


# QUALITY ENGINEERING IN THE AI ERA

From test execution to engineering decisions, risks and evidence.

JONAS DAVILA.

FREE E-BOOK

# CREDITS AND HOW TO READ

## QUALITY ENGINEERING IN THE AI ERA

From test execution to engineering decisions, risks and evidence.

Quality Insights · QI/01 · free introductory edition
Author: Jonas Davila da Silva
License: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0). You may share this guide with credit to the author; you may not sell it or distribute modified versions. https://creativecommons.org/licenses/by-nc-nd/4.0/

This guide brings together interpretations and recommendations to support your learning and work decisions. References identified throughout the text let you check the scope of claims attributed to sources.

| | |
|---|---|
| FACT | a referenced claim can be checked against the indicated source. The reference supports the cited point, not necessarily the entire section. |
| INTERPRETATION | a reading proposed by the author, identified in the text. It can be discussed in light of other contexts and evidence. |
| RECOMMENDATION | practical guidance, identified as such. You decide whether it applies to your context and what adaptations are needed. |
| HYPOTHETICAL EXAMPLE | a situation created to explain a decision. It does not describe a verified event or a result obtained by the author. |

The paths are possibilities, and the assessment is an exercise in reflection. Use the guide to formulate questions and produce evidence of your learning; consult the sources when you need to explore a technical decision further.

# CONTENTS


→ **BEFORE YOU BEGIN** 04

I **THE TRANSFORMATION OF QUALITY** 05

01 The future of quality is not more testing 06

02 The quality professional in the AI era 08

II **CAREER POSSIBILITIES** 10

Quality Engineer · SDET · AI Quality Engineer · AppSec Engineer from QE · Quality & Reliability Engineer · Quality Engineering Lead 11

III **QUALITY ENGINEERING AND AI** 17

03 AI to support testing · Testing systems that use AI 18

04 Risk-oriented quality 20

05 AIMA 2.0: from information to decision 22

IV **PROFESSIONAL DEVELOPMENT** 23

06 Skills for deciding and building 23

07 Professional self-assessment 25

08 A 90-day plan 27

→ **WHAT REMAINS OURS** 29

References 30

About the author 31

OPENING

# BEFORE YOU BEGIN

**A passing test tells you something about one execution. To decide whether a release deserves confidence, you also need to know what was tested, which risks were left out, and what evidence supports each choice.**

My view: quality means reducing risk with evidence. Artificial intelligence expands execution capacity; quality engineering guides decisions; evidence helps justify confidence. None of these removes the need for judgment.

This guide is for people who test, automate, develop, or lead and want to broaden their work. You can read it in sequence or choose a path according to your needs.

IF YOU ARE STARTING OUT

read chapters 1 and 2, explore the career paths, and choose one to guide a small project. Use the assessment to identify what to learn next, not to compare yourself with others.

IF YOU ALREADY WORK IN THE FIELD

start with the career map, compare your activities with the paths, and move on to the chapters on AI and risk. Then record a gap and an observable outcome for the next 90 days.

IF YOU LEAD

examine the decision dilemmas and the skills matrix. Use the assessment to guide voluntary development conversations, without turning it into a performance evaluation tool.

Adapt the plan to your available time and context. The value of the exercise lies in explaining a decision and its limits, not in filling every page.

PART I

# THE TRANSFORMATION OF QUALITY


01 The future of quality is not more testing 06

02 The quality professional in the AI era 08


> **A RELEASE CAN PASS MANY CHECKS AND STILL LEAVE AN IMPORTANT QUESTION UNANSWERED.**

CHAPTER 1

CHAPTER 01

# THE FUTURE OF QUALITY IS NOT MORE TESTING

**An approved release can reveal another person's data when an access rule is left out of the checks. The success signal and the expected protection point to different decisions.**

The choice of check depends on the consequence to be avoided. If the access rule is unclear, the team needs to clarify it with someone who knows the product before deciding what response would be acceptable.

A release can pass many checks and still leave an important question unanswered. A test result applies to the conditions in which it ran; the release decision also involves what was not observed.

pipeline · latest run · hypothetical example

| | |
|---|---|
| Automated tests | passed |
| Measured coverage | high |
| **User report** | **"I can see someone else's order"** |

HYPOTHETICAL EXAMPLE

The pipeline (the automated integration and delivery workflow) is green, the team maintains many tests, and measured coverage is high. Even so, a user changes an order identifier and sees another person's data.

