



What to Test in AI-Based Systems Micro-Credential Syllabus

Copyright Notice

Copyright AT*SQA, All Rights Reserved

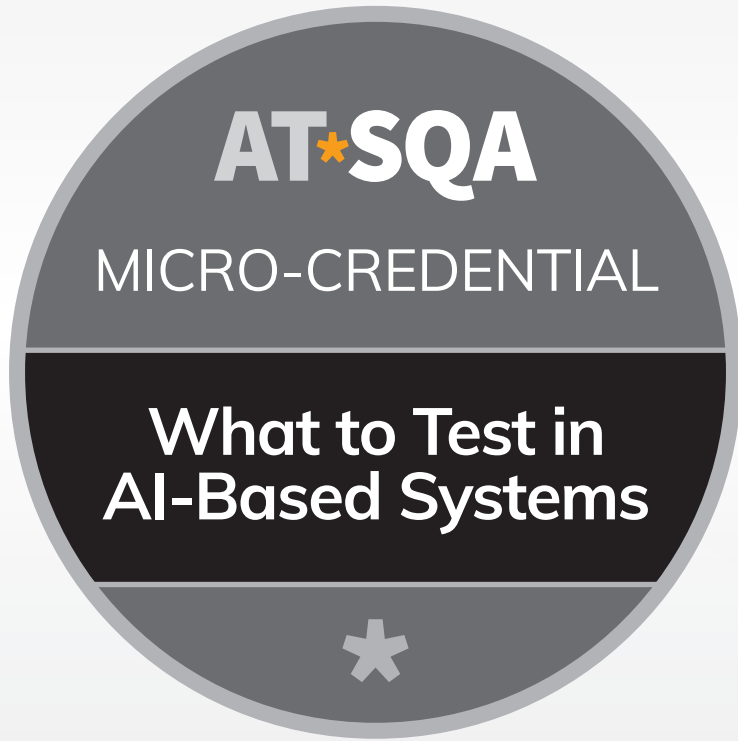


Table of Contents

What to Test in AI-Based Systems

3	Why AI?
5	Keywords
6	Learning Objectives
7	Managing Risks
11	Data Privacy and Security Considerations
15	Quality Characteristics
23	Final Thoughts
24	Appendix A: References
24	ISTQB® Documents
24	Glossary References
24	Standards
25	Appendix B: Glossary

Why AI?

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 AI is a large and emergent field, and any syllabus on the subject will be quickly outdated.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

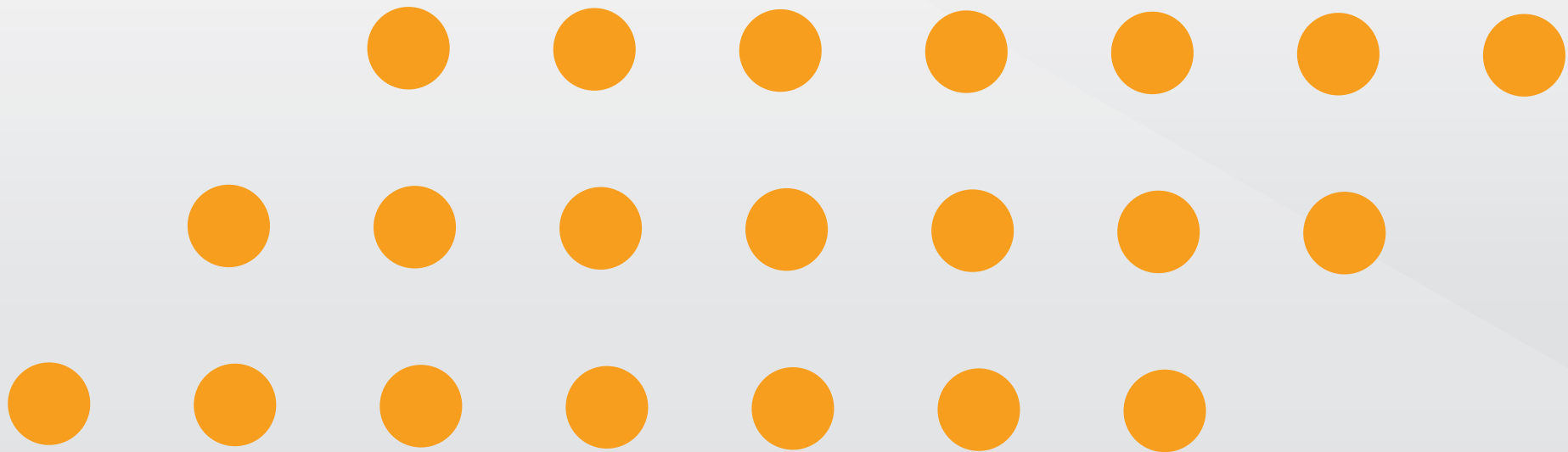
AI, in general, has a plethora of new terms associated with it. The first module of this stack of micro-credential syllabi acquaints the tester with the AI terminology

