的團隊候選版本以 `ai_only_experimental` 作為有效預設模式，由獨立 agent 審查程式碼，免額外的人類程式碼審查核准。這個設定有明確邊界：

- 明確選擇且有效的 author mode 優先。
- 啟用 Teams 會保留已儲存的 solo author-mode 預設值；儲存的值與新團隊候選版本的有效預設模式是兩件事。
- 已封存候選版本保留原有模式與身分。
- 實際 CI、Task approval 與強制 Policy 仍須符合。agent 訊息或閒置的 session 無法取代這些條件。

你仍負責產品意圖與驗收條件。預設模式免除額外的人類程式碼審查步驟，不代表產生的程式碼必定正確，也不代表所有候選版本都自動被接受。

初始化與保留既有對話

全新 `ait init` 預設啟用 Teams 並產生儲存庫工作指引。可用對話仍須由受支援的 agent host 準備。Teams、sprint 綁定與 Local/Remote 範圍是獨立選項；本機初始化不需要執行中的 Server。

既有儲存庫重新初始化時，設定與 session 綁定保持不變。儲存庫已有綁定時，重新初始化或啟用 Teams 會同步適用的設定與指引檔，不替換或重新載入那些對話。觀察到封存、缺漏或未知狀態，不等於可以丟棄已儲存的綁定。尚無綁定的首次設定才走初始啟用流程。

有需要時仍可明確準備 session；準備對話不等於派發 Task。確切前提請看初始化與設定參考及下方 session 準備說明。

範例：一個需求、一個 Task、一份經審查的結果

1. 規劃：Project Manager 準備 sprint 卡，寫明範圍、驗收、相依工作、穩定 Plan 參照、確切未勾選項目、Developer 與獨立 Reviewer。
2. 啟動：Developer 從該項目啟動 Task，只在回傳的 edit_root 工作。Task start 會同步並綁定初始卡片。
3. 實作與檢查：首次建置或擷取版本前先檢查 `.aitignore`，完成實作及相關檢查，再凍結 Snapshot。

4. 審查：Developer 直接傳送 READY 給 Reviewer；Reviewer 檢查該確切 Snapshot，執行獨立及必要檢查。
5. 完成：符合適用條件後，由 Reviewer 執行原生 finish。成功輸出會回報 Task 完成、worktree 清理與適用的綁定項目關閉。

以下示範有 sprint 綁定的本機任務。請依已安裝版本產生的命令操作，並代入實際回傳身分；範例中的佔位符不能直接當成 ID 執行。

```
ait task start --from docs/sprints/shop.md#shop/discount --intent "Discount rule" --local
# In the returned edit_root, after implementation and checks:
ait snapshot create <TASK_ID> --message "Discount rule ready for review"
```

直接交接使用簡短純文字。詳細證據留在原生結果與相關檢查中，不另加 JSON 訊息封裝。

```
READY task=<TASK_ID> round=R01 snapshot=<SNAPSHOT_ID>
```

對乾淨且已審查的本機候選版本，Reviewer 依適用的產生式流程完成，例如 `ait task finish <TASK_ID> --local`。單次 Snapshot、CI 成功或對話回覆都不等於 Task 完成；不要手動勾選 sprint 項目來模擬結案。

範例：審查發現問題後如何修正

假設折扣實作重複四捨五入，導致總額錯誤。Reviewer 直接向 Developer 回覆具體問題：

```
CHANGES_REQUIRED task=<TASK_ID> round=R01 snapshot=<REVIEWED_SNAPSHOT>
F01 | src/price.rs:42 | rounds twice | round final total once | test boundary
```

初始 Task 工作與第一輪審查不建立 Change。開始真正修正之前，若尚無修正 Change，Developer 從確切受審 Snapshot 建立有意義標題的 Change：

```
ait change create <TASK_ID> --title "Fix rounding" --fork-snapshot <REVIEWED_SNAPSHOT> --local
```

在相同範圍內沿用這個 Change 修正，重跑受影響檢查並擷取替代 Snapshot，再交給原 Reviewer：

```
READY task=<TASK_ID> round=R02 snapshot=<REPLACEMENT_SNAPSHOT> resolves=R01/F01
```

Reviewer 查證修正後才關閉問題。新一輪審查不需要新 Task、另一張工單或例行 Project Manager 核准。只有版本差異、輸入與涵蓋範圍仍支持沿用時，才重用既有證據；範圍或擁有權變更交回 Project Manager。

範例：兩位 Developer 同時開發

Developer A 修改價格計算，Developer B 修改說明文字。兩者各有獨立 Task、worktree，以及啟用 sprint 綁定時各自的需求。實作與驗證可以並行。

1. A 先取得審查通過的結果並完成。
2. B 套用前，AIT 會對照已前進的共用目標重新檢查；相容變更在符合適用驗證後協調整合。
3. 若兩個 Task 對同一邏輯的修改不相容，完成程序會停止，由負責 Developer 修正衝突，再將替代版本交回審查。
4. 各 Task 成功 finish 後，清理自己的 worktree 並關閉適用的綁定項目。

