

Introduction to Problem Solving Techniques

Introduction to Problem Solving Techniques

Designed as a quick-reference guide for students, professionals, and anyone who wants a systematic approach to tackling problems.

1. What Is Problem Solving?

Aspect	Description
Definition	A purposeful, organized process for finding solutions to obstacles or achieving desired outcomes.
Goal	Convert an ill-defined situation into a clear, actionable plan and implement it successfully.
Key Attributes	Logical, iterative, creative, and often collaborative.

2. General Problem Solving Framework

1. Identify & Define the Problem

- Clarify *what* is happening, *where*, *when*, and *who* is affected.
- Write a concise problem statement (1–2 sentences).

2. Gather Information

- Collect data, facts, constraints, and stakeholder perspectives.
- Use sources: observations, interviews, documents, metrics.

3. Analyze the Problem

- Break it into sub-problems or root causes.
- Tools: **5 Whys**, **Fishbone (Ishikawa) diagram**, **Pareto analysis**.

4. Generate Options (Ideation)

- Brainstorm without judging.
- Encourage divergent thinking: “What if...?” “How might we...?”

Introduction to Problem Solving Techniques

5. Evaluate & Select Solutions

- Apply criteria: feasibility, impact, cost, time, risk.
- Use decision-making aids: **Decision matrix**, **Cost-Benefit analysis**, **SWOT**.

6. Plan Implementation

- Define steps, responsibilities, resources, timeline.
- Create a **Gantt chart** or **Kanban board** for visual tracking.

7. Execute & Monitor

- Follow the plan, collect real-time data, adjust as needed.
- Use **PDCA (Plan-Do-Check-Act)** cycles for continuous improvement.

8. Review & Document

- Assess outcomes against objectives.
- Capture lessons learned for future reference.

3. Core Techniques & When to Use Them

Technique	Core Idea	Typical Use Cases	Quick Steps
Trial-and-Error	Test possible solutions, learn from failures.	Simple, low-risk problems; hardware debugging.	1. Propose a solution ? 2. Test ? 3. Observe ? 4. Refine.
Algorithmic/Procedural	Follow a fixed sequence of steps.	Repetitive tasks, programming, calculations.	1. Define input ? 2. Apply steps ? 3. Produce output.
Heuristics	Rule-of-thumb shortcuts that work in most cases.	Time-pressured decisions, search problems.	Example: "Pick the most promising option first."
Divide & Conquer	Split a large problem into independent sub-problems.	Complex software, large projects, mathematical proofs.	1. Partition ? 2. Solve sub-problems ? 3. Combine results.

Introduction to Problem Solving Techniques

Technique	Core Idea	Typical Use Cases	Quick Steps
Root Cause Analysis	Identify underlying cause(s) rather than symptoms.	Recurring defects, process failures.	Use 5 Whys or Fishbone diagram.
Brainstorming & Mind Mapping	Generate many ideas rapidly; visualize relationships.	Creative design, marketing strategies.	Write central problem ? branch ideas outward.
Decision Matrix (Weighted Scoring)	Quantify options against weighted criteria.	Choosing vendors, selecting	