The green signal was not false: the checks that ran passed. The problem is that this signal alone did not answer whether each person was authorized to access that object.

OWASP describes broken object-level authorization and recommends checking authorization in functions that access records using client-supplied data. It also recommends tests to evaluate this mechanism and not deploying changes that cause them to fail [R02].

The useful question is not only how many tests exist, but what decisions each set of tests can support. A suite can be extensive and still fail to represent a relevant risk.

CHAPTER 01 · CONTINUED

# EVIDENCE NEEDS CONTEXT

**In my view, risk is the possibility that an undesirable situation will cause a relevant consequence in a particular context. This formulation serves the exercise in this guide; it is not intended to replace a normative definition.**

Start with the consequence: who might be affected, what might happen, and how the team would notice the problem. Then formulate a testable hypothesis and choose evidence proportional to the decision.

In the previous page's scenario, the decision requires more than repeating the permitted access path. The team needs to check access to objects belonging to another person, record the response, and examine where authorization is applied [R02].

Evidence also has limits. A test run with specific data and permissions does not demonstrate the behavior of every possible combination. Record the environment, conditions, result, and gaps so someone else can interpret the signal.

ISO/IEC 25010:2023 presents a product quality model with nine characteristics [R07]. Its existence helps remind us that a single indicator need not represent all of quality; that last statement is my interpretation.

Use counts and coverage to locate areas that need investigation. Do not use them alone as proof that a release is safe or ready.

CHAPTER 02

# THE QUALITY PROFESSIONAL IN THE AI ERA

**A suggested test can look complete while leaving out a rule that is decisive for the person using the product. The team needs to turn that response into a check it can explain.**

A useful response for the team must be open to challenge: another person should be able to find the rule used and understand why the expected result makes sense. This review can also reveal conditions that still need clarification.

Automating a check makes its execution repeatable under the programmed conditions. Using artificial intelligence to suggest checks adds another possibility: generating alternatives for a person to examine.

These activities call for complementary skills. Writing code and understanding the system remain useful; formulating a question, evaluating an answer, and recognizing what evidence lacks help guide the work.

> HYPOTHETICAL EXAMPLE
>
> A team asks a model for test suggestions for a discount rule. The response includes common cases but does not specify what to do when two rules overlap; the team consults someone who knows the rule and adds that condition to the analysis.

The suggestion is raw material. Before turning it into a test, check the requirement, expected result, and relevance of the data. If the expected result is wrong, an automated check may simply repeat that error.

Technical skills include reading code, data, application programming interfaces (APIs), continuous integration, and observability. Analytical skills include breaking down risks, distinguishing fact from hypothesis, and explaining the relevance of evidence.

CHAPTER 02 · CONTINUED

# WHAT AI EXPANDS AND WHAT CALLS FOR JUDGMENT

**Use AI to explore more alternatives when you can review the output and an appropriate policy governs the data supplied. The tool can help organize questions, but it does not automatically know your product's specific constraints.**

OWASP includes sensitive information disclosure, prompt injection, and misinformation among the risks associated with applications using language models [R03]. A fluent response does not confirm its correctness or authorize entering confidential information into the tool.

Before using a suggestion, ask: what is the source of the rule, who can confirm it, what data went into the request, and how can the response be checked? Also record uncertainty when there is insufficient reference material.

Judgment by responsible people remains necessary to prioritize consequences, accept risk, and decide what to do with incomplete evidence. Clearly communicating a limitation is part of technical competence.

Use assisted generation when it lets you formulate and review alternatives. Avoid treating an unverified response as a requirement, test evidence, or release authorization.

| AI SCALES | CALLS FOR HUMAN JUDGMENT |
| --- | --- |
| — the exploration of alternatives | — prioritizing consequences |
| — the organization of questions | — accepting risk |
| — the generation of suggestions for review | — deciding with incomplete evidence |
| | — communicating the limitation |

Summary of pages 8 and 9.

PART II

# CAREER POSSIBILITIES

## MAP OF CAREER POSSIBILITIES

**The following pages present six ways to contribute to quality: Quality Engineer, SDET, AI Quality Engineer, AppSec Engineer from QE, Quality & Reliability Engineer, and Quality Engineering Lead.**

These paths are neither a hierarchy nor a required progression. They are not a universal classification either: responsibilities, names, and boundaries vary with each team's context.