and the basics of AI. It will soon be an expectation that all testers are “familiar” with AI, and that will require having a basic understanding and the ability to learn as AI evolves. The “Introduction to AI” module covers the basics and touches on a bit of the technology behind AI. This is not intended to be a full and technical explanation of AI; there are many books already devoted to that.

This module, “What to Test in an AI-based System” discusses how to identify the risk items and conditions that must be tested. It explains data privacy and security concerns with AI-based systems, and it highlights the quality characteristics that are unique to AI-based systems.

The third module, “How to Test an AI-based System” explains how the traditional test levels can be applied to testing AI-based systems. It also discusses testing techniques that are particularly well-suited for AI system testing. The module includes a discussion of the challenges of testing AI systems.

The fourth module, “Using AI to Test”, explores how to use AI to help with testing. This looks at building test cases, building test automation, recording results, and automating many manual aspects of testing. Along the way, caveats are explored including what should be considered when AI is used. As the world has seen, AI is not always accurate.



Keywords

adaptability, autonomy, bias, drift, evolution, ethics, explainability, explainable AI (XAI), flexibility, hallucination, interpretability, non-deterministic, reasoning error, reward hacking, transparency

LEARNING OBJECTIVES FOR WHAT TO TEST IN AI-BASED SYSTEMS

Managing Risks

- (K2) Explain the causes and symptoms of hallucinations, reasoning errors and bias
- (K2) Describe methods for testing for hallucinations, reasoning errors and biases
- (K2) Summarize issues that a tester can encounter with non-deterministic systems

Data Privacy and Security Considerations

- (K2) Explain the data privacy considerations with AI systems
- (K2) Understand the security risks and mitigation strategies in AI systems

Quality Characteristics

- (K2) Explain how flexibility and adaptability affect testing AI systems
- (K2) Explain how system evolution impacts testing
- (K2) Explain how bias and ethics can impact an AI system
- (K2) Summarize how side effects and reward hacking affect AI systems
- (K2) Explain how transparency, interpretability, and explainability affect user trust

Managing Risks

In its purest form, testing is a risk mitigation exercise. This is no different when testing software that has incorporated AI components, but the extent of mitigation is different. In conventional software, given controlled pre-conditions and data, a risk can be identified, assessed and mitigated by testing. In AI, even when controlling the pre-conditions and the data, the software may give varying results each of which has to be evaluated for correctness. In the AI world, rather than risk mitigation, risk management is a more accurate term.

Hallucinations

A hallucination occurs when an AI model produces an output that is either incorrect or misleading, but is presented as correct. There can be many reasons for a hallucination to occur, including insufficient training data, biases in the training data, invalid training data, or incorrect assumptions by the model.

If we are using AI for generating test cases, a hallucination might manifest as a test case with missing or invalid steps, a test case that verifies acceptance criteria that don't exist, or a test case that expects a result that can't occur.

Reasoning Errors

Reasoning errors occur when the AI system applies the incorrect logic to resolve a problem. This causes the results to be wrong or illogical. Because AI relies on pattern matching rather than human logic, it's not able to consider information beyond its training. For example, we might know that a test case has a high priority because of past failures in that area, but if the AI system does not consider past failures to be a significant consideration, the resulting test case prioritization may be wrong.

Bias

A bias occurs when the AI system produces results which are prejudiced or unfair. This is usually due to an inherent bias in the training data but could also be due to the algorithm being designed incorrectly. For a testing example, this could occur if the AI system produces an unbalanced set of test cases, favoring test cases that test reports over test cases that test input functions. This can result in a skew of the testing toward the areas the AI system has “preferred”.

