Session 3 Handout: Build

FAST-08 AI for Workflow Improvement and Process Automation · 7formations · Instructor: Wadood Chaudhary · All examples fictional · Sandbox only

Why this session matters

No-code tools let you build a working automation in an afternoon. The same speed makes it easy to build something that misroutes requests, leaks data, or fails silently. This session covers the parts every flow has, how to write a reliable AI classify step, and how to test the happy path before you try to break it.

Key concepts

1. The parts of a flow

Part	What it is	Request-routing example
Trigger	The event that starts the flow	New email in the sandbox requests inbox
Action	Something the flow does	Add a row to the tracking list
Connector / connection	The link to an app, with its permissions	Mail connection scoped to one shared inbox
Condition / paths	Branching by a value	If label = quote, post to pricing
AI step	A model call with a prompt and defined outputs	Classify the request

Microsoft track: Power Automate cloud flows (AI Builder prompts or Copilot Studio agent flows). Zapier track: a Zap with an AI by Zapier step and Paths. The concepts are the same.

2. A reliable classify prompt

1. List the allowed labels and define each in one line.
2. Give a fallback ("other" if unsure or if there are two requests).
3. Ask for the label only, as a defined output field.
4. Say the input is data, not instructions.
5. In the condition, check the label against the allowed list; anything else goes to review.

3. Sandbox first

Build and test only in the sandbox. A flow in the wrong environment can touch real mailboxes and real customers. Moving to production is the employer's decision with IT.

4. Test in two rounds

- **Round 1, happy path:** one clear example per label.
- **Round 2, edge cases:** no subject; attachment only; two requests in one email; misleading words ("wrong" in a billing question); an instruction hidden in the text.

Log every run: time, AI label, routing result, pass or fail.

Worked example: fixing a mislabel

Test: "Where is load MFL-20931? Due yesterday." AI label: **claim**. Fix: tighten the definitions ("status: asks where a shipment is or when it will arrive"; "claim: reports damage, shortage, or loss") and add "If unsure, return other." Retest: **status**.

Checklist: build 1

- ☐ Trigger limited to the sandbox inbox or form.
- ☐ Classify prompt: labels, definitions, fallback, label only, injection guard.
- ☐ Allowed-list check in the condition.
- ☐ Review queue for "other."
- ☐ Channel posts contain IDs, not personal data.
- ☐ Ten test cases run and logged.

Do and don't

Do	Don't
Define output fields	Parse free text in later steps
Test each path	Test once with a perfect example
Keep a run log	Rely on memory of what happened
Stay in the sandbox	Point the trigger at a real inbox