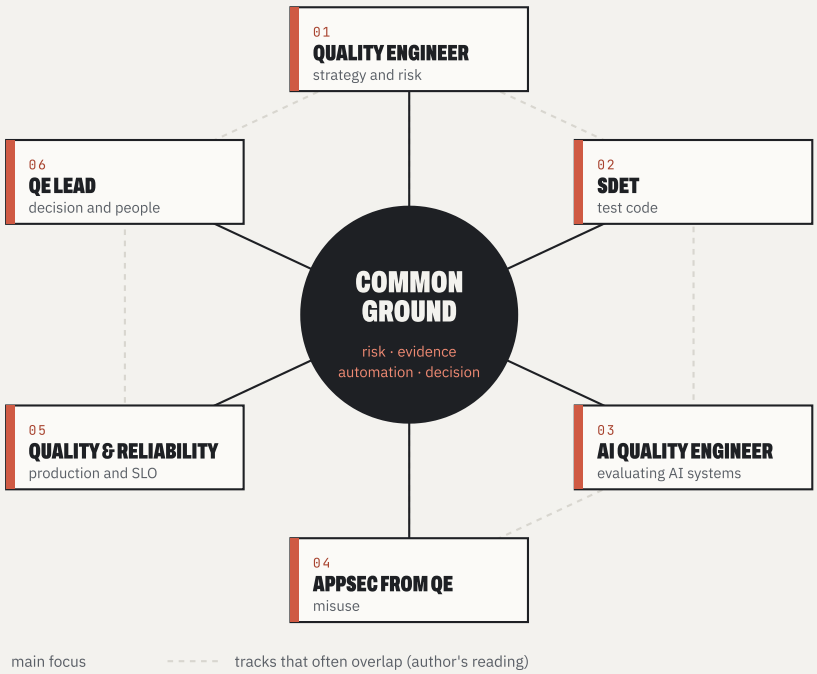
Choose by the kind of problem you want to investigate. You can combine practices from more than one path, deepen a skill without changing roles, or return to fundamentals before exploring a specialty.

In each profile, look at the problem, responsibilities, skills, and a portfolio project. The tools are starting points for study, not a mandatory list or a guarantee of competence.

> HYPOTHETICAL EXAMPLE
>
> A person who automates tests becomes interested in access permissions. They can study application security from their QE experience without abandoning automation or first going through another path.

Use the map to formulate a development hypothesis and validate that choice with a small project. Do not use it to measure a person's value by their job title.

MAP OF CAREER POSSIBILITIES FIG. 1




Not a hierarchy or a required path: the same person can move through several tracks.

TRACK 01 / 06

# QUALITY ENGINEER

**Investigates product risks and organizes checks that support release decisions.**

## PROBLEM IT SOLVES

When a rule changes without explicit expected behavior, the team may check paths that do not answer the release decision. Users may encounter registration errors that were not investigated before the change.

## RESPONSIBILITIES

- Understand rules and affected people.
- Identify risks with the team.
- Plan and run checks.
- Communicate evidence and limits.

## TECHNICAL SKILLS

Test strategy, APIs, data, and automation.

## BEHAVIORAL SKILLS

Curiosity, communication, and priority negotiation.

## TOOLS

Postman | Playwright | SQL

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | A team changes a registration rule. The QE identifies error conditions and records what remains to be checked before the decision. |
| PORTFOLIO | Document a registration rule, its risk hypotheses, and reproducible checks. |

## LEARNING PATH

01 Read the ISTQB CTFL syllabus to organize your study of testing fundamentals and formulate verifiable conditions.

02 Examine API1 in the OWASP API Security Top 10 and identify who can access each object.

03 Use the official Postman documentation to run requests and record expected and observed responses.

04 Consult the official GitHub documentation to present the project, execution conditions, and gaps.

Use this path to explore the problem and adapt your learning to your context. The title does not determine a mandatory career sequence.

TRACK 02 / 06

# SDET

**Develops software solutions to support repeatable tests and their maintenance.**

## PROBLEM IT SOLVES

Checks that are hard to repeat and maintain make the meaning of a suite failure uncertain. A change may reach the product without the team being able to distinguish a defect from a problem in the test itself.

## RESPONSIBILITIES

- Design automated checks.
- Review code and test data.
- Integrate checks into the delivery workflow.
- Investigate failures and suite maintenance.