Testing for Hallucinations, Reasoning Errors and Biases

According to the CT-GenAI syllabus, the following recommendations are provided for testing for these risks:

Hallucination detection:

- **Cross-verification** – compare the AI output with existing sources of information (e.g., documentation, requirements, expected system behavior)
- **Ask an expert** – Just as with manual testing, have an expert check the result to ensure it is correct
- **Check consistency** – Ensure that the outputs are consistent with each other, although this needs to be manually verified because they could all be consistently wrong

Reasoning error detection:

- **Validation of the logic** – Validate the output for coherence and consistency
- **Output testing** – Use the output to see if it gives the expected result (e.g., run a generated test case and check the result)

Bias detection:

- **Check for balanced outcome** – check the outcome to verify that it is representative of the real world (e.g., is generated training data properly balanced)
- **Check balance between types of outputs** – verify that one type of output (e.g., negative test cases) is not overrepresented

If the training data is well-balanced and representative, and the internal logic of the AI system is working correctly, errors may be being introduced via the prompts to the system. Prompts that are lacking relevant information, are unclear, or are too complex can also result in incorrect output. Complex prompts may need to be submitted in stages so the output from one prompt serves as the input to the next one.

Dealing with Non-Deterministic Behavior

The non-deterministic behavior of an AI system is definitely a risk that must be managed. Non-deterministic means that a system can produce different outputs even when given the exact same inputs. The outcome cannot be reliably predicted. For testers who are accustomed to defining and verifying the expected result, this presents a new challenge. It also challenges traditional test automation which is looking to confirm that the output is correct based on a specified value or behavior.

While there are some settings with an AI system that can be used to reduce the range of outputs (e.g., lowering the temperature to narrow the distribution of results or setting starting values for random number generators), these will result in testing a system that is not configured the same way it will be used in production.

In addition to the inherent non-deterministic nature of AI, there is also the concept of drift which means

as the model learns, the outputs may degrade. Drift can result in predictions being inaccurate, bias being introduced, or poor decision making. In this case, a model will need to be retrained in order to improve its accuracy.

Testing non-deterministic systems is difficult because the answers must be evaluated for correctness. For example, if you were expecting an AI system to develop 10 test cases off a set of requirements and you received 20, is it wrong? The 20 test cases will need to be evaluated to see if they are correct and not redundant. While AI can generate information very rapidly, validating the output can be tedious and difficult to automate.



Data Privacy and Security Considerations

AI uses a significant amount of data for training, processing and output. Because AI may draw its information from many sources, data privacy is a significant consideration. Similarly, protecting the data from security vulnerabilities is also critically important.

Data Privacy Risks

AI works on the basis of consuming large amounts of training information that is then used for determining outcomes. The data that is consumed is usually from a variety of sources and may include data that should be protected, such as sensitive (e.g., health information) or personally identifiable data (e.g., birth dates, social security numbers). With pre-trained models, the source of the data that has been consumed in training may be unknown.

According to CT-GenAI, the following data privacy concerns must be considered, both regarding the data being processed as well as the data that was used for training:

- **Unintentional data exposure** – the model may include sensitive information in its outputs
- **Lack of control over data usage** – Sensitive data may be stored and processed by the AI tools without having acquired explicit user consent. This opens the door for potential misuse or access by unauthorized people or systems
- **Compliance risks** – there are strong data protection regulations (e.g., GDPR) that, if violated, can result in legal action

When testing AI systems, it's important for the tester to review the type of data that is being output to ensure that any private or sensitive data is protected and used appropriately. This may also require data masking and other activities to protect the data from exposure during development and testing.



Security Risks

There are some security risks that are unique to AI systems. The most obvious risk is that if AI as a Service (AlaaS) is used, security risks may already exist in that system and those risks will be inherited. Similarly, using a pre-trained LLM will inherit any vulnerabilities and data issues that are already in that LLM. Unique to AI though, attackers may intentionally introduce false or erroneous data that will mislead the LLMs and potentially result in accuracy issues.

The following table, from (CT-GenAI), provides examples of attack vectors. Security testing must be conducted to address these vulnerabilities.

Attack vector	Description	Example
Data exfiltration	Sending requests designed to extract confidential training data.	Exceeding the LLM contextual window with long prompts to overload the AI's memory could lead it to reveal random snippets of its training data and potentially expose sensitive information.
Request manipulation	Introducing data that disrupts the AI's output.	Images that lure the AI into a different context, thus provoking hallucinations on e.g., acceptance criteria.
Data poisoning	Manipulating training data.	Providing fake evaluations when rating the results of an AI-generated test report.
Malicious code generation	Manipulating an LLM to generate backdoors (e.g., external command calls) during use.	Generation of code to open a communication channel with a specific, malicious IP.

