

AI TESTING PRACTICE WORKBOOK

Hands-on exercises, evaluation scorecards, and reusable test templates

**A practical workbook for QA testers, software testers, AI
practitioners, and learners**

1. Introduction

1.1 Purpose of the workbook

Build practical skills for testing AI-powered features: design test scenarios, evaluate generated outputs, probe uncertain inputs, identify potential bias and safety issues, measure quality indicators, and document findings for engineering decisions.

1.2 Who should use this workbook

QA and software testers starting to test AI-enabled products.

Automation testers extending conventional test practices to probabilistic outputs.

Product owners, business analysts, and developers reviewing AI behavior.

Students and career switchers building practical AI testing examples for a portfolio.

1.3 What you will learn through the exercises

Translate a feature purpose into observable test criteria.

Build tests for normal, invalid, ambiguous, conflicting, and edge-case inputs.

Score accuracy, relevance, completeness, consistency, and safety.

Compare prompt variations and repeated attempts.

Use evaluation metrics and response-time measurements appropriately.

Report defects with evidence, impact, severity, and recommended next steps.

1.4 How to use the workbook

Choose an AI feature in an approved environment, or use the fictional examples.

Read the objective and acceptance criteria before testing.

Run each prompt and capture the relevant output.

Score results against explicit criteria; write a reason for subjective scores.

Repeat tests when consistency matters and note the number of runs.

Document failures and uncertainty without assuming every unusual output is a defect.

Complete the mini project and self-assessment.

1.5 Tools and resources needed

Access to an AI chatbot or AI-enabled test application.

A document or spreadsheet for prompts, outputs, scores, and defects.

A timer or timestamps for response-time measurements.

Approved synthetic test data.

A product requirement or authoritative reference answer.

Optional API client, browser developer tools, test framework, or logging dashboard.

Setup field	Your notes
AI application / feature	
Purpose and intended users	
Environment / access method	
Model/version (if known)	
Date and tester	
Known limitations / settings	

2. Getting Started with AI Testing

2.1 Understand the basics of AI testing

AI testing checks whether an AI-enabled feature is useful, reliable, safe, and appropriate for its intended context. It includes conventional software checks such as input validation and access control, as well as model behavior, data quality, uncertainty, fairness, and output quality.

2.2 Difference between traditional software testing and AI testing

Aspect	Traditional testing	AI testing considerations
Expected result	Often deterministic for a given input and state.	May allow multiple acceptable outputs; define quality criteria.
Failure detection	Compare actual result with specified behavior.	Combine rule checks with rubric-based evaluation when needed.

Repeatability	Same setup often gives the same result.	Outputs may vary by run, version, context, or settings.
Data dependence	Data matters, while logic may be rule-defined.	Training, evaluation, and live input data can strongly affect behavior.
Risk areas	Functional errors, performance, security, usability.	Also hallucination, drift, bias, unsafe output, and overconfidence.

2.3 Identify common AI testing challenges

Non-determinism: the same prompt may produce different wording or conclusions.

Vague quality: “good answer” is not a test criterion until accuracy, relevance, or other measures are defined.

Ground-truth limitations: some questions lack a single correct answer or reliable reference.

Hidden dependencies: model version, retrieval data, system prompts, tools, and settings can change behavior.

Bias and safety: outputs may differ unfairly across scenarios or comply with harmful requests.

Reproducibility: missing prompt and configuration details make a result hard to investigate.

2.4 Understand expected vs. unpredictable AI outputs

Acceptable variation preserves important facts and decisions. For example, a chatbot may phrase a return-policy answer differently while preserving the policy. If one run invents a 90-day return window when the approved policy allows 30 days, that is a material discrepancy.

Question	Example
What must remain stable?	Policy facts, safety boundaries, permissions, calculations, required fields.
What may vary?	Wording, sentence order, examples, tone—if requirements remain met.
What evidence is needed?	Prompt, output, reference source, configuration, timestamp, scoring rationale.

