

01

Mob Troubleshooting Workshop Design

Setup (5 min)

- **Facilitator explains rules:**
 - One engineer at a time is “driver” (controls screen & talks through their reasoning).
 - They troubleshoot the issue caused by a **chaos experiment** (pre-triggered or launched live).
 - Each driver gets **max 2 minutes**. If they don't make progress, rotate to next engineer.
- **Observers:**
 - Stay quiet, unless the driver asks for clarification.
 - If they would approach differently, **they must write it down** in a shared doc or sticky notes (e.g., “I'd check service X metrics first” or “I'd sketch dependency graph here”).
- **Assign roles:**
 - **Facilitator** (timekeeper, ensures rotation).
 - **Note-taker** (collects alternative paths).

Experiment Trigger (0–2 min)

- Run a **predefined chaos experiment** (simple = e.g., increase latency on one service; complex = partial outage in a dependency).
- Verify the team sees the failure symptoms (alerts firing, dashboards showing red, user-facing issue).

Mob Troubleshooting Rotation (12–30 min)

- **Each engineer gets 2 minutes max:**
 - Take control.
 - Think out loud: explain what you're checking, why, and what hypothesis you're testing.
 - Run one or two checks (query logs, open dashboard, curl endpoint, etc.).

→ **If progress:** continue.
→ **If stuck:** facilitator rotates to next engineer.

Reflection & Sharing (8–10 min)

- Everyone shares their **written alternative approaches**.
- **Discuss patterns:**
 - What were common strategies?
 - Where did people diverge?
 - Did someone's preferred tool/query/schema save time?
- Capture **gaps** (missing dashboards, missing runbooks, poor alerts, unclear ownership).

Wrap-up (3–5 min)

- **Facilitator highlights:**
 - Key learnings about troubleshooting flow.
 - Improvements for observability/runbooks.
 - How different mental models enrich team resilience.




GOALS

- **Practice** structured troubleshooting in a collaborative way.
- **Compare** different approaches (dashboards, queries, diagrams, intuition).
- Build **shared mental models** of the system under stress.
- Capture **insights** to improve observability and runbooks.


DURATION

- **20 minutes** if the experiment is simple (one bottleneck, clear symptom).
- **40 minutes** if complex (multi-service impact, cascading failure).

VARIANTS

-  **Pair Driving**
Two engineers co-drive (one keyboard, one narrator).
-  **Silent Mode**
Observers cannot speak or chat, only write alternatives.
-  **Schema-first**
First rotation is always a diagramming step before dashboards/queries.

TIP

-  With our new **Steadybit MCP Server**, you can simply ask for a scenario to kick things off. Try it out!