Risk Mitigation Strategies

Security and data privacy risks are inherent in the application of AI. It is important to implement strong controls, particularly around data privacy, to help mitigate these risks. The following measures are proposed in (CT-GenAI):

- **Data minimization** – Only acquire, use and store the absolute minimum amount of data needed. Ideally, avoiding using sensitive data at all. Be sure that any data usage is within the privacy laws of the operating environment.
- **Data anonymization** – Ensure that any sensitive information is used only when masked, or replace the sensitive data entirely. For example, do you really need to use valid birth dates?
- **Secure data handling** – When data is stored or transmitted, it must be done securely using encryption and access control.
- **Train resources** – Ensure that all resources who will be handling data are trained and understand the importance of protecting the data.
- **Securing the LLM** – There may be an option to operate the LLM in a secure cloud or on premise to increase the security of the data and the activities of the LLM.

Quality Characteristics

The CT-AI syllabus discusses the quality characteristics that should be addressed when testing AI systems. Quality characteristics are usually the focus of testing and this is no different with AI systems, but the actual characteristics differ from traditional systems. This requires new approaches to testing and a thorough understanding of the system under test. Testing AI systems requires judgement from the tester because the acceptance criteria are rarely defined in testable terms.

The quality characteristics discussed in this section include the following:

- Flexibility and adaptability
- Autonomy
- Evolution
- Bias
- Ethics
- Side effects and reward hacking
- Transparency, interpretability and explainability
- Safety

Each of these will be discussed in more depth.

Flexibility and Adaptability

Flexibility and adaptability in AI systems are an expectation, but that doesn't mean it's easy to test. Understanding the real requirements is tricky. How flexible? What adaptation is expected vs not allowed? The CT-AI syllabus defines the terms as follows:

- Flexibility – the ability of the system to be used in situations that were not part of the original system requirements
- Adaptability – the ease with which the system can be modified for new situations

When these “situations” are not defined, the tester is left to determine what might happen in the future. While this is patently unclear, the tester still has to use their best judgement to understand the target domain, technology trends, and other variables to devise realistic tests.

In general flexibility and adaptability are needed when the operational environment may change, when new situations may be encountered and when the system will be expected to change its behavior based on what it has

learned. It's generally easier to understand and anticipate environment changes. For example, an autonomous car's environment will likely be restricted to roading systems and slowly changing road rules. However, with climate change, perhaps the likelihood of encountering flooded roads increases, so that's an area where more testing may be needed.

Flexibility and adaptability come down to understanding the environment and the expected usage of the system in the future. Working with the wider development team becomes extremely important to understanding what is anticipated. That can be the starting point for testing, but it must extend beyond that because the system is more likely to break, or give incorrect answers/ predictions, when the unanticipated happens.

One measurable item for testing in this area is to determine the time and resources the system will require when it needs to adapt. If the new environment can be created, the system's response time in adapting can be measured and this can be used to understand the level of adaptability and flexibility the system can demonstrate, and at what cost.

Autonomy

A true fully autonomous system is independent of human control. It does not allow or need oversight. In general, this is not a desirable state because human intervention and oversight are expected to be needed to adjust the course an autonomous system may be taking. In general usage, an “autonomous” system uses multiple inputs into the ML system (such as sensors on self-driving cars) that are then used for the ML model to make decisions and act on those decisions. While these actions are autonomous, there is still an element where a human can make adjustments and “teach” the system where it is making mistakes.

In general, when testing autonomy, the tester is measuring how long the system can function correctly without needing any human intervention. Autonomy testing also includes verifying that a human can take control when needed.

Evolution

AI systems must be able to improve themselves as constraints and influences change. This is called evolution. Self-learning systems that apply that learning to their behavior are “evolving”.

There are generally two ways the self-learning AI system normally evolves. These are as follows:

- The system learns from the decisions it makes and the interactions it has with the environment. This learning is applied to future decisions and interactions.
- The system learns from changes that have been made to its operational environment and adapts accordingly.

