

AI FOR TESTERS

MODULE 1 – UNDERSTANDING WHAT YOU’RE TESTING

1

LESSON 1

AI WITHOUT THE HYPE

2

COURSE OBJECTIVES

- To give testers, test team leaders, and test managers a starting point in applying AI in testing.
- To understand the basic concepts of AI
- Learn how to test AI-based systems
- Learn how to apply generative AI to help perform testing
- Reduce your fear of AI
- Enhance your career by building AI skills



3

3

WHY AI?

- Artificial Intelligence (AI) is here to stay.
- AI is one of the major upheavals in technology, like the PC and the smartphone.
- AI will change what we do, how we do it, and who will do it.
- AI is a large and emergent field, and any syllabus and knowledge on the subject will be quickly outdated.
- This course is designed to help you understand AI, Generative AI, and how this affects testing.



4

4

TERMINOLOGY

- AI, in general, has many new terms associated with it.
- One of the purposes of this course is to acquaint you with AI terminology and the basics of AI.
- Soon (if not already), all testers will be expected to be “familiar” with AI, which will require a basic understanding and the ability to learn as AI evolves.



5

5

WHAT IS AI?

- According to IBM, “Artificial Intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy” (IBM, 2025).
- Wikipedia simplifies this as “the capability of computational systems to perform tasks typically associated with human intelligence.” (Wikipedia, 2025).



6

6

WHY CARE ABOUT AI?

- As testers, why do we care about AI?
- Is this just another blip or trend in the software world, like Agile and test automation – something we need to learn about but that will not change what we do?
- No, AI is set to be a major disruptor, more like the introduction of personal computers and smartphones.
- AI is likely to revolutionize how humans interact with computer systems and, as a result, will impact how we test it and how we test with it.

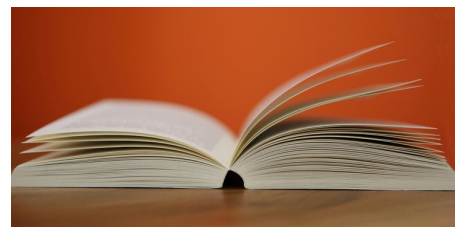


7

7

UNDERSTANDING AI AND ASSOCIATED TERMINOLOGY

- In order to understand what and how to test AI, we need to have a basic understanding of what it is.
- One challenge of entering the AI world is mastering the terminology.
- We will explain new terms as they are introduced.
- AI has *many* new terms.



8

8

LEARNING OBJECTIVE

- Understand basic AI terminology, especially the difference between AI and GenAI



9



9

TERMS

- **Generative AI (GenAI)** - An advanced technological approach that enables the creation of content including text, images, and videos. (the abbreviation for 'generative artificial intelligence')
- **Machine learning (ML)** - The process using computational techniques to enable systems to learn from data or experience (ISO/IEC TR 29119-11).
- **Large language model (LLM)** - A computer program that uses very large collections of language data in order to understand and produce text in a way that is similar to the way humans do.
- **Small language model (SLM)** - A language model that is intentionally designed and trained to be small, offering a balance between efficiency and task-specific language understanding

10



10

AI VS GENERATIVE AI

- **AI and Generative AI are not the same thing.**
- **AI is the broader field that covers the technologies that enable machines to perform tasks that formerly required a human mind.**
 - This includes such things as learning and problem-solving based on the information that has been learned.
- **Generative AI (GenAI) is a subset of AI that focuses on creating or “generating” new text, images, code, etc., based on patterns that have been learned from existing data.**

11



11

EXAMPLE

- For example, if you want the software to identify a rabbit based on a set of images of rabbits upon which it has been trained, you are using AI.
- If you want the software to draw a rabbit wearing a top hat, that's GenAI.



12



12

AI ACCURACY

- The accuracy and capability of AI is heavily dependent on the accuracy and applicability of the data that was used to train it.
- If the AI module had been shown a set of images of dogs, but told they were rabbits, it won't be able to accurately identify (or draw) a rabbit.



13



13

MACHINE LEARNING

- Machine Learning (ML) refers to the ability of the machine (the computer and computer systems) to “learn” based on the information it is given.
- This information is used to create algorithms that are then used to analyze the patterns in data (such as words).
- This information is then used to build predictive models (i.e., what word is likely to follow in a sentence), and make decisions with little or no human intervention.
- ML systems are expected to learn from experience as they encounter more data, and to become more accurate and efficient.



14



14

LARGE AND SMALL LANGUAGE MODELS

- Large Language Models (LLM) and Small Language Models (SLM) are types of machine learning models.
- As indicated by the name, LLMs are trained using huge datasets of text.
- The sheer size of the data used to train an LLM results in significant expense and computational resource requirements.
- For example, a training dataset for an LLM could be the entire Internet.
- Because of the size and depth of the training data, LLMs can write new content, answer complex questions, summarize large amounts of information and can even translate languages.

15



15

SMALL LANGUAGE MODELS

- SLMs are smaller and less complex and are generally more targeted in their learning and application.
- They are generally well-suited for specific tasks or specific domains (such as software testing) based on their focused training.
- They are less expensive than LLMs and don't have the same resource requirements.

