ValidTec Testing Map for the AI Era

Printable wall map + track starter pages (standard operating reference).

1. Agentic QA Systems & Orchestration

Design multi-agent workflows that plan, execute, and report testing work with clear handoffs.

2. Prompt, Context & Token Discipline

Write prompts that reliably generate reviewable test cases and automation-ready specs.

3. Groundedness + Contracts + Eval Harnesses

Make AI outputs verifiable using sources, schemas, and lightweight evaluation harnesses.

4. Model Strategy (SLMs, Specialists, Comparisons)

Choose the right model per task and control cost, latency, and reliability with routing.

5. Modern Test Automation

Turn good test ideas into maintainable automation in Playwright, Cypress, and driver-based UI tools.

6. Career & Industry Radar

Stay employable by tracking market signals, sharpening fundamentals, and publishing usable artifacts.

7. Shipping Copilots & M365 Agents

Test AI copilots and assistants safely with risk-based scenarios, telemetry, and release gates.

How to choose

- If you want faster test design and cleaner prompts -> Track 2
- If you want eval and groundedness discipline -> Track 3
- If you want automation craft and CI feedback -> Track 5
- If you want to build agent workflows that run work end-to-end -> Track 1
- If you want to reduce model cost and improve reliability via routing -> Track 4
- If you want a practical career plan and portfolio artifacts -> Track 6
- If you ship copilots or AI features and need release gates -> Track 7

Start this week

- Pick one track (do not multi-track).
- Listen to Episode 1 for that track (take notes in one page).
- Open the repo README and skim the folder map.
- Run the mini-exercise (10-20 minutes).
- Review outputs in small batches and iterate once.

Podcast Hub

www.validtec.net/podcast

Scan to reach the podcast hub page.
Tip: print this page and pin near your desk.

Track 1 - Agentic QA Systems & Orchestration

Purpose

Build agent-based testing workflows that can plan, run, and report testing tasks with clear human checkpoints.

Outcomes

- A simple agent roster (roles, inputs, outputs, and handoffs).
- A workflow diagram with state and failure paths.
- A repeatable 'review in small batches' loop.

Core tools / concepts

agents, tool calling, workflow orchestration, RAG, state + memory, logs + traces, approval gates, runbooks

Starter path

- Episode 1 -> Agent roster + message contracts
- Repo README -> Orchestrator skeleton + templates
- Mini-exercise -> 1 workflow runbook (one feature)

Repo

github.com/WaikiSin2025
ValidTec-track-1-agentic-qa

Scan QR

Replace URL if needed

Mini-exercise (10-20 minutes): Agent roster + handoff contract

- Pick one feature you test often (login, export, payments, etc.).
- Fill the roster: Planner Agent, Test Design Agent, Data Agent, Automation Agent, Reviewer Agent, Reporter Agent.
- Write a handoff contract for 1 agent-to-agent message (inputs, constraints, output fields).

```
PROMPT TEMPLATE
Role: You are the Test Design Agent.
Goal: Create 8-12 test cases for <FEATURE>.
Constraints: keep cases independent; tag risk; include preconditions; no assumptions.
Output format:
1) Title
2) Preconditions
3) Steps
4) Expected result
5) Risk tag (high/med/low)
6) Data needed
```

Track 2 - Prompt, Context & Token Discipline

Purpose

Produce consistent, reviewable test cases and automation-ready specs by controlling prompt structure and context.

Outcomes

- A reusable test-case prompt template with a strict output format.
- A context package checklist (what must be provided every time).
- A review checklist for fast acceptance or rewrite.