As the system learns and evolves, it should be improving. This is usually measured in terms of effectiveness in accomplishing its tasks and the efficiency with which it performs its tasks. A system can, of course, change in a negative way and become slower or less accurate. While this is

still an evolution, it's not a good one. By measuring if the system is meeting or exceeding its original acceptance criteria, the tester can determine if positive evolution is occurring.

To test that evolution is positive and the processing is still correct, re-applying the test data can be helpful. By comparing the results from the test data from the initial launch of the system to the evolved system's outputs, the tester can determine if the evolution has been generally correct and positive. Any regressions at this point indicate problems and it's likely that corrections to the ML model will be needed.



Bias

According to CT-AI, bias is a statistical measure of the distance between the system's outputs and outputs that would be considered to be "fair". Biases are considered inappropriate when they are related to attributes such as income level, age, gender, race, ethnicity, etc.

Bias can result from two primary issues.

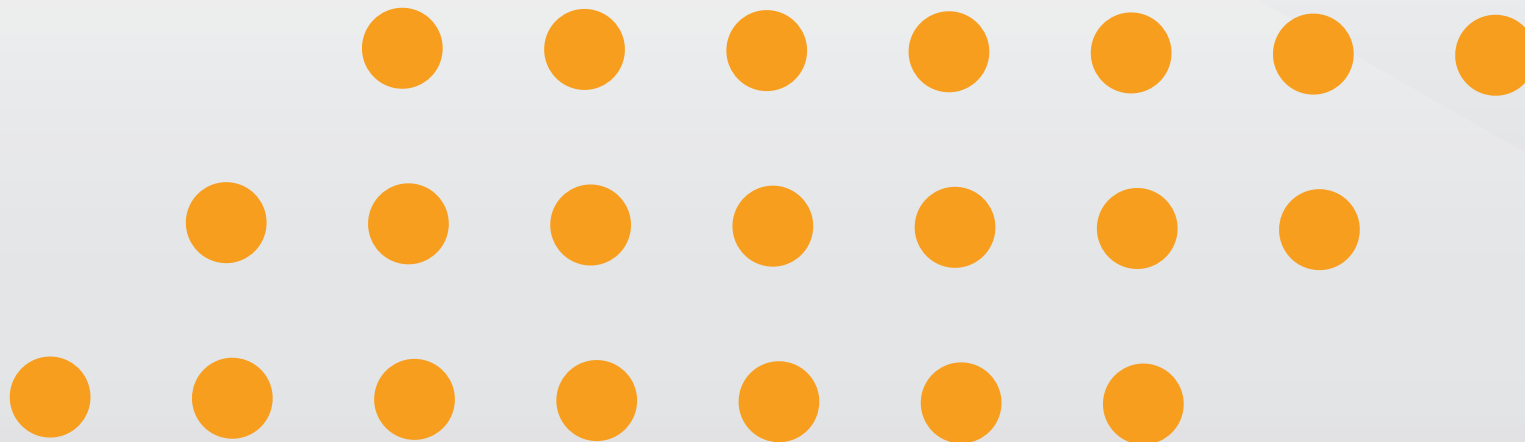
- **Algorithmic bias** – This type of bias occurs when the learning algorithm itself was not configured correctly. This could be due to some data being overvalued or preferred by the model.
- **Sample bias** – This type of bias occurs when there is a mismatch between the training data and operational data. It can also occur when there is insufficient training data.

Algorithmic bias is usually managed by hyperparameter tuning of the ML algorithms. Sample bias is usually fixed by either providing additional training data, or starting over with the training using more representative data.

Testing for bias requires an understanding of what “fair” is for the system. For example, making bank lending decisions based on income level may be valid, but determining who is eligible to apply for a job based on income may be irrelevant, and therefore unfair. The tester must be on the alert for decisions that appear to be based on bias. Selecting a wide variety of data for the testing set will help to identify these types of issues early, but continuous monitoring is also needed because the system may “learn” bias.

Ethics

What is ethical can be difficult to define and may change over time. The Cambridge Dictionary defines ethics as: a system of accepted beliefs that control behavior, especially such a system based on morals. Merriam-Webster defines ethics as: a set of moral principles. Morals are a human quality, not a system quality. It’s up to the humans who train, test and deploy an AI system to ensure that it behaves “morally” and is used in a moral manner.



