



CASE STUDY

Case Study: How to Have an Investor Pitch Deck in Real Time, in Real Time, Updated

The technical walkthrough you asked for

24 March 2026

Step 0 - What we actually have

A Git repository with slides

```
slides/  
  Pitch-decks/  
    Investor-pitch-deck.md    <- Marp markdown, managed by the agent and us  
    Investor-pitch-deck.pptx  <- Auto-generated by Marp  
    Investor-pitch-deck.pdf   <- Auto-generated by Marp  
    Investor-pitch-deck.html  <- Auto-generated by Marp
```

The pitch deck is plain Markdown. It lives in Git. It is **diffable**, **reviewable**, and **regenerable**.

Both us and the agent can update it. We can see the history of changes. We can revert if something breaks.

NO MORE

- No Google Slides drift
- No PowerPoint binary blobs
- No `v7_final_FINAL.pptx`

Step 1 - The Agent

We write a `.book` file — a persistent instruction set for a [Promptbook agent](#).

This is **not a one-shot prompt**. It is a standing agent definition that runs on a schedule.

WHY A .BOOK FILE?

- Behavior is defined **once**, runs **forever**
- Sources are explicit and auditable
- Agent output is bounded — no hallucination
- Human reviews diffs, not entire decks

Pitch deck agent

```
PERSONA You are a professional assistant
GOAL Keep pitch decks up-to-date
RULE Slides are under slides/Pitch-decks/*
RULE Use Marp and commit to main
KNOWLEDGE https://www.ptbk.io/
KNOWLEDGE https://www.ptbk.io/pro-mesta
KNOWLEDGE https://github.com/webgptorg/promptbook
```

```
USE BROWSER To find recent information
USE SEARCH ENGINE For recent context
USE PROJECT https://github.com/webgptorg/slides
USE TIMEOUT Regenerate once a day
```

Step 2 - How the Agent Works

The agent is written in [Promptbook ecosystem](#) which can handle AI grounding, RAG, and rules enforcement...

The agent does not hallucinate content. It reads concrete URLs.

Directive	What it does
KNOWLEDGE https://www.ptbk.io/	Indexes the main product site - positioning, features, pricing
KNOWLEDGE https://www.ptbk.io/pro-mesta	Indexes the public-sector landing page
KNOWLEDGE https://github.com/...promptbook	Reads the repo - README, package versions, changelog
USE BROWSER	Opens pages and renders JS if needed
USE SEARCH ENGINE	Searches the web for recent mentions, ecosystem context
USE PROJECT ...slides	Has write access to the slides repo - reads old decks, commits new ones

The agent output is bounded by what these sources actually say — no hallucination, no drift.

Step 3 - The agent runs

ON EACH RUN THE AGENT:

1. Fetches all `KNOWLEDGE` URLs
2. Reads the current repo state via `USE PROJECT`
3. Optionally searches the web via `USE SEARCH ENGINE`
4. Generates a complete Marp Markdown deck
5. Commits it in `main`
6. The conversion script exports HTML, PDF, PPTX

WHAT TRIGGERS A RUN

- Agent sets itself to run once a day via `USE TIMEOUT`
- Manual trigger before an investor meeting
- On-demand via the Book initial message:

"Create me a pitch deck about Promptbook and keep it up to date!"

Every run is a fresh, sourced, versioned deck — no manual drafting required.

Step 4 - Human stays in the loop

The agent drafts. The human reviews.

AGENT RESPONSIBILITY

- Collect facts from approved sources
- Generate a new dated deck
- Commit to Git

HUMAN RESPONSIBILITY

- Define and approve **KNOWLEDGE** sources
- Review the changes regularly (*diffs in Git*)
- Decide which version to send to the investor

Git is the audit trail. The agent is the author. You are the editor.

Why this is not "just prompting"

ONE-SHOT PROMPTING

- you write a prompt, get a deck once
- next week it is outdated
- you prompt again, diff manually
- no version history, no audit trail

BOOK-BASED AGENT

- the Book defines behavior permanently
- the agent re-runs on schedule
- output is versioned in Git automatically
- sources are explicit and auditable
- you review diffs, not entire decks

The difference is operational: one-shot is a tool, a Book is a process.

Set this up for your own company

OPTION A — MANAGED

We deploy a Promptbook Agent Server for you.

We configure the Book, connect your knowledge sources, and set up the daily schedule. You focus on reviewing the diffs before the meeting.

OPTION B — SELF-HOSTED

Run it yourself from GitHub.

github.com/webgptorg/promptbook

Write your Book, deploy to your own infrastructure, own your data.



ptbk.io

Scan to open · Start for free · 24 March 2026