Lesson Review and Index: Product Management Lessons 1 and 2

This index covers the two reviewed lessons, the two capstone case studies, and the recommended learning order. It links each topic to the guide that teaches it in depth. These lessons did not cover RAG, Replit or Nginx in depth, so those topics have their own guides rather than being forced in here.

1. Lesson 1: AI product costing and an n8n onboarding workflow

- **Define the job** of the AI feature: tone, length, audience. Send useful variables with each request.
- **Unit cost:** calculate cost from input and output tokens, then multiply by expected volume. Compare conservative, realistic and aggressive scenarios before deciding to charge, give a free allowance, or absorb the cost. *Guide: Tokens, context and cost.*
- **Define input, output, owner and cost** before using an LLM in any feature.
- **Employee onboarding:** an HR form captures hire ID, candidate name, email, department, role, manager and start date. Reference data sits in an access registry and a scheduling registry.
- **Three agents:** a Coordinator routes and tracks work; an Access agent decides and records access; a Scheduling agent handles meetings. *Guide: n8n multi-agent onboarding.*
- **Webhooks:** a doorbell. The workflow waits for an event instead of checking repeatedly. Use the test URL while building, then the production URL once active. Include a unique `hire_id` as the primary key.
- **Credentials and tools:** never connect a personal account to a practice automation. Consider permissions before connecting any tool. Build and test one small step at a time.

2. Lesson 2: Make, MCP and case-study thinking

- **Scenarios** are repeatable automations. Variables and settings let one flow serve many inputs. Check credit use, notifications and failure handling before you enable one. *Guide: Make.com.*
- **MCP** (Model Context Protocol) is a standard way for AI applications to connect to outside systems. A **tool** is a capability an agent can call. *Guide: Skills, subagents and agents.*
- **Repeated assignment expectations:** understand the user and the real problem first; define the goal and a success metric; split features into v1, v2 and cut; define the core loop, AI

strategy and feedback; create journey maps, a PRD and three key wireframes; measure events on a dashboard.

3. Capstone: Fireflies, meeting commitments

Part	Content
Problem	Commitments made in meetings show up in summaries but are not reliably assigned, owned, tracked or closed. The class figure: only 28 percent of extracted commitments reached a resolved state within 14 days.
User	Meeting attendees who owe or receive a follow-up.
Goal	Make every commitment visible, owned and resolvable, without becoming a full project-management tool.
Resolution states	A commitment is resolved only when it is **done**, **reassigned**, or **deliberately dropped**. Each state is explicit.
Success metric	Share of commitments resolved within 14 days (the baseline is 28 percent).
v1	Detect commitments from a transcript; show owner and due date; three resolution states.
v2	Reminders; reassignment suggestions; owner summaries.
Cut	Full project boards, task dependencies, time tracking.
Core loop	Meeting → commitment extracted → owner confirms → reminder → resolved state recorded.
AI strategy	Extraction with a confidence score; the owner always confirms before a commitment is saved.
Feedback design	One-tap confirm, edit or reject on each extracted commitment, so the model learns what matters to each team.
Boundaries	No full task management; no changes to the meeting platform itself.
Events	commitment_detected, owner_confirmed, reminder_sent, resolved_done, resolved_reassigned, resolved_dropped
Wireframe 1	Meeting summary with a list of detected commitments, each with owner and due date, and confirm, edit or reject buttons.
Wireframe 2	My commitments: list grouped by due date, with the three resolution states as clear buttons.
Wireframe 3	Weekly summary: open, resolved and dropped counts, plus a link to any item that is overdue.