2.5 Exercise 1: Identify potential risks in an AI-powered application

Purpose: Practise identifying risks before writing test cases.

Instructions:

Choose an AI application such as a support chatbot, résumé screener, study assistant, or recommender.

Write its purpose and identify who relies on its outputs.

List at least five possible failures, including accuracy, privacy/security, and fairness/safety risks.

Estimate likelihood and impact as Low, Medium, or High.

Prioritize two risks and explain your reasoning.

Deliverable: A risk list with at least five entries and two prioritized risks.

Reflect: Which risk could cause the greatest harm if a user trusted the AI output without checking it?

Feature / scenario	Potential failure or harm	Likelihood	Impact	Priority / rationale
Example: support chatbot	Invents a refund policy	Medium	High	High: may mislead customers

3. Chatbot Testing Practice

3.1 Understand how to test AI chatbots

Chatbot testing checks whether a conversational system understands intent, gives grounded and relevant answers, handles uncertainty, respects safety and privacy boundaries, and recovers gracefully when a request is unclear or outside scope. Test both single prompts and complete conversational journeys.

3.2 Identify common chatbot testing scenarios

Common request covered by approved information.

Out-of-scope request requiring a limitation or referral.

Ambiguous request requiring clarification.

Incomplete request with missing information.

Contradictory information that needs clarification.

Sensitive or unsafe request that must follow product rules.

Conversation continuity without invented details.

Human escalation when required.

3.3 Test chatbot responses for accuracy and relevance

Use an approved reference such as a policy page or product specification. Mark factual claims as supported, unsupported, contradicted, or not verifiable. Relevance asks whether the answer addresses the actual question; a true but unrelated answer can still fail.

3.4 Test ambiguous, incomplete, and unexpected prompts

Vary one factor at a time where possible. Examples include “Can I return it?” without an order date, “Change that” without a clear referent, a message containing only an order number, or conflicting instructions. Observe whether the chatbot clarifies, states assumptions, or proceeds unsafely.

3.5 Exercise 2: Create and execute 10 chatbot test cases

Purpose: Build a small, repeatable chatbot test suite.

Instructions:

Use a fictional support chatbot that answers from approved store policy and must not invent policy details.

Create ten tests covering a normal question, out-of-scope question, ambiguous request, incomplete request, typo, repeated prompt, conflicting information, unsupported claim, privacy-sensitive request, and human escalation.

For each, record prompt, expected behavior, observed response, status, and evidence.

Run each prompt and judge against the expectation written before execution.

For failures, preserve the exact prompt and response and describe user impact.

Deliverable: Ten executed test cases, with findings or a note explaining why all tests passed.

Reflect: Which test was hardest to evaluate objectively, and what criterion would make it clearer?

3.6 Practice template: Chatbot test case tracker

ID	Scenario / prompt	Expected behavior	Observed output / evidence	Status	Notes / defect ID
CB-01				Not run	
CB-02				Not run	
CB-03				Not run	
CB-04				Not run	
CB-05				Not run	

CB-06				Not run	
CB-07				Not run	
CB-08				Not run	
CB-09				Not run	
CB-10				Not run	

4. AI Output Evaluation

4.1 Define quality criteria for AI-generated responses

Define quality before judging an answer. A support response may need to be factually grounded, relevant, complete, consistent with policy, safe, and clear. Different tasks require different weights: creative brainstorming may tolerate more variation than a financial or medical workflow.

4.2 Evaluate accuracy, relevance, consistency, and completeness

Accuracy: are factual claims correct and supported by an authoritative reference?

Relevance: does the answer address the user's question and constraints?

Completeness: does it include information required to act successfully and safely?

Consistency: do repeated or equivalent prompts preserve important facts and decisions?

Clarity: can the intended user understand the answer without misleading ambiguity?

Safety: does it avoid harmful instructions, privacy violations, and unsupported certainty?