AI ethics are being defined worldwide which seek to define moral development and use of AI systems. ISO has defined the following key principles of AI ethics (ISO):

- **Fairness:** Datasets used for training the AI system must be given careful consideration to avoid discrimination.
- **Transparency:** AI systems should be designed in a way that allows users to understand how the algorithms work.
- **Non-maleficence:** AI systems should avoid harming individuals, society or the environment.
- **Accountability:** Developers, organizations and policymakers must ensure AI is developed and used responsibly.
- **Privacy:** AI must protect people's personal data, which involves developing mechanisms for individuals to control how their data is collected and used.
- **Robustness:** AI systems should be secure – that is, resilient to errors, adversarial attacks and unexpected inputs.
- **Inclusiveness:** Engaging with diverse perspectives helps identify potential ethical concerns of AI and ensures a collective effort to address them.

A number of these areas provide more test conditions for the tester to examine. Working with the developers of the system to ensure ethical behavior is built-in will save a lot of time that would have to be spent re-training the system later. As with other quality characteristics, understanding the intention and operational environment of the system will help to identify which of these characteristics must be validated during testing.

Side Effects and Reward Hacking

Testing for unintended side effects can be difficult because they are, by definition, unintended and therefore undefined. These unintended results become an issue when the side effect is classified as “negative”. For example, a chatbot might have been designed to improve customer satisfaction, but it may have the negative side effect of customer support people losing their jobs. The best way to combat negative side effects is to anticipate and plan for them. Similarly, positive side effects can be a problem if they become expected and are used. If they are subsequently

removed, this can create problems for anyone dependent on those side effects.

Reward hacking occurs when the system determines a more expeditious and unintended way to achieve the goal. A commonly used example is when the AI system is designed to play a game with the intention of achieving the highest score, but, rather than play the game, the system just updates the data that records highest scores. In this way the reward is achieved, but the anticipated method for achieving it was not followed. Testing for these loopholes requires examining the prompts that could be used and ensuring the system is following the expected steps to achieve the goal.

Both of these areas are difficult to test and are problems that are usually observed in the operational environment rather than during testing. Good design is the best way to combat these quality issues, and that requires the tester to be asking good questions during the design sessions to ensure these issues have been considered and prevented.

Transparency, Interpretability and Explainability

While many users are using AI-based systems as black boxes, people are becoming more aware that trust in the system is necessary. This is obviously true in the case of safety-critical systems and systems handling sensitive data, but as AI becomes more commonly used in many capacities, the trust issue is becoming more important.

In order to build trust, there must be the following:

- **Transparency** – The users must be able to understand the inner workings of the AI models, such as how the model was trained, how it learns, and how it generates its outputs.
- **Interpretability** – The users must be able to understand the relationships between the inputs and outputs of the model. The user should be able to understand why a model made a particular prediction based on the inputs it received.
- **Explainability** – Model decisions should be clearly explained and understandable by the intended users (e.g., developers, end users, regulators).

Together, these three are the key characteristics of Explainable AI (XAI). XAI is a field that is working to allow users to understand, trust and participate in improving AI models. While these characteristics are important for building trust with the user community, they are also important to allow testers to understand what the system is doing and why it is doing it. The more transparent, interpretable and explainable a system is, the easier it is to test because the outputs can be predicted more accurately. Systems that are easier to test are also easier to maintain and, in some cases, easier to develop.

Safety

While safety may be expected by the users, it will not happen by default. This is another area where design and implementation of safety are critical and where testing is required. The goal of a “safe” system is that it will not cause harm to people, property or the environment. As AI-systems become more entrenched in our lives, they have a higher potential to affect safety.

Safety is difficult to test. As has been discussed, the very nature of AI-based systems makes them hard to predict. As with all quality characteristics, safety has to be designed into the system, including safeguards to allow human intervention if needed. As a minimum, the tester needs to review the following to assess the potential for safety issues:

- Complexity of the system
- Variability in the predictions (non-determinism)
- Probabilistic nature of the system
- Self-learning capabilities and the ability to monitor the learning that has occurred and is being applied
- The amount of transparency, interpretability and explainability in the system
- Robustness of the system in the face of unexpected circumstances

Final Thoughts

It is important for the testing effort that the team understands that testing AI systems is different from testing traditional systems. The characteristics of the systems make testing more difficult and more time consuming. AI systems also require testing after deployment, which requires an ability to do periodic (if not continuous) testing in the production environment.

