Business Strategy / SOP Creation

Create if-then decision guides for situations with multiple paths, exceptions, or conditional logic.
Difficulty: Intermediate → Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Conditional Processes, Exception Handling, Troubleshooting Guides
Updated: May 2026
Why This Prompt Exists
Most SOPs break when exceptions occur — they assume everything goes right.

You get:

- linear steps that don’t handle exceptions
- no guidance for decision points
- employees making inconsistent decisions
- escalations to managers for simple decisions
- errors from wrong conditional handling

But a decision tree is not a linear process.

It is a map of possible paths.

- Decision points: where a choice is made
- Conditions: what determines the path
- Actions: what to do based on condition
- Escalation: when to involve a manager
- Default path: what to do if no condition matches

Without decision trees, employees guess.

This framework forces AI to build decision trees that guide conditional processes.

The Prompt

```
Assume the role of a process designer who creates decision trees for conditional processes.

Your task is to create a decision tree SOP.

Generate:

1. PROCESS NAME AND GOAL

2. STARTING POINT
   - What triggers the decision process

3. DECISION TREE (text-based or bulleted)
   - IF [condition] THEN [action]
   - ELSE IF [condition] THEN [action]
   - ELSE [default action]

4. DECISION POINTS MAPPED
   - Key questions to ask
   - Possible answers for each

5. ESCALATION PATHS
   - When to escalate to manager
   - Who to escalate to
```

6. DEFAULT PATH
   - What to do if no condition matches

7. DECISION TREE VISUALIZATION (text-based)
   - Indented tree structure

INPUTS:

Process Name:
[INSERT]

Decision Triggers (what situations require decisions):
[LIST]

Possible Outcomes (paths):
[LIST]

Known Exceptions or Edge Cases:
[LIST]

Authority Levels (who can make which decisions):
[DESCRIBE]

RULES:
- Decision points must be clear (yes/no or multiple choice)
- Conditions must be objective (not "if customer seems upset")
- Actions must be specific (not "handle appropriately")
- Escalation paths: when and to whom
- Default path: catch-all for unhandled conditions

```
- Test decision tree with real scenarios before publishing
```

How To Use It

- Decision trees work best for troubleshooting and customer support.
- Conditions must be objective and verifiable.
- Escalation paths prevent employees from making out-of-authority decisions.
- Default path catches edge cases.
- Test decision tree with real scenarios before publishing.

Example Input

Process Name: Customer Refund Request Handling

Decision Triggers: Customer requests refund, product defective, wrong item shipped, customer changed mind, outside return window

Possible Outcomes: Approve full refund, Approve partial refund, Offer store credit, Deny refund, Escalate to manager

Known Exceptions: VIP customer (high lifetime value), product damage caused by customer, expired warranty

Authority Levels: Support agent ($0-50), Team lead ($51-200), Manager ($200+ or exceptions)

Why It Works
Most SOPs break on exceptions.

This framework improves outcomes by forcing:

- decision point identification (clarity)
- condition-action mapping (logic)

- escalation paths (authority boundaries)
- default path (edge case handling)
- visual representation (comprehension)

Great decision trees don’t guess — they guide, condition by condition.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI business strategy tools, SOP frameworks, and practical strategies for leaders and operators.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also The Checklist-Style SOP Creator