## TECHNICAL SKILLS

Programming, test architecture, APIs, and continuous integration.

## BEHAVIORAL SKILLS

Collaboration, clarity, and failure investigation.

## TOOLS

Playwright | Selenium | GitHub Actions

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | A practice suite has intermittent failures. The person investigates execution conditions and records a hypothesis before changing the test. |
| PORTFOLIO | Create a suite for a practice application and document execution, failures, and limits. |

## LEARNING PATH

01 Study functions and error handling in the official Python documentation, applying them to a small check.

02 Read the ISTQB CTFL syllabus to connect test conditions to expected results.

03 Use the official Playwright documentation to build and run a suite for a practice application.

04 Consult the official GitHub Actions documentation to record execution and investigate workflow failures.

Use this path to explore the problem and adapt your learning to your context. The title does not determine a mandatory career sequence.

TRACK 03 / 06

# AI QUALITY ENGINEER

**Organizes the evaluation of systems that use artificial intelligence, considering data, responses, and risks.**

## PROBLEM IT SOLVES

Without clear evaluation criteria, a plausible response may be taken as correct even when it is ungrounded. This can lead a user to act on incorrect or insufficient information.

## RESPONSIBILITIES

- Define evaluation cases and criteria.
- Examine data representativeness.
- Investigate security and adversarial behavior.
- Communicate results and uncertainties.

## TECHNICAL SKILLS

Data, model evaluation, APIs, and security of applications using language models.

## BEHAVIORAL SKILLS

Critical thinking, communication, and collaboration with specialists.

## TOOLS

Python | Jupyter | MLflow

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | A study assistant responds based on supplied documents. The team checks whether responses are grounded and how the system reacts when information is missing. |
| PORTFOLIO | Assemble an evaluation set with fictional documents, criteria, and a record of responses. |

## LEARNING PATH

01 Read the ISTQB CT-AI syllabus to identify data and evaluation challenges for AI systems.

02 Study misinformation and data disclosure in the OWASP Top 10 for LLM Applications.

03 Use the official Python documentation to organize cases, responses, and evaluation criteria into a reproducible set.

04 Consult the official Jupyter documentation to present results and cases that require human review.

OWASP includes misinformation, sensitive information disclosure, and excessive agency among the risks of LLM applications [R03].

Adapt the path to your context; the title does not determine a career sequence.

TRACK 04 / 06

# APPSEC ENGINEER FROM QE

**Brings a testing perspective to application security, investigating how a system can be misused.**

## PROBLEM IT SOLVES

A functional check may confirm allowed access while leaving unauthorized access unexplored. A user's data may be exposed even with an approved delivery workflow.

## RESPONSIBILITIES

- Identify threats with the team.
- Translate risks into negative tests.
- Investigate code analysis alerts.
- Communicate evidence and limits.

## TECHNICAL SKILLS

HTTP, APIs, authentication, authorization, and dependency analysis.

## BEHAVIORAL SKILLS

Curiosity, communication without alarmism, and collaboration.

## TOOLS

OWASP ZAP | Semgrep | Burp Suite | Postman

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | The workflow is green and there are many tests, but changing an order identifier returns someone else's data. The investigation examines authorization applied to each object. |
| PORTFOLIO | Build a practice API with fictional data and positive and negative authorization tests. |

## LEARNING PATH

01 Study requests in the official Postman documentation, observing identifiers and credentials.

02 Read API1 in the OWASP API Security Top 10 on object-level authorization.

03 Use the OWASP ZAP documentation to explore a practice API and record negative tests.

04 Consult the GitHub documentation to document fictional evidence and test limits.

OWASP recommends checking authorization when a function accesses a record from client input [R02].

Adapt the path to your context; the title does not determine a career sequence.

TRACK 05 / 06

# QUALITY & RELIABILITY ENGINEER

**Connects the quality of changes to the behavior and recovery of services in operation.**

## PROBLEM IT SOLVES

A release may work along the usual path without the team knowing how it responds to a dependency failure. A user may lose access to the service while the team is still figuring out how to detect or recover from the problem.

## RESPONSIBILITIES

- Define indicators with the team.
- Investigate failures and recovery.
- Monitor operational signals.
- Support release decisions.

## TECHNICAL SKILLS

Observability, load testing, service objectives, and incident response.

## BEHAVIORAL SKILLS

Collaboration, uncertainty analysis, and communication.