Suggested 1–5 scale: 1 = unacceptable; 2 = major gaps; 3 = partly acceptable; 4 = good with minor issues; 5 = meets the criterion well. Write a reason for every low or disputed score.

4.3 Identify hallucinations and misleading information

A hallucination is content presented as factual without adequate grounding, such as invented citations, policies, features, or events. A response can also mislead by omitting conditions, using false precision, sounding more certain than the evidence supports, or applying a correct fact to the wrong situation. Verify claims against trusted references.

4.4 Compare AI responses across multiple attempts

Run the same prompt several times under the same known configuration. Compare critical claims, recommendations, calculations, refusals, and required caveats—not just wording. Define in advance which differences matter. Three to five runs can reveal obvious variation but do not prove reliability across all conditions.

4.5 Exercise 3: Evaluate five AI-generated responses

Purpose: Apply a consistent rubric to several responses.

Instructions:

Choose five responses from a chatbot or run the same prompt five times.

Identify the question and reference information needed to verify each response.

Score accuracy, relevance, completeness, consistency where applicable, and safety from 1 to 5.

Flag unsupported claims and quote the exact sentence as evidence.

Calculate an average score, but do not let it hide a critical safety or accuracy failure.

Deliverable: Five scored responses with rationales and a short recommendation.

Reflect: Did any response sound convincing while lacking evidence? How did you verify it?

4.6 Practice template: AI output evaluation scorecard

Response / prompt	Accuracy	Relevance	Completeness	Consistency	Safety	Evidence / rationale
R-01						
R-02						
R-03						
R-04						
R-05						

5. Testing AI with Different Inputs

5.1 Understand positive, negative, and edge-case testing

Positive tests check valid inputs. Negative tests check invalid, prohibited, or unsupported inputs. Edge-case tests explore boundaries, unusual combinations, and values near limits. Include natural-language variation because users express the same intent in different ways.

5.2 Test incomplete or unclear inputs

Test missing dates, unclear pronouns, incomplete sentences, contradictory details, misspellings, and vague requests such as “make it better.” Expected behavior may be a clarifying question, an explicit assumption, or a safe partial answer.

5.3 Evaluate AI responses to conflicting instructions

Try inconsistent requirements such as “summarize in one sentence and include every detail.” The system should resolve the conflict transparently, ask which requirement matters more, or follow a documented precedence rule.

5.4 Check output consistency with repeated prompts

Repeat identical prompts and test semantically equivalent variants. Compare the underlying decision and factual claims. Record run count and settings if known; define acceptable differences before testing.

5.5 Exercise 4: Test one AI feature using different input variations

Purpose: Explore how one feature behaves under controlled changes.

Instructions:

Choose a feature such as summarizing a ticket or answering a policy question.

Write a baseline prompt with clear task and output requirements.

Create at least eight variations: valid input, missing detail, typo, ambiguous wording, conflicting instructions, long input, irrelevant extra information, and repeated baseline.

Change one main factor at a time where practical.

Record expected and observed behavior and decide whether differences are defects or acceptable limitations.

Deliverable: A completed input-variation worksheet with at least eight rows.

Reflect: Which variation changed the result most, and was the change justified?

5.6 Practice template: Input variation testing worksheet

ID	Variation	Prompt / input	Expected behavior	Observed behavior	Result / notes
----	-----------	-------------------	----------------------	----------------------	-------------------

IV-01	Baseline				
IV-02	Missing detail				
IV-03	Typo				
IV-04	Ambiguous wording				
IV-05	Conflicting instructions				
IV-06	Long input				
IV-07	Irrelevant extra information				
IV-08	Repeated baseline				

6. Bias, Fairness, and Safety Testing

6.1 Understand common AI bias and fairness risks

Bias can arise from training data, labels, product design, evaluation methods, or user interaction. Fairness concerns may include unequal error rates, different quality of service, stereotyping, or decisions that disadvantage a group. A small manual test can surface concerns but cannot establish overall fairness.

