Orchestrate agents and workstreams

Product and delivery teams, or developers after Session M2-05

Session M7-05 | 3 hours

Agenda

Segment	Time
Parent objective and boundaries	10 min
Minimal workstream plan	10 min
Monitor and intervene	12 min
Live demonstration	13 min
Review and lab handoff	15 min

Required access

- `/orchestrate` must create and coordinate live child sessions.
- Product route: GitHub MCP must read the selected issues.
- Developer route: local project read access and Python 3.10 or later.
- You must be able to open, redirect, and stop child sessions.

Stop if orchestration or the selected route's inputs are unavailable.

Choose one lab route

Route	Inputs	Result
Product and delivery	Four live GitHub issues	Parent result used by Session M7-06
Developer	Local evaluator, tests, synthetic PR snapshot	Separate implementation, coverage, and PR conclusions

Each route takes 120 minutes. Developers need Session M2-05, not Sessions M7-01 to M7-04.

One parent objective

Review the four approved status-visibility issues and return one reviewed result for each.

The parent owns scope and review.

Keep the plan small

Each workstream needs:

- a question with a clear scope;
- approved inputs;
- a stop condition;
- a reviewer.

The orchestration surface already holds session links, status, timing, and usage.

Four independent workstreams

Workstream	Question
Public mapping	Is the mapping ready for approval?
Requester status	Is the issue ready for implementation review?
Empty and stale states	Are fallback states testable?
Access boundary	Are the approved requester-access results ready?

Move shared questions back to the parent.

Developer workstreams

Question	Input
Does implementation meet the behavior contract?	Brief and evaluator
What do tests prove or miss?	Brief, tests, and evaluator
Is the supplied PR ready?	Brief and PR snapshot

Children read the same fixed inputs. They do not wait for one another or edit.

Practise steering

Redirect the test reviewer after its first answer:

```
Check criterion 5. Does the test prove sorted reason codes, the exact complete
reason list, or both? Do not edit files.
```

Inspect the revised answer. A follow-up question does not prove the first answer was wrong.

Reconcile the readiness claim

The synthetic author says: "Local tests pass, so the PR is READY."

Check the actual snapshot:

- required `integration-tests` failed;
- blocking thread `R1` is unresolved.

Check the parent's decision against the snapshot. Do not manufacture conflicting child answers. Local success does not override PR blockers.

Start live orchestration

```
/orchestrate Use the four workstreams in my approved plan. Keep them independent.
Each result must answer its question, cite approved inputs, state unresolved
gaps, and give a recommendation. Obey each stop condition.
```

Monitor on the platform

Signal	Action
Safe and on scope	Wait
Safe work needs correction	Redirect
A boundary is crossed	Stop
The result meets the plan	Accept
The result lacks support	Reject
Human authority is required	Escalate

Do not maintain a second status log.

Review each result

- Does it answer the assigned question?
- Does it use only the approved issues and their linked sources?
- Did it obey the stop condition?
- Are gaps listed?
- Is the recommendation supported?

Reject answers that lack supporting sources.

The parent result is the handoff

The reviewed result:

- links findings to live GitHub issues;
- shows rejected claims;
- names gaps and owners;
- recommends the next follow-up.

Session M7-06 reads this result and the live issues.

The human keeps authority

Copilot coordinates work. A person reviews results and approves any issue update, publication, or release statement.

Session M5-01 is different

Session M5-01	Session M7-05
Persistent roles and routing	Temporary workstreams
Optional framework setup	Built-in orchestration
Shared memory	Live issues and their supporting sources

Lab deliverable

1. Approved orchestration plan
2. Live child-session results
3. Reviewed parent result linked to live issues

Keep progress on the orchestration surface. Do not copy it into a delivery-status form or copy session identifiers.

For developers, use the developer plan and reviewed parent result instead of live issues. Include the actual redirection, rejected or unresolved claims, and observed usage.