## TOOLS

k6 | Prometheus | Grafana

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | A study team defines an objective for a service and observes failures in a simulated dependency. It records how it would detect the problem and what recovery it needs to verify. |
| PORTFOLIO | Document a practice service, its objective, and a recovery exercise. |

## LEARNING PATH

01 Read Embracing Risk from the Google Site Reliability Engineering book to study objectives and error budgets.

02 Consult the official Prometheus documentation to choose an observable signal for a practice service.

03 Use the official k6 documentation to plan a load check in a controlled environment.

04 Study the official Grafana documentation to present signals and limits concerning recovery.

The error budget can support decisions to continue or slow releases against a service target [R08].

Adapt the path to your context; the title does not determine a career sequence.

TRACK 06 / 06

# QUALITY ENGINEERING LEAD

**Helps people and teams align quality strategy, decisions, and skills development.**

## PROBLEM IT SOLVES

When quality priorities and responsibilities are unclear, a gap may have no owner in the release decision. A user may receive a change whose known risk was not discussed by the team.

## RESPONSIBILITIES

- Facilitate risk analysis.
- Align decision criteria.
- Support people's development.
- Review evidence and improvements.

## TECHNICAL SKILLS

Test strategy, continuous delivery, metrics, and systems.

## BEHAVIORAL SKILLS

Listening, negotiation, communication, and facilitation.

## TOOLS

GitHub | Grafana | Jira

| TRACK CARD | HYPOTHETICAL EXAMPLE |
|---|---|
| SITUATION | A team needs to decide which checks to prioritize for a change. Leadership brings the responsible people together to make risks, limits, and the decision explicit. |
| PORTFOLIO | Create a quality strategy for a fictional product, with risks, criteria, and gaps. |

## LEARNING PATH

01 Read the ISTQB CTFL syllabus to structure a conversation about strategy and test conditions.

02 Study delivery metrics in the DORA documentation, relating each signal to the team's context.

03 Consult the official GitHub documentation to record risks, owners, and decisions in a fictional project.

04 Read Embracing Risk from the Google Site Reliability Engineering book and discuss how an agreed target can support release choices.

DORA presents five delivery performance metrics, grouped into throughput and instability [R09].

Use this path to explore the problem and adapt your learning to your context. The title does not determine a mandatory career sequence.

PART III

# QUALITY ENGINEERING AND AI


03 AI to support testing · Testing systems that use AI 18

04 Risk-oriented quality 20

05 AIMA 2.0: from information to decision 22


> **A PROPOSED TEST IS NOT EVIDENCE THAT IT PASSED.**
>
> CHAPTER 3

CHAPTER 03

# AI TO SUPPORT TESTING

**A suggestion generated to test a rule can contain an incorrect expected result. When AI itself is part of the product, the error can reach the user directly.**

The distinction changes the evidence needed. In the first case, the suggestion must be checked before use; in the second, the product's behavior must be evaluated, including what happens when the response is insufficient.

A team can use artificial intelligence to organize test ideas without the product under test containing AI. The object of evaluation remains the system under development; the generated output is a suggestion subject to review.

Version 2.0 of the ISTQB CT-AI syllabus focuses on testing AI-based systems and removed content about using AI for testing [R05]. This separation helps formulate two different questions: how does the tool support the work, and how do we evaluate a product that uses AI?

> HYPOTHETICAL EXAMPLE
>
> A person asks for test scenario suggestions for a registration rule and receives a list of valid and invalid inputs. Before running any test, they compare the list with the approved rules and identify a missing condition.

Review the expected result, data, and relevance of each suggestion. Do not send sensitive data without checking the rules applicable to use of the tool; OWASP includes sensitive information disclosure among the risks of applications using language models [R03].

A proposed test is not evidence that it passed. Keep the execution and its context separate from the suggestion that prompted it.

CHAPTER 03 · CONTINUED

# TESTING SYSTEMS THAT USE AI

**When AI is part of the product, evaluate the response and the surrounding system. A large language model (LLM) can produce an incorrect claim that appears convincing; a grounded response needs to be checked against accessible, relevant sources.**

The CT-AI syllabus mentions data representativeness and bias, simulated attack testing, changes in behavior over time, and difficulty establishing the correct result [R05]. OWASP identifies misinformation, prompt injection, sensitive information disclosure, and excessive agency among the risks of LLM applications [R03].