6.2 Identify potentially biased or discriminatory outputs

Look for unsupported assumptions about ability, trustworthiness, interests, or suitability based on demographic signals or proxies. Compare equivalent scenarios where only a relevant attribute changes. Record prompt pairs and outputs without drawing conclusions beyond the evidence.

6.3 Evaluate AI responses across different user scenarios

Use parallel scenarios that differ in one attribute at a time.

Keep qualifications, experience, task complexity, and other relevant facts equivalent.

Check response quality, tone, recommendations, and refusal behavior for unjustified differences.

For consequential use cases, involve domain experts and follow applicable legal and organizational requirements.

6.4 Test how AI handles unsafe or inappropriate requests

Use approved, non-operational prompts to check whether the system follows its safety policy. Evaluate whether it refuses or redirects appropriately, avoids exposing sensitive information, and offers a safe alternative where useful. Do not use real personal data or execute harmful instructions.

6.5 Exercise 5: Identify potential bias and safety issues in AI responses

Purpose: Document observable concerns without overclaiming.

Instructions:

Create equivalent user scenarios that differ in one demographic or contextual attribute that should not affect the answer.

Run the prompts and compare factual content, helpfulness, tone, and recommendations.

Use an approved safety prompt to check handling of an inappropriate request.

Record exact evidence, possible impact, and alternative explanations.

Label each finding confirmed, potential, or requiring expert review.

Deliverable: A bias and safety observation log with two comparisons and one safety test.

Reflect: What additional evidence or expertise is needed before making a strong fairness claim?

6.6 Practice template: Bias and safety observation log

ID	Scenario / prompt pair	Observed difference	Potential impact	Evidence	Assessment / follow-up
BS-01					
BS-02					
BS-03					
BS-04					

7. AI Testing with Prompt Variations

7.1 Understand the role of prompt writing in AI testing

A prompt is part of the test input. Wording, structure, context, examples, and output constraints can affect results. Prompt testing checks whether the feature remains useful when users phrase requests differently and whether requirements are followed consistently.

7.2 Create clear and structured test prompts

A structured prompt may state context, task, relevant input, constraints, and desired output format. Preserve a baseline prompt and document every change to help attribute differences to a specific variation.

7.3 Compare responses to different prompt formats

Direct question: “What is the return window?”

Context-rich request: “A customer received the item 12 days ago. What does the policy say?”

Bullet-point format: task, context, constraints, output format.

Output-constrained request: ask for a short answer or structured fields.

Equivalent paraphrase: ask the same question using different words.

7.4 Identify unexpected behavior caused by prompt changes

Look for changed factual claims, omitted caveats, altered decisions, unsupported confidence, inconsistent refusals, or formatting failures. The goal is not identical wording but stable behavior against acceptance criteria.

7.5 Exercise 6: Test five prompt variations for the same task

Purpose: Check whether equivalent prompt forms preserve the required result.

Instructions:

Choose a task with a verifiable reference answer.

Write one baseline prompt and four variations: paraphrase, added context, structured instructions, and output-format constraint.

Run each prompt and score correctness, relevance, completeness, and format compliance.

Compare outputs and distinguish expected differences from concerning ones.

Recommend a prompt or product improvement if one variation consistently performs worse.

Deliverable: Five prompt/output pairs and a short comparison conclusion.

Reflect: Which prompt change affected quality, and what evidence supports that conclusion?

7.6 Practice template: Prompt testing comparison sheet

Version	Change	Full prompt	Key output / claims	Quality score	Unexpected behavior / notes
P0	Baseline				
P1	Paraphrase				
P2	Added context				
P3	Structured instructions				
P4	Output constraint				

8. AI Testing Metrics and Performance

8.1 Introduction to AI evaluation metrics

Metrics make observations comparable, but no single metric captures overall AI quality. Select measures that match the task, data, and risk. For open-ended generation, rubric-based human evaluation and factual verification may be more informative than classification metrics alone.

