

# AI Foresight Copilot – Product Development and Validation Case Study

<b>Project Overview</b>	<b>1</b>
<b>Project Objective</b>	<b>1</b>
<b>Phase 1 — Problem Framing and Hypothesis</b>	<b>2</b>
1.1 Problem statement	2
1.2 Primary hypothesis	2
1.3 Target users (initial)	2
<b>Phase 2 — Discovery and Market Scan</b>	<b>2</b>
2.1 Desk research (market landscape)	2
2.2 Benchmarking (feature-level)	3
2.3 Value proposition (positioning)	3
<b>Phase 3 — User Needs and Validation Plan</b>	<b>3</b>
3.1 Jobs-to-be-done (JTBD)	3
3.2 Success criteria (validation metrics)	3
3.3 Lean validation approach	4
<b>Phase 4 — Solution Design (Method + Product)</b>	<b>4</b>
4.1 Define the foresight workflow	4
4.2 Product decisions (MVP constraints)	4
<b>Phase 5 — Prototyping and Build (No-code Architecture)</b>	<b>5</b>
5.1 Architecture (end-to-end)	5
5.2 Why multi-agent instead of one prompt	5
5.3 Reliability and error handling	5
<b>Phase 6 — Testing, Iteration, and Quality Improvement</b>	<b>5</b>
6.1 Test cases	5
6.2 Quality iteration loop	5
6.3 “Consulting realism” improvements	6
<b>Phase 7 — Go-to-Market Thought Process (MVP)</b>	<b>6</b>
7.1 Primary use cases	6
7.2 Distribution and acquisition experiments	6
7.3 Pricing hypothesis (future)	6

# Project Overview

Organizations across industries are facing increasing strategic uncertainty driven by rapid technological change, particularly the emergence of artificial intelligence, evolving regulatory environments, and shifting competitive dynamics. Traditional strategic foresight and scenario planning approaches can help organizations navigate uncertainty, but they are typically resource-intensive, requiring expert facilitation, workshops, and significant analytical effort.

As a result, many organizations — especially smaller companies, innovation teams, and academic institutions — lack accessible tools to systematically explore potential future scenarios and prepare strategic responses.

This project explores whether generative AI and workflow automation can be used to create a lightweight, accessible foresight tool that supports early-stage strategic thinking.

**AI Foresight Copilot** was developed as a prototype decision-support system that allows users to input an industry context, a future time horizon, and a strategic decision question. The system then generates a structured foresight report including emerging trends, alternative future scenarios, and potential strategic responses.

The goal of the project was to design and validate a minimal viable product capable of translating established scenario planning methodologies into an automated and accessible digital tool.

## Project Objective

The objective of this project was to investigate whether generative AI can be used to democratize access to strategic foresight by automating parts of the scenario planning process traditionally conducted by consultants and strategy teams.

Specifically, the project aimed to:

- Identify an industry need for accessible foresight tools that support strategic decision-making under uncertainty.
- Design a lightweight product concept that operationalizes key steps of the foresight process (trend identification, uncertainty mapping, scenario construction, and strategic implications).
- Develop a working prototype that demonstrates how AI agents and workflow automation can generate structured foresight outputs.

- Validate the usability and perceived value of the tool through pilot testing with potential users.

## Phase 1 — Problem Framing and Hypothesis

### 1.1 Problem statement

Organizations face rising uncertainty (AI, regulation, labor markets, geopolitics). Traditional scenario planning is valuable but:

- time-intensive (workshops + expert facilitation)
- expensive (consulting engagements)
- inaccessible for smaller organizations and teams

### 1.2 Primary hypothesis

A “minimum viable foresight” tool can deliver value if it:

- produces a coherent scenario structure (2x2 uncertainties + 4 scenarios)
- creates actionable strategy outputs (no-regrets, bets, hedges, 90-day actions)
- is simple enough to use in <2 minutes
- costs near-zero to run and requires no complex onboarding

### 1.3 Target users (initial)

- innovation managers / strategy teams in SMEs
- consultants needing quick scenario drafts
- academic leaders exploring AI strategy (universities)
- founders exploring market uncertainty

## Phase 2 — Discovery and Market Scan

### 2.1 Desk research (market landscape)

I mapped three adjacent solution categories:

1. **Strategy/foresight consultancies** (high-touch, expensive, slow cycle)

2. **Knowledge and insight tools** (trend reports, research databases, newsletters)
3. **AI assistants** (general-purpose tools that lack structured foresight methodology)

Key gap observed: AI tools generate content, but few enforce **foresight structure** (weak signals → trends → uncertainties → scenarios → strategy).