Task 隔離不代表所有寫入都互不影響。`docs/` 與儲存庫根目錄共用，其他共用目標或工作區也須協調。Worktree 是工作流程邊界，不是作業系統沙箱。最終驗證須對應受審版本；finish 可能移除工作區，因此完成時應暫停相關寫入者。

本機完成、遠端封存與重試

Local finish 與遠端正式接受是不同結果。本機審查本身不授權遠端準備或發布。已授權的遠端整合版本由一位指定 Reviewer 負責 readiness、CI 與受保護的 finish。

`ait workflow ready` 預設唯讀檢查 readiness；加 `--apply` 才準備候選版本。保留原生命令回傳的確切身分，依其下一步操作；不同的整合版本須取得適用的審查與驗證。

情境	必要行為
遠端候選 S1 驗證中，另一個 Local Task 完成。	S1 維持封存。後續 Local finish 不改寫遠端候選，也不改變該次 CI 檢查的內容。
Server 在正式接受前重啟。	沿用原始完整 expected identity 與操作脈絡，執行已接納的 ready apply。核對新 Job 使用相同封存內容與原模式，重新取得 CI 與必要審查；身分缺漏或矛盾時停止復原。
Server 已接受 finish，但用戶端沒有收到回應。	正式重試沿用原 acceptance。保留成功結果，只續行失敗階段，不重複套用。

ready/finish 細節請看1.2.0 工作流指南。原生完成結果才是完成證據；session 閒置或單一 CI 工作成功都不足以證明。

唯讀聯絡資料

```
ait team contacts --json
```

此命令讀取本機綁定與原生可用性，不建立、替換或重新載入 session。Teams 停用時，`team_enabled: false` 與空的 `members` 不代表保留的名單遭刪除。

欄位	意義與限制
`planned_seat_count`	基準席位數，不是可用或執行中 session 數。四個規劃角色涵蓋 Project Manager、Product Manager、Developer 與獨立 Reviewer。
`actual_member_count` / `bound_session_count`	列出成員數／非空 session 綁定數，都不是執行中工作數。
`agent_index` / `session_id`	成員身分與原生 session 綁定。查詢桌面狀態時，同時核對指定 session 與 host ID。
角色路徑／revision、`loaded_role_revision_index`、`role_reload_status`、`role_state`	指定角色與投遞觀察。載入角色的中繼資料只記錄投遞，不證明仍保有上下文。
`availability` / `host_status`	原生可用性與主機觀察。`available` 可包含 `notLoaded`，都不證明工作執行中。

未綁定、封存、缺漏、不相符與未知觀察都不能證明 session 可用。另以桌面 `list_threads` 工具查詢 running/idle，並同時比對 session 與 host ID。Contacts 無法權威判定另一桌面主機的執行狀態。篩除無關資料；有限列表可能漏掉 session，缺漏、失敗觀察與 `notLoaded` 應維持未知。Idle 不證明 Task 完成；原生 Task/workflow finish 才能證明。

準備既有成員

以下為語法，請以指定成員與編號角色替換參數。不要為了查執行狀態而呼叫 ensure。

```
ait team session ensure --agent <AGENT> --json
ait team session ensure --agent <AGENT> --role <ROLE> --json
```

Ensure 建立、沿用或復原既有成員的 session，並投遞編號的 `docs/roles/` Plan。額外成員需要明確角色。沿用有效綁定，只有證明原始原生儲存中不存在時才能替換。主機證據不確定時，保留綁定並回報阻礙；執行中的 session 不代表不存在。準備 Reviewer 需要伺服器註冊。

原生輸出提供啟用要求時，在同一對話完成必要 app 動作，再確認只屬於該對話的 token：

```
ait team session acknowledge --activation <ACTIVATION> --json
```

確認會核對標題、專案、綁定與角色收件，不會指派工作。準備回傳 `work_started: false`；PM 另行指派意圖、範圍、卡片與 Reviewer。準備好的席位不代表工作執行中或 Task 完成。

角色投遞與繼續工作

Plan sync 記錄角色修改，不會自動重新載入 session。需要投遞變更角色時使用受保護的準備流程。新 Task 本身不需要重新載入。壓縮上下文後，恢復完整適用指引與目前工作單。停用 Teams 保留 session、綁定、角色歷史與未完成 Task；明確交接前，既有擁有權與門檻仍有效。

歷史 workflow-mode 與 team-size 預設不是現行命令。Task 啟動、獨立審查與 finish 以產生的 `AGENTS.md` 為準；獨立的 Teams、sprint 與範圍設定見初始化與設定參考。

權威與版本邊界

本指南依據已發布的 1.2.0 原始碼與已審查的 Agent Teams 說明。PDF 由本頁同一份來源產生；歷史手冊保留原版本。

本參考依據 Core 已審查的 Agent Teams、init、config、Task 與 workflow 參考，以及 Native 現行 distribution 與 release SOP。現行版本與管道記錄請看發布狀態。定版的 1.1.4 發布來源記錄歷史 1.1.4 基準；封存技術手冊與 PDF 保留原版本及證據。