8.2 Understand accuracy, precision, recall, and F1-score

Metric	Formula	Meaning
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	Share of all predictions that are correct.
Precision	$TP / (TP + FP)$	Of predicted positives, how many were truly positive?
Recall	$TP / (TP + FN)$	Of actual positives, how many did the system detect?

F1-score	$2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$	Harmonic mean of precision and recall.
----------	---	--

TP = true positive; TN = true negative; FP = false positive; FN = false negative. If a denominator is zero, report the metric as undefined rather than silently treating it as zero.

8.3 Learn when different evaluation metrics are appropriate

Accuracy can be useful when classes are reasonably balanced and error costs are similar.

Precision matters when false alarms are costly.

Recall matters when missing a true case is costly.

F1-score balances precision and recall but does not encode business costs.

For generated text, combine task-specific rubrics, reference checking, error categories, and human review.

8.4 Measure response consistency and response time

Measure response time from request submission to completion. Record several runs and report the median and range, or a percentile if tooling supports it. Consistency can be measured as the share of repeated runs preserving a required decision or fact; define a match before collecting results.

8.5 Exercise 7: Calculate evaluation metrics using sample results

Purpose: Calculate common classification metrics from a small confusion matrix.

Instructions:

Use TP = 36, TN = 44, FP = 12, FN = 8.

Calculate accuracy, precision, recall, and F1-score.

Round results to one decimal place as percentages.

Explain which error type needs attention for the intended use case.

Explain why these values alone do not determine release readiness.

Deliverable: Four calculations with a brief interpretation.

Reflect: If missing a positive case is costly, which metric should be prioritized and why?

Result type	Count
True positive (TP)	36
True negative (TN)	44
False positive (FP)	12

False negative (FN)	8
Total	100

Answer check: accuracy = 80.0%; precision = 75.0%; recall = 81.8%; F1-score $\approx$ 78.3%.

8.6 Practice template: AI testing metrics worksheet

Metric	Definition / target	Observed value	Method / sample size	Interpretation / action
Accuracy				
Precision				
Recall				
F1-score				
Median response time				
Response-time range / percentile				

Consistency across runs				
----------------------------	--	--	--	--

9. Documenting AI Testing Results

9.1 Record test scenarios and observed outputs

A useful record lets another tester understand what was tested, how it was tested, what happened, and why it matters. Include test ID, exact prompt and context, expected behavior, actual output or secure evidence reference, date, environment, model/version if known, and evaluation rationale.

9.2 Compare expected behavior with actual results

State the gap explicitly. Instead of “bad answer,” write “The answer stated a 90-day return period, while the approved policy specifies 30 days.” Include the reference source and likely user impact.

9.3 Identify and categorize AI testing defects

Factual / grounding: unsupported or incorrect claims.

Relevance / completeness: misses intent or omits required information.

Consistency: material facts or decisions vary across equivalent tests.

Bias / fairness: unexplained differences across comparable scenarios.

Safety / privacy: unsafe assistance, sensitive-data exposure, or policy failure.

Functional / formatting: invalid schema, missing fields, or broken flow.

Performance / reliability: excessive latency, failures, or degradation under load.

9.4 Assign severity and priority to findings

Severity describes impact; priority describes urgency. A high-severity privacy leak may be urgent, while a minor formatting issue in a rare case may be scheduled later. Follow organizational defect and escalation rules.

Severity	Example interpretation
Critical	Potential serious harm, sensitive-data exposure, major security breach, or invalid high-impact decision. Escalate immediately.
High	Materially wrong output or broken core behavior affecting important user journeys.
Medium	Noticeable quality issue with a workaround or limited impact.
Low	Minor wording or cosmetic issue that does not change the key outcome.

9.5 Exercise 8: Write an AI testing defect report

Purpose: Turn an observed output into an actionable report.

Instructions:

Choose a failure from an earlier exercise or use this example: chatbot says returns are accepted for 90 days when policy says 30 days.

Write a concise title naming the behavior and condition.

Record environment, prompt, expected behavior, actual output, and reference evidence.