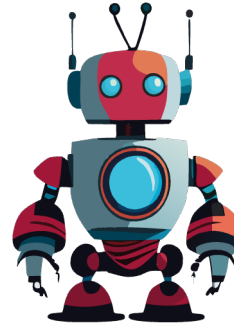
16



16

AI AGENTS

- **Agentic AI can independently plan and take actions to achieve a goal, often using tools, making decisions, and adapting its approach based on results with limited human direction.**



17

17

EXERCISE

- **In this exercise, we will use Google Gemini to generate some test cases from a short requirement.**
- **The requirement:**
 - The system shall be able to calculate postage based on weight:
 - Under 1 pound, the postage is \$10
 - From 1 pound to 4.99 pounds, the postage is \$15
 - From 5 pounds to 9.99 pounds, the postage is \$20
 - Over 10 pounds, the postage is \$2.95 per pound



18

18

EXERCISE (2)

- **Instructions**

- Go to google.com and click on AI Mode.
- Enter the following prompt. (Note the key elements of the prompt)
 - Context – I am a tester working on creating test cases to verify a requirement for calculating postage rates.
 - Instruction – Create test cases that contain specific conditions and expected results for the following requirement.
 - Data – [Insert requirement here]
 - Format – The test case must show test conditions and expected results
 - Constraints – Limit the number of test cases to the minimal amount needed to verify the requirement.
- Verify the output. Did it match your expectations?

19



19

EXPECTED OUTPUT

To cover the logic and boundary values for this requirement with the minimum number of tests, here are the essential test cases:

Test Case	Test Condition (Weight)	Expected Result (Postage)
TC-01	0.5 lbs (Under 1 lb)	\$10.00
TC-02	1.0 lbs (Lower bound: 1 - 4.99)	\$15.00
TC-03	4.99 lbs (Upper bound: 1 - 4.99)	\$15.00
TC-04	5.0 lbs (Lower bound: 5 - 9.99)	\$20.00
TC-05	9.99 lbs (Upper bound: 5 - 9.99)	\$20.00
TC-06	12.0 lbs (Over 10 lbs)	\$35.40 (\$2.95 x 12)

Note: There is a logical gap in the requirement for weights between **10.00 lbs and 10.04 lbs** (or any value precisely at 10.0). I have assumed TC-06 covers the "Over 10" rule, but you may want to clarify the handling of exactly 10 pounds with your stakeholders.

Would you like me to add **negative test cases** for invalid inputs like zero or negative weight?

NOTE: Many times, output from AI will not be totally predictable. However, in this case, you should see something similar to this.

Also, note that a defect in the requirement was detected.

As an additional step, the AI is asking if we want to see negative tests. Let's say "yes."

20



20

EXPECTED OUTPUT (2)

To ensure the system handles invalid data gracefully without crashing or returning incorrect rates, add these negative test cases: 

Test Case 	Test Condition (Input)	Expected Result
TC-07	-5.0 lbs (Negative weight)	Error Message: "Invalid weight entered."
TC-08	0.0 lbs (Zero weight)	Error Message: "Weight must be greater than zero."
TC-09	"Ten" (Non-numeric input)	Error Message: "Please enter a numeric value."
TC-10	Leave Blank (Null input)	Error Message: "Weight is required."

21



21

EXPECTED OUTPUT (3)

Key Considerations for Negative Testing:

- **Error Messages:** Verify that the [system response](#) provides a clear, user-friendly message rather than a generic system crash or "NaN" (Not a Number) result.
- **Data Integrity:** Ensure that [invalid inputs](#) do not trigger any backend calculations or accidentally save a \$0.00 rate to a database.
- **Boundary Gaps:** As noted earlier, the requirement doesn't explicitly define what happens at **exactly 10.0 lbs**. Depending on the system's precision, you might want to add a test for **10.00 lbs** to see if it defaults to a specific tier or throws an error.  Qase +1

Should we also include **extreme value testing**, such as a weight of 1,000,000 lbs, to check for **buffer overflow** or calculation limits?

The AI is now showing the rationale for the negative tests. This is a great thing!

Let's also ask for the extreme value tests.

22



22

EXPECTED OUTPUT (4)

To ensure the system doesn't crash or behave unpredictably under heavy load or massive inputs, add these extreme value test cases. These are designed to check for **buffer overflows, integer limits, and logical constraints**.

Test Case	Test Condition (Input)	Expected Result
TC-11	1,000,000 lbs (Large numeric value)	\$2,950,000.00 (Verify the system can handle large calculations)
TC-12	99,999,999.99 lbs (Upper UI limit)	Verify the UI doesn't "break" and the total is calculated correctly.
TC-13	0.000001 lbs (Extremely small positive)	\$10.00 (Verify it rounds/handles high precision correctly)
TC-14	10.00001 lbs (Just over the edge)	\$29.50 (Verify the trigger for the per-pound rate)

Now, it is interesting that in this output, the AI has assumed that the UI is 8-digit integer and 2 positions decimal. We never told it that!