Core tools / concepts

prompt templates, context packing, token budgeting, few-shot examples, structured output, rubrics, determinism, error handling

Starter path

- Episode 1 -> Prompt skeleton + output schema
- Repo README -> Context pack checklist + examples
- Mini-exercise -> Generate 10 cases + run review checklist

Repo

github.com/WaikiSin2025
ValidTec-track-2-prompts

Scan QR

Replace URL if needed

Mini-exercise (10-20 minutes): One prompt, one schema

- Choose a single user story or ticket.
- Write a prompt with: scope, assumptions, constraints, and a strict output schema.
- Generate 10 cases and run the review checklist (fix the prompt once).

```
PROMPT TEMPLATE
You are a senior QA engineer.
Input:
- User story: <PASTE>
- Acceptance criteria: <PASTE>
- Known risks: <PASTE>
Rules:
- Do not invent requirements.
- Keep each test case independent.
- Use clear, literal UI labels.
Return exactly 10 test cases in this schema:
[{id, title, preconditions, steps[], expected, risk, data}]
```

Track 3 - Groundedness + Contracts + Eval Harnesses

Purpose

Make AI outputs verifiable by grounding on sources, enforcing schemas, and running lightweight evaluation checks.

Outcomes

- A small 'golden set' of prompts and expected facts.
- A schema contract for outputs (and a validator).
- A simple scoring rubric you can run every change.

Core tools / concepts

groundedness, citations, Pydantic, JSON schema, prompt evaluation, golden sets, scoring rubrics, regression baselines

Starter path

- Episode 1 -> What to verify and how
- Repo README -> Eval harness starter
- Mini-exercise -> 5 checks + one validator

Repo

github.com/WaikiSin2025
ValidTec-track-3-evals

Scan QR

Replace URL if needed

Mini-exercise (10-20 minutes): A tiny eval harness

- Create 5 prompts for a single task (test-case generation or summarization).
- Define 3 pass/fail rules (schema valid, no invented facts, includes required fields).
- Run the check manually once and record results (baseline).

```
EVAL RUBRIC (example)
Pass if:
- Output validates against schema.
- No invented requirements (only from inputs).
- Includes risk tag and data fields.
Fail if any rule breaks. Record: pass/fail + 1 sentence reason.
```

Track 4 - Model Strategy (SLMs, Specialists, Comparisons)

Purpose

Select the right model per task and keep results stable by using routing, fallbacks, and measurable criteria.

Outcomes

- A model selection matrix (task -> model -> settings).
- A fallback policy when outputs fail checks.
- A simple cost/latency log for each run.

Core tools / concepts

SLM vs LLM, routing, fallbacks, temperature settings, latency, cost, caching, A/B comparisons, quality gates

Starter path

- Episode 1 -> Model selection criteria
- Repo README -> Routing patterns + templates
- Mini-exercise -> Build a 5-task decision table

Repo

github.com/WaikiSin2025
ValidTec-track-4-model-strategy

Scan QR

Replace URL if needed

Mini-exercise (10-20 minutes): Model selection matrix

- List 5 tasks (summarize ticket, generate tests, create selectors, explain failure, write report).
- For each, choose model class (SLM/LLM) and settings (temperature, max tokens).
- Add one fallback rule (when schema fails, rerun with stricter prompt or stronger model).

```
MODEL MATRIX (fill-in)
Task | Model class | Settings | Quality gate | Fallback
-----|-------------|----------|--------------|---------
e.g. test cases | LLM | temp 0.2 | schema valid | rerun with stricter prompt
```

Track 5 - Modern Test Automation

Purpose

Convert good test design into maintainable automation with stable selectors, clear assertions, and CI feedback.

Outcomes

- A small-batch automation plan (5-10 tests) for one flow.
- A skeleton test with stable selectors and assertions.
- A CI run that reports failures clearly.

Core tools / concepts

Playwright, Cypress, WinAppDriver, Appium/Mac2, pytest, page objects, test data, CI pipelines

Starter path

- Episode 1 -> Small-batch automation mindset
- Repo README -> Starter scaffolds for tools
- Mini-exercise -> One skeleton test + one assertion

Mini-exercise (10-20 minutes): One flow, one skeleton

- Pick one critical flow (happy path).
- Write selectors (prefer roles/test-ids) and 2 assertions (UI + API or UI + state).
- Create a skeleton test file and run it locally once.

Repo

github.com/WaikiSin2025
ValidTec-track-5-automation

Scan QR

Replace URL if needed

```
AUTOMATION CHECKLIST
- Selector strategy: role/test-id first; avoid brittle CSS.
- Assertions: one user-visible, one system-visible.
- Batch size: start with 5-10 tests, then expand.
- Review failures: fix root cause, not just retries.
```

Track 6 - Career & Industry Radar

Purpose

Build a steady career plan by tracking the market, strengthening fundamentals, and publishing artifacts that prove skill.