## 2.2 Benchmarking (feature-level)

I defined what “good” looks like based on scenario planning practice:

- clear decision question framing
- explicit time horizon
- trend list linked to drivers
- uncertainties derived from tensions in trends
- scenarios that differ meaningfully (not cosmetic variations)
- implications translated into strategic options

This informed the design requirements for prompts, outputs, and report format.

## 2.3 Value proposition (positioning)

**“Foresight in minutes”**: A structured scenario planning output without a workshop.

Not a replacement for consulting—an accelerator for early thinking and internal discussion.

# Phase 3 — User Needs and Validation Plan

## 3.1 Jobs-to-be-done (JTBD)

- “Help me explore uncertainty so I can make a better strategic decision.”
- “Give me scenario options I can reuse in a deck or workshop.”
- “Help me articulate strategic moves under different futures.”

## 3.2 Success criteria (validation metrics)

For a first prototype, success was defined as:

- users understand the output without guidance
- scenarios feel plausible and differentiated
- strategy section is actionable (not generic)
- delivery is reliable (email arrives + formatted correctly)

- total experience feels “consulting-grade”

### 3.3 Lean validation approach

Because I built solo, I used a rapid approach:

- 5–10 pilot users (consulting/innovation/academia)
- 15-minute structured feedback sessions
- simple feedback form: usefulness, clarity, realism, what they would change
- iterate prompts and report format after each batch

## Phase 4 — Solution Design (Method + Product)

### 4.1 Define the foresight workflow

I translated a classic foresight process into a lightweight pipeline:

1. **Input framing**  
Industry + time horizon + decision question
2. **Weak signal scan (AI-generated in MVP)**  
Generate potential weak signals relevant to the domain
3. **Trend synthesis**  
Cluster weak signals into 6 emerging trends with drivers + uncertainty rating
4. **Scenario framework**  
Derive 2 critical uncertainties from tensions within trends, define poles
5. **Four scenarios**  
Generate 2x2 scenarios with differentiated narratives + implications
6. **Strategy & action**  
No-regrets / bets / hedges + 90-day plan
7. **Report output**  
Clean HTML report emailed to user (fast distribution, no login)

### 4.2 Product decisions (MVP constraints)

I optimized for:

- **speed to test** (no-login, email delivery)
- **zero-cost stack** (Google Forms/Sheets + Make)
- **repeatable structure** (JSON outputs to reduce randomness)
- **portfolio demonstration** (clear artifacts to show product thinking)

# Phase 5 — Prototyping and Build (No-code Architecture)

## 5.1 Architecture (end-to-end)

- **Google Form:** input collection (industry, horizon, decision question, email)
- **Google Sheets:** response storage (trigger source)
- **Make:** orchestration + error handling + email sending
- **OpenAI modules:** multi-step agent flow (signals → trends → scenarios → strategy → report writer)

## 5.2 Why multi-agent instead of one prompt

To increase reliability and consulting structure:

- each module returns **valid JSON** with strict schemas
- downstream modules reuse upstream outputs
- the report writer converts structured results to client-ready language

## 5.3 Reliability and error handling

I added:

- explicit JSON schema constraints
- fallback handling (if a module fails, the report still indicates what is missing)
- email body fallback (`ifempty(...)`) to avoid blank emails
- shorter report structure to avoid token truncation

# Phase 6 — Testing, Iteration, and Quality Improvement

## 6.1 Test cases

I tested across varied domains to validate generalizability:

- AI in universities (policy + pedagogy + integrity)
- HR tech (regulation + data + adoption)
- management consulting (AI disruption + business model change)

## 6.2 Quality iteration loop

I ran an iterative loop:

1. run scenario with real decision question
2. assess output against success criteria:
  - are uncertainties derived from trends?
  - do scenarios differ structurally?
  - are implications concrete?
  - are strategic moves distinct categories?
3. adjust prompts using “smallest change” principle
4. re-test with same input to measure improvement

### **6.3 “Consulting realism” improvements**

Key prompt refinements:

- force uncertainties to be derived from tensions in trends (reduces generic axes)
- cap bullet lengths and enforce counts (improves clarity and speed)
- require scenario naming + quadrant label mapping (improves usability)
- avoid raw JSON in output (client-ready report formatting)

## **Phase 7 — Go-to-Market Thought Process (MVP)**

### **7.1 Primary use cases**

- consultants: generate a first scenario draft before workshops
- innovation teams: quick uncertainty scan before strategy sessions
- academic leaders: explore AI governance and future university models

### **7.2 Distribution and acquisition experiments**

Low-effort acquisition experiments:

- embed form on SVFL website (Resources page)
- share tool link in LinkedIn posts (thought leadership + demo)
- invite 5–10 testers via personal network (qualitative feedback)

### **7.3 Pricing hypothesis (future)**

Early options considered:

- free basic report + paid “extended report”
- monthly subscription for teams
- “report credits” model (pay-per-report)