An agent may be allowed to choose and perform actions. Evaluate which tools it can invoke, which data it can access, where a person must approve an action, and how to stop or reverse the process.

Results can vary between runs. Define an evaluation set, correctness criteria, and metrics suited to the task: for example, the proportion of responses supported by the supplied sources, observed authorization failures, and cases referred for human review. Record denominators, conditions, and limits; no single metric guarantees safety.

HYPOTHETICAL EXAMPLE

An assistant answers questions about an internal policy using supplied documents. The team checks whether each answer is grounded in the available material and whether the system recognizes when the material is insufficient.

Use repeatable evaluations and human review to investigate relevant risks. Do not treat one correct response in a single case as proof of safe behavior in every case.

AI FOR TESTING × TESTING AI — FIG. 2

| | AI FOR TESTING | TESTING AI |
|---|---|---|
| OBJECT | the software you already test | the system that uses AI |
| QUESTION | does the AI output help the test? | is the system's behavior acceptable? |
| TYPICAL RISK | wrong suggestion accepted without review | hallucination, prompt injection, bias, excessive agency |
| ORACLE | known expected result | hard to define; evaluation by samples and criteria |
| EVIDENCE | human review of the output and test execution | evaluation sets, red teaming, monitoring |
| REFERENCE | tool documentation | ISTQB CT-AI v2.0 [R05]<br>OWASP LLM Top 10 2025 [R03] |

Human oversight runs through both columns: AI scales execution, it does not sign off the decision.

CHAPTER 04

# RISK-ORIENTED QUALITY

**A team may have evidence that a service works and still not know how it responds to a failure. The release decision must make that gap visible to those who bear its consequences.**

A documented decision lets the team return to the reason for the choice when new signals emerge. It can then review the priority, seek additional evidence, or change the release conditions with the responsible people.

When verification time is limited, choosing what to investigate is also a quality decision. My view of risk considers an undesirable situation, its possible consequence, and the context in which it may occur.

> HYPOTHETICAL EXAMPLE
>
> A release has a green pipeline, many tests, and high coverage. Even so, a user can see another person's order by changing the identifier sent to the API.

OWASP describes object-level authorization as a necessary check in functions that access records from client inputs [R02]. Record who can access each resource, test allowed and denied access, and examine the response without exposing real data.

An API contract can specify inputs, outputs, and expected error conditions. Reviewing it helps formulate integration tests, but a contract that omits the authorization rule does not demonstrate that it was followed.

For each risk, record the consequence, hypothesis, check, result, and gap. Then present what is known and what remains unknown to the person responsible for the decision.

CHAPTER 04 · CONTINUED

# MAKING A RELEASE DECISION WITH EXPLICIT LIMITS

HYPOTHETICAL EXAMPLE

A team intends to release a change that passed functional tests but could not verify its recovery after a dependent service fails. The responsible person must decide between delaying the release and releasing it with an explicitly accepted operational limitation.

Make the choice reviewable: describe the possible impact, available evidence, recovery gap, and who has authority to accept the risk. Also define how to detect a problem and who can stop or reverse the change.