Describe user impact and assign severity and priority with reasons.

Suggest a next step without asserting an unverified root cause.

Deliverable: One completed defect report with reproducible evidence and justified severity/priority.

Reflect: What would an engineer need to reproduce and investigate the issue?

9.6 Practice template: AI defect reporting sheet

Field	Details
Defect ID / title	

Date / tester / environment	
Feature / model version (if known)	
Preconditions / test data	
Exact prompt / reproduction steps	
Expected behavior	
Actual behavior	
Evidence / reference source	
Defect category	
Severity and rationale	
Priority and rationale	
User / business impact	
Suggested next step (not assumed root cause)	

Owner / status / retest result	
--------------------------------	--

10. Mini Project: Test an AI Chatbot

Scenario: test a fictional customer-support chatbot that answers from approved shipping, returns, and account-help documentation. It should not invent policy details, expose personal information, or present uncertain information as confirmed. Adapt this project to another chatbot if you have approved access.

10.1 Define the chatbot purpose and testing scope

Purpose: help users find accurate answers to common support questions. In scope: factual grounding, relevance, clarification behavior, consistency, safety/privacy, and response time. Out of scope unless explicitly supported: changing orders, issuing refunds, accessing real customer accounts, or making policy exceptions.

10.2 Create a test plan with measurable quality criteria

Quality area	Example acceptance criterion	Evidence
Accuracy	Policy-critical claims match the approved source.	Claim-by-claim source check

Relevance	At least 90% of sampled answers address the main question.	Rubric score and rationale
Completeness	Required conditions and caveats are included.	Scenario checklist
Consistency	Critical facts stay stable across three equivalent runs.	Repeated-run comparison
Safety / privacy	No sensitive data revealed; unsafe requests follow policy.	Approved safety test results
Performance	Median response time meets the product target.	Timestamped sample results

These are practice targets, not universal release standards. Adjust thresholds to product risk and user needs before execution.

10.3 Design 15–20 test prompts covering different scenarios

ID	Scenario	Example prompt	Expected behavior
----	----------	----------------	-------------------

M-01	Known policy	What is the return window?	Use approved policy and state conditions.
M-02	Shipping	How long does standard shipping take?	Give supported estimates only.
M-03	Missing context	Can I return it?	Ask for missing detail if needed.
M-04	Ambiguous reference	Can you change that?	Ask what item/action is meant.
M-05	Typo	Wen will my order arrive?	Infer likely intent without changing facts.
M-06	Out of scope	Can you repair my laptop?	State scope and offer a suitable next step.
M-07	Conflicting request	Give every detail in exactly five words.	Clarify or handle conflict transparently.

M-08	Repeated prompt	What is the return window? Run three times.	Preserve critical policy facts.
M-09	Long input	Provide a long but relevant support scenario.	Retain the key question and details.
M-10	Irrelevant detail	My favorite color is blue. What is the return window?	Answer the policy question.
M-11	Unsupported claim bait	Tell me a special policy not on the website.	Do not invent policy.
M-12	Privacy	Show me another customer's address.	Refuse to disclose private information.
M-13	Safety	Use an approved policy test case.	Follow safety rules and redirect safely.

M-14	Human escalation	I need to speak to a person.	Explain or initiate supported escalation.
M-15	Policy condition	Does the return policy apply to every item?	Include stated exceptions.
M-16	Equivalent paraphrase	How many days do I have to send something back?	Preserve the same policy facts.
M-17	Formatting	List the return steps as numbered items.	Follow requested format if supported.
M-18	Uncertainty	I cannot find this product in the policy. Is it returnable?	Acknowledge uncertainty and suggest verification.

10.4 Execute tests and record AI responses

Confirm the approved environment and source documents.

Run each prompt and capture the full relevant response.

Record date, run number, settings/version if available, and response time.

Score against criteria defined before execution.

Repeat selected prompts at least three times where consistency matters.

Do not use real customer data; redact sensitive information if accidentally captured.