4. Capstone: ixigo, the confirmed-seat problem

Part	Content
Problem	A traveller knows the origin and destination but has no confirmed seat on the preferred train or class.
User segments	Last-minute travellers; price-sensitive planners; travellers who must arrive by a fixed time.
Goal	Help the user discover, compare and choose viable alternatives, and understand each option honestly.
Decision factors	Price, travel time, arrival time, station convenience, confirmation probability and cancellation flexibility.
Uncertainty messaging	Show likelihood as a range, never a guarantee. Say what is unknown and why.
v1	Compare alternative trains and classes for the same route and date; show the six decision factors; show uncertainty.
v2	Nearby stations and dates; bus and flight options where data is available.
Cut	Redesigning rail inventory, timetables, quota rules, fares, payments, refunds, or support staffing.
Success metric	Share of users who choose an option and complete the journey without leaving to search again.
Core loop	Search → see viable options → compare on the six factors → choose → track the decision.
AI strategy	Rank options and explain trade-offs in plain language; never invent availability.
Feedback design	After each choice, ask whether the option was right, and log the reason.
Events	search_started, option_viewed, option_chosen, uncertainty_shown, option_changed_later
Wireframe 1	Results list with six factor columns and a clear confidence range per option.
Wireframe 2	Option detail: trade-offs in plain language, and cancellation terms.
Wireframe 3	Confirmation: choice saved, with a reminder of what remains uncertain.

5. The nine-step case-study framework

1. Problem framing
2. User research

3. Goal
4. Success metric
5. Scope (v1, v2, cut)
6. User flow and journey map
7. PRD
8. Wireframes
9. Metrics and event taxonomy (plus an optional experiment plan)

6. Recommended learning order

1. Tokens, context and cost (LLM basics and cost).
2. RAG and retrieval.
3. Skills, subagents and agents.
4. Webhooks, JSON and test versus production.
5. One n8n workflow with fake data.
6. Credentials, permissions and test accounts.
7. Make scenarios and MCP.
8. App building with Replit or Lovable.
9. One capstone, using the nine-step framework.

7. Trusted next steps

- Build one safe n8n webhook demo with fake data and a practice account.
- Learn MCP from its introduction, then the specification when you build.
- For Make, learn scenarios first, then MCP with the smallest permissions.
- Review any connector's permissions before you connect it.

Official sources: check current documentation for n8n, Make, Lovable, Replit and MCP before relying on a feature. Prices change; check the live pricing page on the day you decide.

8. Fireflies case study, in full

Original wording and original diagrams. The case: meetings produce commitments, but many are never owned, tracked or closed.

8.1 Problem and user

Item	Detail

Problem	Commitments made in meetings appear in summaries, but they are not reliably assigned, tracked or closed. The class figure: only 28 percent of extracted commitments reach a resolved state within 14 days.
Primary user	A meeting attendee who owes a follow-up, or who is waiting for one
Secondary user	A team lead who wants to know what is open
Not the user	Project managers who need full task boards (out of scope)
Goal	Every commitment is visible, owned and explicitly resolved, without becoming a full project-management tool

8.2 The resolution model

A commitment is resolved only in one of three ways, and each is explicit:

State	Meaning	Who sets it
Done	The work was completed	The owner
Reassigned	Someone else now owns it	The owner or the lead, with the new owner's name
Deliberately dropped	Someone chose not to do it, and said why	The owner or the lead, with a reason

Anything else is open. A commitment that is silently ignored stays visible as open, which is the problem this product exists to solve.

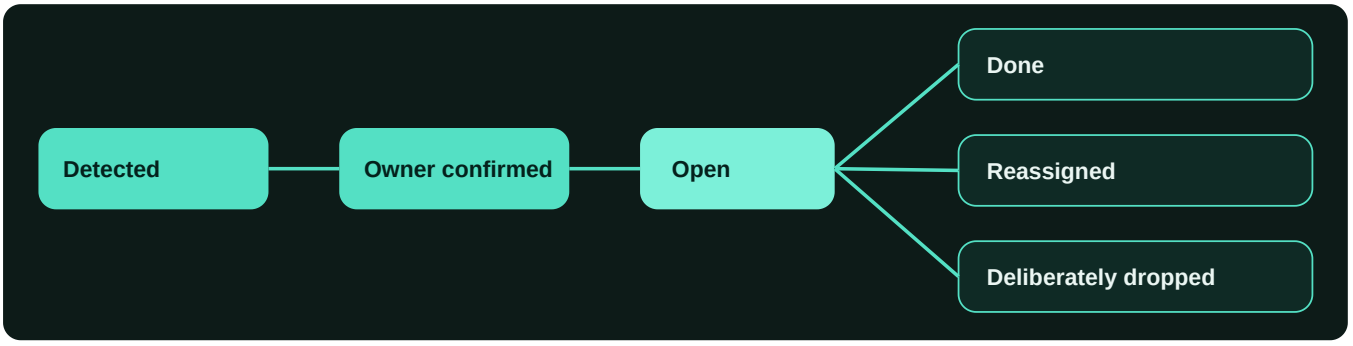


Figure 1. The commitment life cycle. Only three states end it.