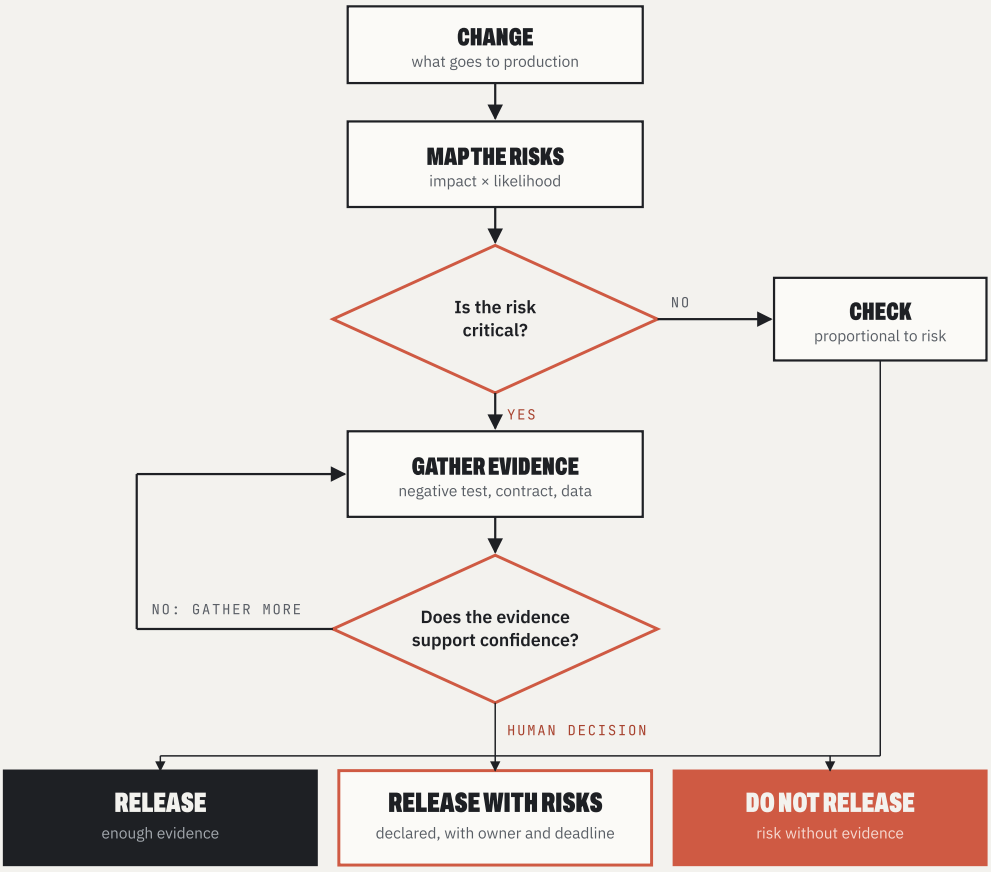
A service level objective (SLO) states a target for a service. In the risk chapter of Google's SRE book, the error budget connects the objective to the observed service and supports decisions to continue or slow releases [R08].

This instrument helps when there is an agreed objective and relevant measurement. It does not replace a specific decision about privacy, authorization, or another risk that the chosen measure does not represent.

RECOMMENDED FLOW

context → possible consequence → evidence → gaps → options → responsible decision → follow-up. Record the conditions for revisiting the decision; do not present an absence of data as evidence of an absence of failure.

RISK-BASED DECISION FLOW FIG. 3




The decision is human and recorded: who decided, on what evidence, and which risk remained open.

CHAPTER 05

# AIMA 2.0: FROM INFORMATION TO DECISION

**Accumulated test results can leave a team without a clear decision about the remaining risk. The method proposes organizing the conversation at this transition between information and choice.**

Running tests can produce results without clarifying what decision they support. AIMA 2.0 presents itself as a Quality Engineering method that transforms context, risks, and evidence into explicit, traceable, and reliable strategies; AI expands critical thinking without replacing it [R01].

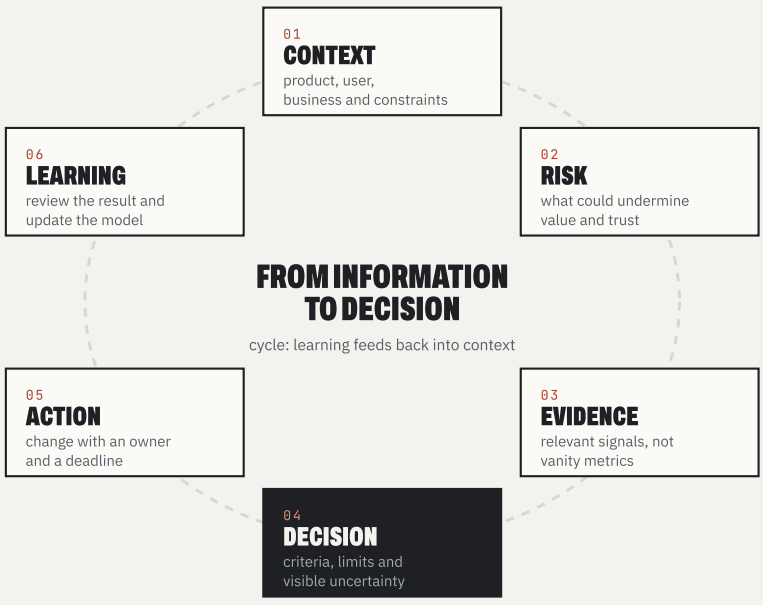
The site organizes the journey into six stations, in this order: **Context, Risk, Evidence, Decision, Action, and Learning** [R01]. In my view, the sequence invites the team to make the situation explicit, examine what might matter, gather signals, and make the choice reviewable.

