Brady's Squad: Human-led AI teams

Module 5: Specification-driven frameworks | Advanced

Session M5-01 | 3 hours | Optional

Prerequisites: Sessions M1-01 to M1-03, M2-01 to M2-04 and M3-02 to M3-05

Agenda

Segment	Time
Coordination trade-offs	7 min
Roles, routing, and memory	20 min
Issues and monitored work	8 min
Prepared team demonstration	12 min
Decision framework and lab handoff	13 min
Lab	2 hours

Coordination has a cost

Use a team when work has separate owners, clear interfaces, and a decision record to share. An individual agent is better for a small, well-understood change.

More agents create more state, review work, and cost.

Parallelism only helps with real independence

Work can run in parallel when each stream has its own inputs, a clear output, and no hidden dependency on another stream's decision.

Work item	Parallel?	Reason
Review separate test files	Often	Each reviewer can return findings
Choose a shared domain model	No	Every implementation depends on the decision
Update unrelated documentation	Often	Output boundaries are clear

Splitting an ambiguous task across agents spreads the ambiguity.

The coordinator owns the integration risk

A coordinator needs to know:

1. Which decision makes the workstreams compatible.
2. Which files or systems only one role may change.
3. Which results each role must return.
4. When a conflict should stop the workflow rather than be merged automatically.

The coordinator stays human-led even when agents do the initial work.

Return a result summary

Use:

```
Scope completed:
Evidence:
Files or records changed:
Checks run:
Open questions:
Recommendation:
```

The parent can compare these summaries without reading each agent's full conversation.

Report missing results. Do not guess.

Coordinator pattern

```
Issue with acceptance criteria
        ↓
Coordinator selects a bounded role
        ↓
Role writes a proposal or change
        ↓
Focused checks and human review
        ↓
Decision record, merge, follow-up, or pause
```

The coordinator assigns work with a defined scope. Clarify an ambiguous request before implementation.

Team boundaries

Role	Responsibility
Lead	Scope, decisions, review gate
Implementer	A defined product area or file set
Tester	Focused checks and reported gaps
Scribe	Shared decisions and history
Work monitor	Flag ready, approved work

Agents need charters that state inputs, outputs, and limits. Avoid concurrent writes to the same files.

Shared memory

Commit decision records with the choice, supporting results, owner, and date. Keep temporary drafts separate.

Agents can write individual proposals. A designated owner reviews and merges the shared record. Git history keeps the trail.

Issues and monitored work

An issue is a work contract. It needs acceptance criteria, non-goals, constraints, checks, and a reviewer.

```
Ready issue → assignment → bounded change → focused checks →
human review → merge or follow-up issue
```

Treat Ralph as a monitored queue. Stop when approval, ownership, policy, test status, scope, or usage limits are unclear.

When orchestration helps

Situation	Path
One small repair	Individual agent or manual work
Separate work with defined interfaces	Coordinated team
Overlapping ownership or unclear design	Plan before implementation
Restricted path or unapproved tool	Manual workflow

One issue, one review trail

The worked issue adds validation to `POST /api/users`.

```
Issue contract
  → lead assignment
  → backend implementation
  → tester evidence
  → lead review
  → durable decision
```

Keep the same acceptance criteria through every step. A role may add supporting results, but it must not quietly expand the issue.

Reference `.squad/` result

```
.squad/
├── team.md
├── routing.md
├── decisions.md
├── agents/{lead,backend,tester,scribe}/
├── decisions/inbox/issue-001-proposal.md
├── orchestration-log/issue-001.md
└── reviews/issue-001-review.md
```

The reference project also includes the implemented route and passing tests.

Required access

Check GitHub Copilot, Squad, repository integration,
and the network path before the lab. If a requirement is unavailable, stop and resolve it. A tabletop simulation does not demonstrate orchestration.

Lab handoff

Copy the starter project and check Squad access before work begins.

Submit:

- one issue assignment;
- the input-validation implementation;
- passing focused tests;
- a lead review and decision record.

Use `lab/solution/squad-project/` to check the expected result.