8.3 Core loop, success metric and scope

Part	Detail
Core loop	Meeting ends, commitments are extracted, owner confirms, reminder is sent, commitment is resolved

Success metric	Share of commitments that reach a resolved state within 14 days (baseline: 28 percent)
Guardrail metrics	Share of extracted items the owner rejects (extraction quality); reminder opt-out rate
v1	Extract commitments from a transcript; show owner and due date; three resolution states; one reminder
v2	Reassignment suggestions; weekly lead summary; recurring-meeting carry-over
Cut	Task dependencies, boards, time tracking, full project management

8.4 AI strategy and feedback

- **Extraction:** the model proposes commitments with a confidence score. Low-confidence items are shown with a question mark, not hidden.
- **Owner confirmation:** no commitment is saved until its owner confirms, edits or rejects it. This protects trust.
- **Feedback loop:** every confirm, edit or reject is logged, so extraction quality can be measured and improved.
- **Honest limits:** the product says when it is unsure, and never invents an owner or a date.

8.5 Journey map

Stage	User does	Feels	Opportunity
Before	Joins the meeting	Focused on the discussion	Nothing to do
During	Agrees to a follow-up	Confident they will remember	Capture the commitment quietly
Right after	Sees the summary	Busy; the summary is long	A short list of "your commitments" with confirm and edit
Days later	Gets a reminder	Mild pressure	One-tap done, reassign or drop
Week end	Lead checks open items	Wants a clear view	Weekly summary of open, resolved and dropped

8.6 PRD outline

1. Problem statement and evidence (the 28 percent figure and how it was measured)
2. Users and jobs to be done
3. Goals, success metric and guardrails
4. Scope: v1, v2 and cut

5. User flows: detect, confirm, remind, resolve
6. Requirements for extraction, ownership, states and reminders
7. AI behaviour: confidence, abstention and feedback
8. Privacy: who can see which commitments
9. Risks and open questions
10. Launch plan and experiments

8.7 Three wireframes (described)

Screen	Contents
1. After-meeting list	Title and date; a list of detected commitments, each with owner, due date and Confirm, Edit, Reject buttons; a note on how sure the system is
2. My commitments	Grouped by due date; each row has Done, Reassign and Drop buttons; dropping asks for a reason
3. Weekly summary	Counts of open, resolved and dropped; a list of overdue items; a link to each meeting

8.8 Event taxonomy

Event	When	Key properties
commitment_detected	The system extracts an item	meeting_id, confidence
owner_confirmed / edited / rejected	The owner responds	commitment_id, change made
reminder_sent	A reminder goes out	commitment_id, days_open
resolved_done / reassigned / dropped	The commitment ends	commitment_id, reason (for dropped)
summary_viewed	A lead opens the weekly summary	team_id

8.9 Experiments

Question	Test	Measure
Does a reminder after two days beat one after five?	A/B on reminder timing	Resolved within 14 days
Does showing confidence reduce rejections?	A/B with and without the note	Rejection rate and confirmation speed
Does one-tap resolution increase completion?	A/B on button layout	Time to resolve

9. ixigo case study, in full

The case: a traveller knows the origin and destination but does not have a confirmed seat on the preferred train or class.