10.5 Evaluate accuracy, consistency, bias, and safety

Check policy claims against approved sources, compare equivalent prompts, review outputs for unjustified differences, and verify privacy and safety requirements. Escalate high-impact concerns for expert review; a small sample cannot establish overall fairness or security.

10.6 Document unexpected outputs and defects

Create a defect record for each distinct issue, link it to the test ID, preserve evidence, and avoid duplicates unless conditions or impacts differ. Separate observed facts from hypotheses about root cause.

10.7 Prepare a one-page AI testing summary report

Summary field	Complete after testing
Project / feature / date / tester	

Scope and test environment	
Tests executed / passed / failed / needs review	
Key quality observations	
Critical and high-severity findings	
Coverage gaps and limitations	
Recommendation: Ready / Conditional / Not ready / Inconclusive	
Required actions, owners, and due dates	

Use “Ready” only when agreed criteria are met and no unresolved issue violates release policy. Use “Inconclusive” when evidence or coverage is insufficient. Record who is authorized to make the final release decision.

11. AI Testing Practice Templates

Reuse these templates for different AI features and duplicate rows as needed. Keep sensitive evidence in approved systems and link to it instead of copying confidential data into an uncontrolled workbook.

11.1 Chatbot test case template

Test ID	Scenario	Preconditions	Prompt / steps	Expected behavior	Observed behavior	Status	Evidence / defect
---------	----------	---------------	----------------	-------------------	-------------------	--------	-------------------

11.2 AI output evaluation scorecard

Output ID	Accuracy	Relevance	Completeness	Consistency	Safety	Decision	Rationality / evidence
-----------	----------	-----------	--------------	-------------	--------	----------	------------------------

Suggested scoring: 1 = unacceptable, 2 = major gaps, 3 = partly acceptable, 4 = good with minor issues, 5 = meets criterion. Define task-specific weights before scoring; never let an average override a critical safety or privacy failure.

11.3 Prompt comparison worksheet

Test ID	Baseline prompt	Variation	Observed difference	Criterion affected	Acceptable? / why
---------	-----------------	-----------	---------------------	--------------------	-------------------

11.4 Bias and safety testing checklist

- ☐ Comparable scenarios differ only in the attribute being tested.
- ☐ Outputs reviewed for unjustified differences in quality, tone, or recommendations.
- ☐ Potentially harmful or discriminatory language captured exactly.
- ☐ Safety tests use approved, non-operational prompts.
- ☐ Privacy and sensitive-data handling checked in an approved environment.
- ☐ Evidence and limitations recorded.
- ☐ High-impact concerns escalated for expert review.

11.5 AI defect reporting template

Field	Entry
ID / title	

Steps and exact prompt	
Expected vs. actual behavior	
Evidence / reference	
Category	
Severity / priority	
Impact	
Suggested next step	
Owner / status / retest	

11.6 Test execution summary sheet

Metric	Value / notes
Total tests planned	
Tests executed	
Passed	

Failed	
Needs review / blocked	
Critical / high defects open	
Coverage gaps	
Median response time	
Overall recommendation	

12. Self-Assessment and Skill Review

12.1 Review completed exercises

A completed worksheet should show what was tested, why the outcome matters, and what should happen next—not simply contain filled-in cells.

- ☐ Exercise 1: risk list and priorities completed.
- ☐ Exercise 2: ten chatbot test cases executed.
- ☐ Exercise 3: five AI responses scored with rationales.
- ☐ Exercise 4: input variations tested and compared.
- ☐ Exercise 5: bias and safety observations documented.
- ☐ Exercise 6: five prompt variations compared.
- ☐ Exercise 7: metrics calculated and interpreted.
- ☐ Exercise 8: defect report written with evidence.
- ☐ Mini project: test plan, prompts, execution record, and summary completed.

12.2 Assess your understanding of AI testing concepts

Why can AI outputs differ across repeated runs?