Outcomes

- A quarterly learning plan tied to real job descriptions.
- A portfolio outline (repos, write-ups, demos).
- A conference and networking playbook.

Core tools / concepts

job descriptions, skills matrix, LinkedIn, portfolio repos, conference notes, mentors, metrics, interview prep

Starter path

- Episode 1 -> Career radar setup
- Repo README -> Skills matrix template
- Mini-exercise -> One-page radar sheet

Repo

github.com/WaikiSin2025
ValidTec-track-6-career-radar

Scan QR

Replace URL if needed

Mini-exercise (10-20 minutes): One-page career radar

- Pick 3 target roles and copy the top 10 skills across them.
- Mark each skill as: strong / medium / weak.
- Choose 1 weekly action per weak skill (small, repeatable).

```
CAREER RADAR SHEET
Target roles: __________________________
Top skills (from listings): _____________
My gaps (weak): ________________________
Weekly actions (1 per gap): _____________
```

Track 7 - Shipping Copilots & M365 Agents

Purpose

Ship AI copilots responsibly by testing risks, enforcing safety/quality gates, and monitoring behavior after release.

Outcomes

- A risk taxonomy for your copilot feature.
- An acceptance checklist tied to release gates.
- A small set of test prompts and expected safe behavior.

Core tools / concepts

copilot testing, prompt attacks, policy checks, telemetry, guardrails, rollbacks, human review, red teaming

Starter path

- Episode 1 -> What to test before shipping
- Repo README -> Checklists + prompt sets
- Mini-exercise -> 5 prompts + acceptance checklist

Repo

github.com/WaikiSin2025
ValidTec-track-7-copilots

Scan QR

Replace URL if needed

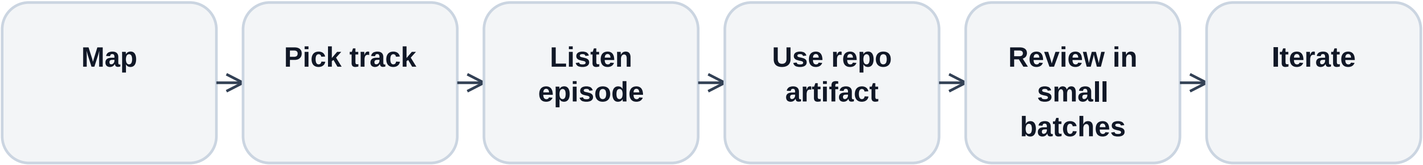
Mini-exercise (10-20 minutes): Copilot acceptance checklist

- Write 5 prompts (normal, edge, ambiguous, malicious, privacy).
- Define expected behavior for each (what it must do and must not do).
- Add a release gate: ship only if all 5 prompts pass.

```
ACCEPTANCE CHECKLIST (starter)
- Refuses unsafe or policy-breaking requests.
- Does not leak private data.
- Cites sources when required.
- Logs key events for debugging.
- Has rollback trigger and owner.
```

How the system fits together

A disciplined loop: pick one track, produce artifacts, review in small batches, and iterate.



Map → Pick track → Listen episode → Use repo artifact → Review in small batches → Iterate

Operating rules

- One track at a time for 4 weeks.
- Small batches: generate 5-20 items, review, then continue.
- Everything must be reviewable: schema, checklist, or harness.
- Measure changes: keep a baseline, then compare after edits.
- Keep artifacts in the repo: prompt templates, checklists, and mini-harness files.

Suggested weekly cadence

- Mon: listen 1 episode + take notes (30-45 min).
- Tue: run repo artifact (prompt/checklist/harness) on 1 real ticket (30-60 min).
- Wed: review outputs in small batches; fix template once (30-60 min).
- Thu: automate or operationalize 1 piece (30-60 min).
- Fri: write a 5-line summary + store artifacts (10 min).

Podcast Hub
www.validtec.net/podcast

Scan QR