9.1 Problem, users and boundaries

Item	Detail
Problem	The preferred train or class has no confirmed seat, and the traveller does not know which alternative to choose
Users	Last-minute travellers; price-sensitive planners; travellers with a fixed arrival time
Goal	Help the user discover, compare, understand and choose a viable alternative
Out of scope	Redesigning rail inventory, timetables, quota rules, fares, payments, refunds, or customer-support staffing

9.2 Decision factors

Factor	Why it matters	How to show it
Price	Total cost to the traveller	One number, with the class shown
Travel time	How long the journey takes	Hours and minutes
Arrival time	Whether it fits the plan	Local time, with a next-day marker
Station convenience	Distance to the destination or origin	Station name and a short note
Confirmation likelihood	How likely the seat is to be confirmed	A range with plain words, never a guarantee
Cancellation flexibility	How easy it is to change plans	A short, honest summary of the terms

9.3 Uncertainty messaging

- Show likelihood as a range with words such as "likely" and "unlikely", and say what the estimate is based on.
- Never present a prediction as a promise.
- Say what is unknown, and when the information was last updated.

9.4 Core loop, metric and scope

Part	Detail
Core loop	Search, see viable options, compare on six factors, choose, track the decision

Success metric	Share of users who choose an option and complete the journey without leaving to search again
Guardrails	Complaints about confirmation outcomes; users who say the estimate misled them
v1	Alternatives on the same route and date; the six factors; honest uncertainty
v2	Nearby stations and dates; bus and flight options where data exists
Cut	Everything listed as out of scope

9.5 AI strategy and feedback

- Rank options and explain the trade-off in plain language ("cheaper by 300 rupees, arrives two hours later").
- Use real availability data only; never invent a seat or a time.
- After the journey, ask whether the choice was right, and log the reason.

9.6 Journey map

Stage	User does	Feels	Opportunity
Search	Looks for the preferred train	Hopeful	Show the result fast
No seat	Sees waitlist or unavailable	Anxious	Offer clear alternatives at once
Compare	Weighs options	Overloaded	Six factors, sorted by what matters to this user
Choose	Picks one	Wants reassurance	Honest uncertainty, clear next step
After	Travels or changes plans	Relieved or frustrated	Ask if the choice was right

9.7 PRD outline

1. Problem and evidence
2. Users and situations
3. Goals, success metric and guardrails
4. Scope: v1, v2, cut, and explicit out-of-scope list
5. Flows and decision factors
6. Uncertainty and wording rules
7. Data sources and refresh times
8. Risks, edge cases and open questions
9. Experiments and launch

9.8 Three wireframes (described)

Screen	Contents
1. Options list	Alternatives sorted by the user's priority; six factor columns; a clear confidence range for each
2. Option detail	Plain-language trade-offs; cancellation terms; when the data was last updated
3. Decision saved	Chosen option; a reminder of what remains uncertain; a way to change the choice

9.9 Event taxonomy and experiments

Event	When
search_started	The user searches
no_seat_shown	The preferred option has no confirmed seat
option_viewed	The user opens an alternative
uncertainty_shown	The range is displayed
option_chosen	The user picks one
option_changed_later	The user changes the choice

Question	Test
Does sorting by the user's top priority raise choice rate?	A/B on sort order
Does a plain-language range reduce complaints?	A/B on wording

10. Practice: build your own case study

1. Pick a product you use. Write the problem in one sentence with one fact you can check.
2. Name the user and what they are trying to do.
3. State the goal and one success metric with a baseline.
4. Split features into v1, v2 and cut. Write what you will not do.
5. Draw the core loop in five boxes.
6. Describe the AI part, its limits, and how users can correct it.
7. Write the journey map in five stages.
8. Outline the PRD, then describe three screens.
9. List six events and two experiments.

Pass when: a teammate can read your pages and say what to build first, what to leave out, and how you will know it worked.

11. Interview questions on these cases (answers follow)

1. How would you define "resolved" for a meeting commitment, and why?
2. What is the biggest risk of automatic extraction, and how do you handle it?
3. How would you tell a traveller about uncertainty without scaring them?
4. What would you cut from v1 of the ixigo product, and why?
5. Which two metrics would you watch in the first month?

Answers: 1. Done, reassigned or deliberately dropped, because each is a clear decision and silence is not. 2. Wrong or invented items destroy trust; require owner confirmation and show confidence. 3. Use a range with plain words, say what it is based on, and give a clear next step. 4. Everything outside choosing an alternative: fares, payments and refunds, because they are separate problems. 5. For Fireflies, resolved within 14 days and owner rejection rate; for ixigo, completed journeys without re-search and complaints about misleading estimates.