

Problem Solving — Workbook

This workbook gives you hands-on exercises, structured worksheets, and ready-to-use templates for every module of the Problem Solving course. Work through each section alongside the lessons to convert frameworks into practiced skills. Every exercise is designed to be completed with a real problem from your current work — not a hypothetical.

Diagnosing the Real Problem

Practice spotting the difference between symptoms and root causes using the 5 Whys and fishbone diagram on a live problem.

Exercise: 5 Whys Chain

Choose one recurring problem from your work — something that has happened at least twice in the last 90 days. Run a full 5 Whys chain. Do not stop until you reach a cause you can directly control through a process or system change.

- State the problem as an observable fact with a measurable impact (e.g., 'Customer onboarding takes 14 days instead of the 5-day target, costing us an estimated 2 clients per quarter').

- Write your 5 Whys chain below. For each answer, ask why again before writing the next. Stop when you reach a systemic cause you can address.

- Does your final root cause point to a person, a process, or a system? If it points to a person, ask one more why.

- What single systemic fix would prevent this problem from recurring? Write it as a specific action with an owner and a deadline.

Worksheet: Fishbone Diagram Worksheet

Use this worksheet to map all possible causes of a complex problem before selecting your top candidates for investigation. Adapt the category labels to fit your context.

Problem statement (the effect at the head of the fish):

Category 1 — People: list contributing causes

Category 2 — Process: list contributing causes

Category 3 — Technology: list contributing causes

Category 4 — Policy: list contributing causes

Category 5 — Environment: list contributing causes

Category 6 — Data: list contributing causes

Top 3 most likely root causes (dot-voted or estimated):

Which root cause will you investigate first, and why?

Checklist: Diagnosis Quality Check

- I described the problem as an observable gap, not a solution or a complaint.
- My 5 Whys chain reached a systemic cause, not a person.
- I identified at least 4 contributing causes across different categories on my fishbone.
- I dot-voted or scored causes to select the top 3 before investigating.
- My proposed fix addresses the root cause, not just the symptom.
- I can measure whether the fix has worked within 30 days.

Framing and Scoping the Challenge

Write a sharp POV statement and SCQA frame for your problem, and stress-test the assumptions behind your current approach with first-principles decomposition.

Exercise: POV Statement Workshop

Using a problem you are currently working on, draft a POV statement using the Stanford d.school formula. Then rewrite it twice — each time shifting the user archetype slightly — to see how the solution space changes.

- Who is the specific person most affected by this problem? Describe them in one sentence (role, context, key constraint) — avoid demographic labels.

- What does this person actually need? State the need as a verb phrase, not a feature or product (e.g., 'to feel confident presenting data to senior leadership', not 'a better dashboard').

- What surprising or non-obvious insight did your empathy work or observation reveal about why this need matters?

- Write your full POV: [User] needs [need] because [insight]. Read it aloud — does it open up the solution space or close it down?

Worksheet: SCQA Scoping Frame

Complete this SCQA frame for a strategic problem you need to present to a leadership team or stakeholder group. The goal is to separate known facts from the real question before proposing a direction.

Situation — what stable background context is universally agreed upon?

Complication — what has changed or what tension has emerged that disrupts the situation?

Question — what is the central question this complication raises? (One sentence, starting with 'How' or 'What' or 'Should')

Answer (hypothesis) — what direction are you proposing to explore? State it as a testable hypothesis, not a final recommendation.

What assumptions are baked into your 'Answer' that you need to validate first?

Exercise: First-Principles Assumption Audit

List 5–8 assumptions embedded in how your organisation currently approaches this problem. Classify each one and identify which to challenge.

- Write out 5–8 beliefs your team holds about this problem or its constraints (e.g., 'We can't change this without executive approval', 'This has always been done this way', 'We don't have the budget').

- For each belief, classify it as: Verified Fact (evidence supports it), Untested Assumption (plausible but not checked), or Explicit Convention (a rule someone made that could be revisited).

- Pick the 2 Untested Assumptions with the highest potential impact if they turned out to be wrong. What is the cheapest way to test each one this week?

Checklist: Framing Readiness Check

My POV statement names a specific user, not a broad demographic.

The need in my POV is a verb (what they need to do or feel), not a product feature.

My SCQA situation contains only facts stakeholders already agree on.

My SCQA complication clearly explains why the status quo is no longer acceptable.

I have identified at least 3 untested assumptions in my current framing.

I have a plan to test the most consequential assumption before committing to a solution.

Generating and Selecting Solutions

Practice structured ideation to expand your solution space, then use a weighted decision matrix to make a defensible selection.

Exercise: Brainwriting 6-3-5 Solo Sprint

Set a timer for 5 minutes and write at least 15 potential solutions to your problem — no filtering, no evaluating. Then apply the SCAMPER lenses to generate 7 more unconventional options.

- Write your 15+ raw ideas below. Quantity is the goal. Include ideas that seem impractical — they often contain the seed of a breakthrough.

- Apply each SCAMPER lens to your best existing solution: Substitute (what component could be replaced?), Combine (what could be merged?), Adapt (what from another domain applies here?), Modify (what could be scaled up or down?), Put to other uses (who else could benefit from this?), Eliminate (what could be removed?), Rearrange (what if the sequence or structure were reversed?).

- After ideation, organise your ideas into three buckets: Quick Win (low effort, positive impact), Strategic Bet (high effort, high impact), Long Shot (unconventional, worth keeping). What does your Long Shot bucket reveal?

