AIT 1.2.0 / 协作指南

Agent Teams：协作与审查

以 1.2.0 实例说明角色分工、默认 agent 审查、并行 Task、修正交接与远程故障恢复。

角色分工、独立审查、并行 Task 与故障恢复。示例依据已发布的 1.2.0；本指南不是完整 CLI 命令手册。

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

目录

1.2.0 · 参考 3

从需求到独立审查完成的工作 3

默认 agent 审查与仍须满足的条件 3

初始化与保留已有对话 3

示例：一个需求、一个 Task、一份经审查的结果 3

示例：审查发现问题后如何修正 4

示例：两位 Developer 同时开发 4

本地完成、远程封存与重试 4

只读联系信息 5

准备已有成员 5

角色交付与继续工作 6

权威与版本边界 6

1.2.0 • 参考

本 1.2.0 参考保留原有 /development/ 网址，让已有链接持续可用。安装请看快速上手，各渠道实际可用性与资格范围以发布状态为准。1.1.1 技术手册与 PDF 仍为归档参考。

从需求到独立审查完成的工作

Agent Teams 将需求、指定对话、隔离的 Task worktree、准确的审查版本与原生完成结果连在一起。你决定目标与验收条件；AIT 记录并检查工作流，由 agent 执行规划、实现与审查。

角色	职责	交接
Product Manager	明确使用场景、范围、优先级与可衡量的验收条件。	将已确定的需求与必要决策交给 Project Manager。
Project Manager	准备可执行工单、依赖工作、本地或已授权远程范围，指定 Developer 与独立 Reviewer。	完整派工，协调范围变更与阻碍。
Developer	在返回的 edit_root 实现、执行相关检查并冻结 Snapshot。	直接向指定 Reviewer 发送 READY；在同一 Task 修正问题。
Reviewer	独立检查冻结版本与证据、核查必要条件，执行适用的 finish。	直接向 Developer 发送 CHANGES_REQUIRED；原生完成后向 Project Manager 报告 COMPLETED。

Reviewer 不编写自己审查的候选版本。检查失败时，Reviewer 应核查到足以提出可执行的问题：受影响位置、观察结果、要求的修正与验证方法。只转交 CI 失败消息不足以形成审查结论。

默认 agent 审查与仍须满足的条件

新的团队候选版本以 `ai_only_experimental` 作为有效默认模式，由独立 agent 审查代码，无须额外的人工代码审查批准。这项设置有明确边界：

- 明确选择且有效的 author mode 优先。
- 启用 Teams 会保留已保存的 solo author-mode 默认值；保存的值与新团队候选版本的有效默认模式是两件事。
- 已封存候选版本保留原有模式与身份。
- 实际 CI、Task approval 与强制 Policy 仍须满足。agent 消息或空闲的 session 无法替代这些条件。

你仍负责产品意图与验收条件。默认模式免除额外的人工代码审查步骤，不代表生成的代码必定正确，也不代表所有候选版本都自动被接受。

初始化与保留已有对话

全新 `ait init` 默认启用 Teams 并生成仓库工作指引。可用对话仍须由受支持的 agent host 准备。Teams、sprint 绑定与 Local/Remote 范围是独立选项；本地初始化不需要运行中的 Server。

已有仓库重新初始化时，设置与 session 绑定保持不变。仓库已有绑定时，重新初始化或启用 Teams 会同步适用的设置与指引文件，不替换或重新加载那些对话。观察到归档、缺失或未知状态，不等于可以丢弃已保存的绑定。尚无绑定的首次设置才走初始启用流程。

有需要时仍可明确准备 session；准备对话不等于派发 Task。准确前提请看初始化与配置参考及下方 session 准备说明。

示例：一个需求、一个 Task、一份经审查的结果

1. 规划：Project Manager 准备 sprint 卡，写明范围、验收、依赖工作、稳定 Plan 引用、准确的未勾选条目、Developer 与独立 Reviewer。
2. 启动：Developer 从该条目启动 Task，只在返回的 edit_root 工作。Task start 会同步并绑定初始卡片。
3. 实现与检查：首次构建或捕获版本前先检查 `.aitignore`，完成实现及相关检查，再冻结 Snapshot。
4. 审查：Developer 直接发送 READY 给 Reviewer；Reviewer 检查该准确 Snapshot，执行独立及必要检查。

5. 完成：满足适用条件后，由 Reviewer 执行原生 finish。成功输出会报告 Task 完成、worktree 清理与适用的绑定条目关闭。

以下演示有 sprint 绑定的本地任务。请依已安装版本生成的命令操作，并代入实际返回的身份；示例占位符不能直接作为 ID 执行。

```
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"
```

直接交接使用简短纯文本。详细证据留在原生结果与相关检查中，不另加 JSON 消息封装。

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

对干净且已审查的本地候选版本，Reviewer 依适用的生成式流程完成，例如 `ait task finish <TASK_ID> --local`。单次 Snapshot、CI 成功或对话回复都不等于 Task 完成；不要手动勾选 sprint 条目来模拟结案。

