

0-07

How I Work, Async

The operating manual for the brief, the plan, the daily update, pull requests, review, handover and response times in CET.

Axdale. Software engineering by Michael Lip, Warsaw, Poland (CET).

Website <https://www.axdale.com>. Code portfolio on GitHub at <https://github.com/theluckystrike>.

Version 1.0. 23 October 2026.

This manual tells you what I send, when I send it and what I need from you. It's part of how I deliver work under the Axdale Engagement Agreement.

1. The short version

1. You send a written brief. I send a written plan within 1 Working Day.
2. Your Decision Maker approves the acceptance criteria in writing before day 1.
3. I work in Weeks of up to 40 hours over 5 Working Days. The first Week is free once both sides sign the Engagement Agreement. After that, Weeks are paid one at a time.
4. You get a written update by 18:00 CET every Working Day, a weekly report at the end of each Week, and every change as a pull request with tests.
5. No calls, no meetings and no video. Everything is in writing, so nothing depends on memory.

2. Why async

Written work leaves a record. You can read it when it suits your day, forward it to your team and check it later. A decision in a pull request comment doesn't get lost the way a spoken one does. And time zones stop mattering. Teams in London get my update the same afternoon, and teams in New York get it in their morning.

3. The brief

Send your brief by email to michael@axdale.com. It can be a few lines or several pages, and any context you can share helps, such as docs, tickets or past incidents. Answer these 5 questions.

1. What should change for your users, and how will you know it worked?
2. Where does the code live, and what is the stack?
3. What is out of bounds? Name any systems, data or AI providers I must not touch.
4. Who is the Decision Maker who approves the plan and signs off?
5. What is the size of the whole job, and is there a date that matters?

A short brief is fine. I ask written questions if something is missing.

4. The plan

I reply within 1 Working Day with a written plan. The plan has these parts.

1. The goal for the first Week in one sentence.
2. 3 to 5 acceptance criteria.
3. What is in scope and what is out of scope.
4. The access I need, kept to the minimum.
5. The paid follow-on, with the number of Weeks and the hours I estimate.

The plan becomes the Statement of Work once your Decision Maker approves it in writing.

5. Acceptance criteria

Each criterion has 4 parts. Given is the starting state and When is the action. Then is the result you can see, and Check is the test, command or screenshot that proves it.

Criteria are fixed before day 1. A new idea during the Week goes into the plan for the next Week. This keeps each Week honest and easy to judge.

6. The daily written update

You get one update by 18:00 CET on each Working Day. It goes by email, or as a comment in your tracker if you prefer. It always has the same 6 parts.

Day 3 of 5, Week 2, SOW EXAMPLE-02

Shipped

- PR #41 eval set grown from 15 to 28 cases (review requested)
- Fixed timeout on long tickets, test added

Next

- Add the eval run as a CI gate
- Token cap per request

Blocked

- Need a staging test key with a higher spend cap (asked in issue #38)

Decisions needed from you

- Keep the 15 second limit, or allow 20 seconds for long tickets?

Hours

- 7.5 today, 23.0 of 40 this Week

Risks

- None new

If nothing is blocked, the update says "Blocked. Nothing." I never skip an update on a Working Day of a Week.

7. Pull request rules

One pull request does one thing. I keep each one small enough to review in about 20 minutes, and every pull request has tests or says in writing why a test isn't possible. CI must pass before I ask for review.

The description has 5 parts. What changed, why, how to test it, risks, and which acceptance criterion it serves. I also list each new dependency and its license.

I never force-push to a shared branch and I never push to your main branch directly. I don't merge my own pull request unless you allow that in writing for a named repository.

I review every change that an AI tool helped to write before I open the pull request, and I say in the description when an AI tool helped. Nothing goes to production without written approval from your Decision Maker for that specific deploy.

8. Review

1. Please review pull requests within 1 Working Day. Comments in the pull request are the best channel.
2. I answer review comments within 4 working hours.

3. At the end of each Week you have 5 Working Days to tell me in writing which criteria fail. After that, the Week counts as accepted.
4. In a Paid Week, I fix failed criteria at no charge if you report them within 10 Working Days after the handover.

9. Questions and blockers

1. I ask questions in writing and group them, so you aren't interrupted many times a day.
2. Each question offers a default answer. If you agree, you can reply "OK".
3. If a question blocks the work, I say so in the daily update and work on the next unblocked item.
4. In the free first Week, 2 Working Days of silence pause the Week. 5 Working Days of silence end it.

10. The weekly report

On the last Working Day of each Week you get a weekly report. It shows the status of each acceptance criterion, the pull requests, the hours, the measurements, the risks and the plan for the next Week. The Weekly Sprint Report template shows the exact format.

11. Handover

At the end of each Week, and when the engagement ends, you get a handover note with 6 parts.

Part	What it covers
Run	How to start the work on a clean machine, with exact commands
Test	How to run the tests and the evaluation set
Change	Where the main parts live and how to change them safely
Known issues	What isn't done or not perfect, said plainly
Next steps	What I would do next, in order
Access	The list of access I had and the date I removed or returned each one

12. Response times

All times are Warsaw local time (CET, or CEST in summer). Working hours are 09:00 to 17:00, Monday to Friday, except public holidays in Poland.

Item	When you get it
Reply to a new brief	Within 1 Working Day
Written plan and acceptance criteria	Within 1 Working Day after the brief
Reply to a question or review comment during a Week	Within 4 working hours
Daily written update	By 18:00 CET on each Working Day of a Week
Weekly report	By 18:00 CET on the last Working Day of each Week
Handover note when the engagement ends	Within 2 Working Days
Notice of a security incident	Within 24 hours
Data deletion after the engagement ends	Within 5 Working Days, with written confirmation

What I need from you	When
Answer to a blocking question	Within 1 Working Day where possible
Pull request review	Within 1 Working Day where possible
Written list of failed criteria after a Week	Within 5 Working Days

13. Tools

Purpose	Tool
Code and review	Your GitHub or GitLab, on a branch or a fork
Tasks	Your issue tracker (for example GitHub Issues, Linear or Jira), or a simple list in the pull request
Updates, approvals and notices	Email to and from michael@axdale.com
Secrets	Your secret manager, or a one-time encrypted link
AI coding tools	Only the providers you approve in writing. My defaults are Claude Code (Anthropic) and Codex (OpenAI)
Documents	Markdown in the repository, so the documentation lives next to the code

I work inside your tools. I don't ask you to adopt new ones.

14. What I don't do

1. Calls, meetings, video or screen shares.
2. On-call or out-of-hours support, unless a Statement of Work says so.
3. Production changes without written approval.
4. Work with no written brief.
5. Work on a provider you haven't approved.

15. Holidays and leave

I don't work on public holidays in Poland. A Week skips them and ends on the next Working Day. I give at least 5 Working Days' written notice of any other leave, and I never start a Week that my leave would break.

16. Changes to this manual

I may improve this manual. I tell you in writing before a change applies to your work. A change never reduces what an accepted Statement of Work promises.

Axdale. Michael Lip, independent engineer, Warsaw, Poland (CET). michael@axdale.com.

Axdale, Michael Lip. Warsaw, Poland (CET). michael@axdale.com.
www.axdale.com. Code record github.com/theluckystrike. Version
1.0. 23 October 2026.