It is important for the entire development team to understand the need for testing and to work together to build a testable system. As was explained in this section, some of these testable characteristics, such as transparency, must be designed into the system. Careful scrutiny of the training, evaluation and test data sets is needed to avoid and detect such issues as bias.

Even more than in traditional systems, testing cannot be an afterthought. Because it is still a relatively new field, understanding the needs and requirements for the test effort must be built within the entire team.

Appendix A: References

ISTQB® Documents

[CT-AI] ISTQB Certified Tester – Artificial Intelligence Syllabus, v1.0, 2021

[CT-GenAI] ISTQB Certified Tester – Generative AI Syllabus, v1.0, 2025

Glossary References

ISTQB® Glossary <https://glossary.istqb.org/>

Standards

ISO. (n.d.). Responsible AI. Retrieved from <https://www.iso.org/artificial-intelligence/responsible-ai-ethics>

The previous references point to information available on the Internet and elsewhere. Even though those references were checked at the time of publication of this syllabus, AT*SQA cannot be held responsible if the references are not available anymore.

Appendix B: Glossary

This glossary is composed of excerpts from the [CT-AI] and [CT-GenAI] syllabi. Refer to those syllabi for additional terms and references.

adaptability: The ability to learn, evolve and adjust behavior based on new data and changing environments

autonomy: The ability to operate, make decisions, and execute tasks with little or no human intervention

bias: The systematic difference in treatment of certain objects, people or groups in comparison to others (ISO/IEC DIS 22989)

drift: The degradation of a model's performance over time due to changes in the real-world data it processes

evolution: The process of continuous change from a lower, simpler, or worse state to a higher, more complex, or better state

ethics: A set of moral principles guiding the development, deployment, and use of artificial intelligence to ensure it is fair, transparent, accountable, and aligns with human values

explainability: The level of understanding how an AI-based system came up with a given result (ISO/IEC TR 29119-11)

explainable AI (XAI): The field of study related to understanding the factors that influence AI system outputs

flexibility: The ability of a system to work in contexts outside its initial specification (After ISO/IEC TR 29119-11)

hallucination: Wrong information created by an LLM

interpretability: The level of understanding how the underlying AI technology works (ISO/IEC TR 29119-11)

non-deterministic: A system which will not always produce the same set of output and final state when given a particular set of inputs and starting state

reasoning error: An instance where AI, while using individual facts that might be correct, applies the wrong logic or reasoning steps to a problem, leading to a faulty or illogical conclusion

reward hacking: The activity performed by an intelligent agent to maximize its reward function to the detriment of meeting the original objective (After ISO/IEC TR 29119-11)

transparency: The level of visibility of the algorithm and data used by an AI-based system (After ISO/IEC TR 29119-11)

Purpose of this Document

This syllabus forms the basis of the AT*SQA certification for AI for Testers. AT*SQA is an International Standards Organization (ISO) compliant certification body for software testers. AT*SQA provides this syllabus as follows:

1. To training providers - to produce courseware and determine appropriate teaching methods.
2. To certification candidates - to prepare for the exam (as part of a training course or independently).
3. To the international software and systems engineering community - to advance the profession of software and systems testing, and as a basis for books and articles.

AT*SQA may allow other entities to use this syllabus for other purposes, provided they seek and obtain prior written permission.

This syllabus has been constructed to be tool-agnostic, but tools will be discussed where they are commonly used. When tools are referenced, this is not a recommendation for the use of a particular tool, but examples of commonly used tools to help clarify points.

There are no prerequisites required for this certification. The certification can be achieved by passing the assessments for all four AI micro-credentials. As a part of AT*SQA's ISO compliant offerings, the certification must be kept current with additional learning completed within the defined timespan. This helps software testers to continue to expand their knowledge and marketability and acknowledges the very real need for continuing education in the software testing industry. For more details, see AT*SQA's website.

This syllabus has been built from information in the ISTQB Certified Tester – Artificial Intelligence [CT-AI] and Certified Tester – Generative AI [CT-GenAI] syllabi, as well as information gathered from multiple other sources. This is an introduction to both of these syllabi, but it will not provide a full preparation for the ISTQB exams. For those certifications, it is best to study the full syllabi and consider attending some focused training.

AT*SQA

MICRO-CREDENTIAL

**What to Test in
AI-Based Systems**



www.atsqa.org