Worksheet: Weighted Decision Matrix

Select your top 3–5 candidate solutions and evaluate them against 4–6 criteria that reflect what genuinely matters for this decision. Complete the matrix, then sanity-check the top scorer.

Solution A name and one-line description:

Solution B name and one-line description:

Solution C name and one-line description:

Criterion 1 name and weight (out of 100):

Criterion 2 name and weight (out of 100):

Criterion 3 name and weight (out of 100):

Criterion 4 name and weight (out of 100):

Criterion 5 name and weight (out of 100, optional):

Do all weights sum to 100? (yes/no):

Scores for Solution A (1–5 per criterion) — list each score:

Scores for Solution B (1–5 per criterion) — list each score:

Scores for Solution C (1–5 per criterion) — list each score:

Weighted total for each solution (score × weight, summed):

Does the top-scoring solution feel right? If not, which weight is driving a counterintuitive result?

Checklist: Solution Selection Quality Check

- I generated at least 15 ideas before evaluating any of them.
- My shortlist includes at least one unconventional or Long Shot option.
- My decision criteria are measurable and genuinely differentiate between options.
- I included a reversibility criterion in my matrix.
- All weights sum to 100.
- I ran a sanity check on the top scorer and can explain the result to a sceptical stakeholder.
- I have a plan to run a pre-mortem on the selected solution before committing.

Implementing, Learning, and Preventing Recurrence

Design a rapid test for your chosen solution, run a pre-mortem to surface hidden risks, and build the AAR habit into your team's workflow.

Exercise: Prototype Learning Plan

Before implementing your selected solution, design the cheapest possible test that validates its riskiest assumption. Complete this plan before running the prototype.

- What is the single riskiest assumption in your solution — the one thing that, if wrong, would cause the solution to fail entirely?

-
- What is the cheapest test (paper simulation, wizard-of-oz, concierge, A/B, or tabletop) that could validate this assumption in 5 business days or fewer?

-
- State your success threshold in advance: 'This prototype will succeed if [specific, measurable outcome] is achieved by [date].'
-

- If the prototype fails to meet the threshold, what will you do: iterate, pivot to the second-ranked solution, or reframe the problem?
-

Worksheet: Pre-Mortem Risk Register

Before launching your solution, complete this pre-mortem. Write your failure scenarios individually first, then review them together as a team.

Solution being stress-tested:

Implementation timeline:

Failure scenario 1 — what went wrong and why:

Failure scenario 2 — what went wrong and why:

Failure scenario 3 — what went wrong and why:

Failure scenario 4 — what went wrong and why:

Failure scenario 5 — what went wrong and why:

Top 3 failure modes by probability x impact:

Mitigation action for Risk 1 (owner + deadline):

Mitigation action for Risk 2 (owner + deadline):

Mitigation action for Risk 3 (owner + deadline):

Early-warning indicator for each high-impact low-probability risk:

Exercise: After-Action Review Template Run

Run a 60-minute AAR on a recently completed project or problem-solving initiative using the four standard questions. Use this as a template for every future significant project.

- What was supposed to happen? State the original goal, plan, and success criteria exactly as they were defined at the start — not revised in hindsight.

- What actually happened? Describe the outcome with data. What did you ship, when, and what did the metrics show?

- Why was there a difference? List systemic causes — processes, tools, information gaps, decision points. Resist naming individuals. For each cause, ask: what in the system allowed this to happen?

- What will we do differently next time? Commit to 1–3 specific actions, each with an owner and a 30-day deadline. What is the first step each owner will take this week?

Checklist: Implementation and Learning Checklist

- I ran a pre-mortem before committing to the implementation plan.
- Every top-3 risk has a named mitigation owner and a deadline.
- My prototype had a pre-stated learning goal and success threshold.
- I made a go/pivot/kill decision on the prototype within 5 business days.
- I scheduled the AAR within 48 hours of the project completing.
- The AAR produced 1–3 action items with owners and deadlines.
- The AAR one-pager was shared with the wider team within 24 hours.
- I have updated my personal problem log with any new recurring issues surfaced this week.

Your Action Plan

1. This week: identify one recurring problem from your work and run a full 5 Whys chain to find the root cause — not a symptom.
2. This week: write a POV statement for the same problem using the Stanford d.school formula; share it with one colleague for a reaction.
3. Within 10 days: map the problem on a fishbone diagram with at least 4 cause categories populated; dot-vote the top 3 root causes.
4. Within 2 weeks: list all assumptions behind your current approach, classify each as Verified Fact, Untested Assumption, or Explicit Convention, and design a cheap test for the most consequential untested assumption.
5. Within 2 weeks: run a structured ideation session (brainwriting or SCAMPER) and produce at least 15 candidate solutions before evaluating any.
6. Within 3 weeks: build a weighted decision matrix for your top 3–5 solutions with explicit criteria and weights; review the result with a stakeholder.
7. Before any significant implementation: run a pre-mortem session with your core team and produce a risk register with mitigations.
8. For your next project: design and run a prototype or pilot to validate the riskiest assumption before committing full resources.
9. After your next significant project: facilitate a 60-minute AAR using the four questions and publish a one-page summary within 24 hours.
10. Monthly: review your problem log, flag any issue that has recurred 3+ times, and bring a structured problem statement to your team as a proactive improvement proposal.