That is an example of a "hallucination."

However, we could refine our prompt to convey the correct field size.

23



23

EXPECTED OUTPUT (5)

Why these matter:

- **Buffer Overflow:** If the weight field is stored as a small data type (like a 16-bit integer), entering a massive number like **TC-11** or **TC-12** could cause the system to [overflow](#) and [crash](#) or return a negative price.
- **Rounding & Precision:** **TC-13** ensures that even a microscopic weight is charged the minimum base rate rather than being rounded down to zero (which would trigger an error).
- **Physical Reality:** While a 1-million-pound package is physically impossible for most carriers, the [software logic](#) must still handle the number gracefully unless a **maximum weight limit** (e.g., 70 lbs for USPS) is explicitly added to the requirements.

Since we've hit the extremes, would you like me to draft a **Requirement Clarification** document for the "missing" 10.0-pound rule to give to your product owner?

Once again, we see the rationale for the test cases.

Let's also generate the requirements clarification document.

24



24

EXPECTED OUTPUT (6) – STAKEHOLDER CLARIFICATION DOCUMENT

1. Issue Description

During the creation of test cases for the postage rate requirement, a "dead zone" was identified in the logic. The current requirements define rates for weights up to **9.99 pounds** and for weights **over 10 pounds**, but they do not specify the rate for a package weighing **exactly 10.0 pounds**.

2. Current Requirement Mapping

- **Tier 3:** 5 lbs to 9.99 lbs = \$20.00
- **Gap:** 10.00 lbs = [No Rate Defined]
- **Tier 4:** Over 10 lbs = \$2.95 per pound

3. Proposed Clarifications

To ensure [consistent software enforcement](#), please confirm which logic should be applied:

- **Option A (Extend Tier 3):** Change Tier 3 to "5 pounds to **10.00 pounds**."
- **Option B (Inclusive Tier 4):** Change Tier 4 to "**10.00 pounds or more**."
- **Option C (Precision update):** Change Tier 3 to "5 pounds to **less than 10 pounds**" and Tier 4 to "**10 pounds and above**."

25



25

EXPECTED OUTPUT (7) - STAKEHOLDER CLARIFICATION DOCUMENT

4. Impact if Unresolved

Without clarification, the system may default to an [unintended state](#), potentially resulting in:

- \$0.00 postage for a 10lb package.
- System calculation errors or "Unhandled Exception" crashes.
- Customer disputes due to inconsistent pricing at the 10lb mark.

Action Required: Please specify the intended postage for a weight of **exactly 10.0 pounds** so the test cases and development logic can be finalized.

26



26

LESSON 2

HOW AI SYSTEMS ARE DIFFERENT

27

LEARNING OBJECTIVES

- Differentiate between deterministic vs. probabilistic behavior.
 - Why the same input may not produce the same output.

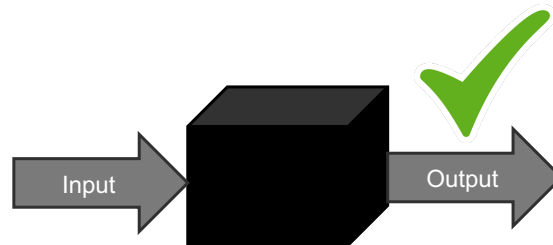


28

28

IN TRADITIONAL SYSTEMS

- **Deterministic** - Input led to predictable output.



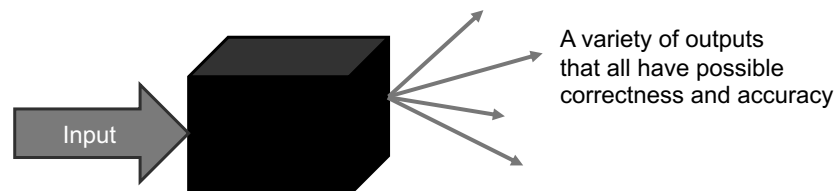
29



29

IN AI

- **Input may lead to many differing outputs**
 - Some or all may be acceptable ("correct")
 - Others may be totally incorrect.
- **Instead of determining correct behavior, we need to consider the probability that the result is correct.**



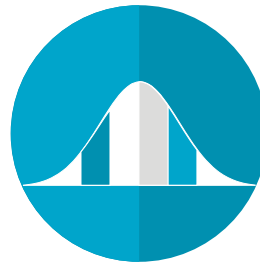
30



30

WHAT THIS MEANS...

- We often can't define an exact expected result.
- Instead, we have to define what constitutes an acceptable result.



31



31

EXAMPLE

- **Deterministic behavior:**
A sales tax calculator receives a purchase amount of \$100 and a tax rate of 8%.
 - Every time you run it, it should return **\$108**.
 - Same input → same output.
- **Probabilistic behavior:**
Ask a generative AI system, *"Summarize this customer complaint in one sentence."*
 - Run the same request several times, and you may get slightly different summaries.
 - They may all be acceptable - or some may be better or worse than others.
 - Same input → potentially different outputs.

32



32