示例：审查发现问题后如何修正

假设折扣实现重复四舍五入，导致总额错误。Reviewer 直接向 Developer 回复具体问题：

```
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 工作与第一轮审查不创建 Change。开始真正修正之前，若尚无修正 Change，Developer 从准确的受审 Snapshot 创建有意义标题的 Change：

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

在相同范围内沿用这个 Change 修正，重跑受影响检查并捕获替代 Snapshot，再交给原 Reviewer：

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

Reviewer 核查修正后才关闭问题。新一轮审查不需要新 Task、另一张工单或例行 Project Manager 批准。只有版本差异、输入与覆盖范围仍支持沿用时，才复用已有证据；范围或所有权变更交回 Project Manager。

示例：两位 Developer 同时开发

Developer A 修改价格计算，Developer B 修改说明文字。两者各有独立 Task、worktree，以及启用 sprint 绑定时各自的需求。实现与验证可以并行。

1. A 先取得审查通过的结果并完成。
2. B 应用前，AIT 会对照已前进的共享目标重新检查；兼容变更在满足适用验证后协调合入。
3. 若两个 Task 对同一逻辑的修改不兼容，完成程序会停止，由负责 Developer 修正冲突，再将替代版本交回审查。
4. 各 Task 成功 finish 后，清理自己的 worktree 并关闭适用的绑定条目。

Task 隔离不代表所有写入都互不影响。`docs/` 与仓库根目录共享，其他共享目标或工作区也须协调。Worktree 是工作流边界，不是操作系统沙箱。最终验证须对应受审版本；finish 可能移除工作区，因此完成时应暂停相关写入者。

本地完成、远程封存与重试

Local finish 与远程正式接受是不同结果。本地审查本身不授权远程准备或发布。已授权的远程集成版本由一位指定 Reviewer 负责 readiness、CI 与受保护的 finish。

`ait workflow ready` 默认只读检查 readiness；加 `--apply` 才准备候选版本。保留原生命令返回的准确身份，依其下一步操作；不同的集成版本须取得适用的审查与验证。

场景	必要行为
远程候选 S1 验证中，另一个 Local Task 完成。	S1 维持封存。后续 Local finish 不改写远程候选，也不改变该次 CI 检查的内容。
Server 在正式接受前重启。	沿用原始完整 expected identity 与操作上下文，执行已接纳的 ready apply。核对新 Job 使用相同封存内容与原模式，重新取得 CI 与必要审查；身份缺失或矛盾时停止恢复。
Server 已接受 finish，但客户端没有收到回复。	正式重试沿用原 acceptance。保留成功结果，只继续失败阶段，不重复应用。

ready/finish 详情请看1.2.0 工作流指南。原生完成结果才是完成证据；session 空闲或单一 CI 作业成功都不足以证明。

只读联系信息

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

此命令读取本地绑定与原生可用性，不创建、替换或重新加载 session。Teams 禁用时，`team_enabled: false` 与空的 `members` 不代表保留的名单被删除。

字段	含义与限制
`planned_seat_count`	基准席位数，不是可用或运行中 session 数。四个规划角色涵盖 Project Manager、Product Manager、Developer 与独立 Reviewer。
`actual_member_count` / `bound_session_count`	列出成员数／非空 session 绑定数，都不是运行中工作数。
`agent_index` / `session_id`	成员身份与原生 session 绑定。查询桌面状态时，同时核对指定 session 与 host ID。
角色路径／revision、`loaded_role_revision_index`、`role_reload_status`、`role_state`	指定角色与交付观察。加载角色的元数据只记录交付，不证明仍保有上下文。
`availability` / `host_status`	原生可用性与主机观察。`available` 可包含 `notLoaded`，都不证明工作运行中。

未绑定、归档、缺失、不匹配与未知观察都不能证明 session 可用。另用桌面 `list_threads` 工具查询 running/idle，并同时比对 session 与 host ID。Contacts 无法权威判定另一桌面主机的运行状态。筛除无关数据；有限列表可能漏掉 session，缺失、失败观察与 `notLoaded` 应维持未知。Idle 不证明 Task 完成；原生 Task/workflow finish 才能证明。

准备已有成员

以下为语法，请用指定成员与编号角色替换参数。不要为了查运行状态而调用 ensure。

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

Ensure 创建、沿用或恢复已有成员的 session，并交付编号的 `docs/roles/` Plan。额外成员需要显式角色。沿用有效绑定，只有证明原始原生存储中不存在时才能替换。主机证据不确定时，保留绑定并报告阻碍；运行中的 session 不代表不存在。准备 Reviewer 需要服务器注册。

原生输出提供激活要求时，在同一对话完成必要 app 操作，再确认只属于该对话的 token：

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

确认会核对标题、项目、绑定与角色收件，不会分配工作。准备返回 `work_started: false`；PM 另行分配意图、范围、卡片与 Reviewer。准备好的席位不代表工作运行中或 Task 完成。

角色交付与继续工作

Plan sync 记录角色修改，不会自动重新加载 session。需要交付变更角色时使用受保护的准备流程。新 Task 本身不需要重新加载。压缩上下文后，恢复完整适用指引与当前工作单。禁用 Teams 保留 session、绑定、角色历史与未完成 Task；显式交接前，已有所有权与门槛仍有效。

历史 workflow-mode 与 team-size 预设不是当前命令。Task 启动、独立审查与 finish 以生成的 `AGENTS.md` 为准；独立的 Teams、sprint 与范围设置见初始化与配置参考。

权威与版本边界

本指南依据已发布的 1.2.0 源代码与已审查的 Agent Teams 说明。PDF 由本页同一份来源生成；历史手册保留原版本。

本参考依据 Core 已审查的 Agent Teams、init、config、Task 与 workflow 参考，以及 Native 当前 distribution 与 release SOP。当前版本与渠道记录请看发布状态。定版的 1.1.4 发布来源记录历史 1.1.4 基线；归档技术手册与 PDF 保留原版本及证据。