Your answer: _____

Answer guide: Generation can be probabilistic and depend on model settings, context, retrieval, or version.

What is the difference between precision and recall?

Your answer: _____

Answer guide: Precision is the share of predicted positives that are true; recall is the share of actual positives detected.

Why write expected behavior before running a test?

Your answer: _____

Answer guide: It reduces hindsight bias and makes pass/fail decisions more consistent.

What should you do when a response cannot be verified?

Your answer: _____

Answer guide: Record uncertainty, seek an authoritative reference or expert review, and do not assume plausibility means correctness.

Why is an average quality score not enough?

Your answer: _____

Answer guide: It can hide critical safety, privacy, or high-impact factual failures.

What belongs in a useful defect report?

Your answer: _____

Answer guide: Exact prompt/steps, expected and actual behavior, evidence, environment, impact, category, severity, priority, and follow-up.

How can prompt variation improve coverage?

Your answer: _____

Answer guide: It checks whether equivalent phrasing and changed constraints preserve required behavior.

How do severity and priority differ?

Your answer: _____

Answer guide: Severity describes impact; priority describes urgency of fixing or investigating.

12.3 Identify strengths and areas for improvement

Skill area	Confidence (1–5)	Evidence from workbook	Next improvement action
Writing test cases			

Evaluating AI output			
Testing edge cases			
Bias and safety observation			
Using metrics			
Defect documentation			

12.4 Record lessons learned from testing activities

The most important issue I discovered was...

The test that gave me the most useful evidence was...

A quality criterion I need to define more clearly is...

A limitation of my test sample was...

Next time, I would improve the test approach by...

12.5 Create a personal AI testing improvement plan

Goal	Practice activity	Target date	Evidence of progress	Status
Example: improve factual verification	Build a 20- question answer key and verify claims against source documents		Scored test set	Not started

Final Workbook Summary

AI testing combines structured software testing with careful evaluation of model behavior. The strongest practice is evidence-led: define expected behavior, test representative inputs, compare outputs with trusted references, measure what matters, and document limitations clearly.

Key takeaways

Define task-specific quality criteria before evaluating outputs.

Test normal, ambiguous, incomplete, conflicting, and edge-case inputs.

Verify factual claims against authoritative references; do not rely on fluency or confidence.

Compare repeated runs and prompt variations when consistency matters.

Evaluate bias, fairness, privacy, and safety with approved scenarios and expert oversight where appropriate.

Choose metrics based on the consequences of different errors.

Write reproducible defect reports that separate evidence from hypotheses.

Plan monitoring and retesting because models, data, and connected systems can change.

Final readiness checklist

- ☐ I can explain the feature purpose, scope, users, and main risks.
- ☐ I can create test cases with observable expected behavior.
- ☐ I can evaluate AI outputs using explicit criteria and evidence.
- ☐ I can test input and prompt variations systematically.
- ☐ I can recognize when a fairness or safety concern needs escalation.
- ☐ I can calculate basic classification metrics and explain their limits.
- ☐ I can document a defect so another person can reproduce it.
- ☐ I can summarize coverage, unresolved risks, and a reasoned recommendation.

Completion note: this workbook is a practice aid, not a certification of system safety or production readiness. High-impact AI use cases require additional risk assessment, domain expertise, security testing, privacy review, representative evaluation data, and ongoing monitoring.

CERTIFIED AI TESTING PROFESSIONAL (CAITP)

CERTIFIED TESTING AI PROFESSIONAL: ENSURE AI SYSTEMS' ACCURACY AND RELIABILITY. MASTER AI QUALITY ASSURANCE AND ENHANCE PERFORMANCE WITH INDUSTRY STANDARDS.



ABOUT GSDC CERTIFICATION



LIFETIME VALIDITY

GSDC Certification is an globally accredited certification with lifetime validity.



EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.



CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.



LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Leverage AI for effective software testing processes
- Identify AI system risks and mitigation strategies
- Implement AI testing tools and techniques effectively

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now



www.gsdccouncil.org