The method is in Preview Edition and undergoing public validation, so it may change. Not every framework is executable. Quality Confidence is experimental: it is not a scientific probability and does not approve a release; recommendations do not replace human approval [R01].

According to the site, the Analyzer and Assessment run in the browser and do not send data. The Assessment is neither a certification nor a benchmark between teams [R01].

Use the method to structure a conversation about context, evidence, and decision, respecting its stated limits. To learn about the proposal and its documentation, visit aima20.dev.

AIMA 2.0 · THE METHOD'S 6 STATIONS FIG. 4




Source: aima20.dev, Preview Edition 0.9 (accessed Nov 17, 2026) [R01]. Recommendations always go to human review.

PART IV

# PROFESSIONAL DEVELOPMENT

CHAPTER 06

## SKILLS FOR DECIDING AND BUILDING

**A team can master automation and still struggle to explain what risk its suite covers. A matrix helps choose what to develop without reducing a person to a job title.**

Treat the matrix as a study map, not a universal job description. Choose the skills relevant to your context and look for observable evidence of application.

| AREA | SKILL | GUIDING QUESTION |
|---|---|---|
| Strategy | Risk analysis | What consequence deserves investigation? |
| Strategy | Test strategy | What check answers the hypothesis? |
| Strategy | Decision and evidence | What do we know and what is missing? |
| Strategy | Product knowledge | Who is affected by the change? |
| Technical | Programming | How can the check be made repeatable? |
| Technical | Test automation | What maintenance does the suite require? |
| Technical | APIs and contracts | What behavior was agreed upon? |
| Technical | Data | Does the data represent relevant conditions? |
| Systems | Security | Who can access each resource? |
| Systems | Reliability | How can a failure be detected and recovered from? |
| Systems | Continuous integration and delivery (CI/CD) | What evidence accompanies the change? |
| Systems | AI and evaluation | How can output and action be checked? |
| People | Communication | Is the limitation understandable? |
| People | Collaboration | Who needs to take part in the decision? |

CHAPTER 06 · CONTINUED

# SKILLS MATRIX BY TRACK

SKILLS MATRIX BY TRACK FIG. 5

| | QE | SDET | AI QE | APPSEC | Q&R | LEAD |
|---|---|---|---|---|---|---|
| **STRATEGY** | | | | | | |
| Risk analysis | ■■■ | ■■□ | ■■■ | ■■■ | ■■■ | ■■■ |
| Test strategy | ■■■ | ■■□ | ■■□ | ■■□ | ■■□ | ■■■ |
| Decision and evidence | ■■■ | ■□□ | ■■■ | ■■□ | ■■■ | ■■■ |
| Product knowledge | ■■■ | ■□□ | ■■□ | ■□□ | ■■□ | ■■■ |
| **TECHNICAL** | | | | | | |
| Programming | ■■□ | ■■■ | ■■□ | ■■□ | ■■□ | ■□□ |
| Test automation | ■■□ | ■■■ | ■■□ | ■□□ | ■■□ | ■□□ |
| APIs and contracts | ■■□ | ■■■ | ■■□ | ■■■ | ■■□ | ■□□ |
| Data | ■■□ | ■■□ | ■■■ | ■□□ | ■■□ | ■□□ |
| **SYSTEMS** | | | | | | |
| Security | ■□□ | ■□□ | ■■□ | ■■■ | ■□□ | ■□□ |
| Reliability | ■□□ | ■■□ | ■□□ | ■□□ | ■■■ | ■■□ |
| CI/CD | ■■□ | ■■■ | ■■□ | ■■□ | ■■■ | ■■□ |
| AI and evaluation | ■■□ | ■■□ | ■■■ | ■□□ | ■□□ | ■■□ |
| **PEOPLE** | | | | | | |
| Communication | ■■■ | ■□□ | ■■□ | ■■□ | ■■□ | ■■■ |
| Collaboration | ■■□ | ■■□ | ■■□ | ■■□ | ■■□ | ■■■ |

■ weight in the track (1 to 3)

The author's reading to guide study; not market research nor an official job description.

DORA presents five delivery performance metrics, grouped into throughput and instability [R09]. They can help a conversation about delivery, provided their context and limits are made explicit; they are not an individual skills assessment.

Use the matrix to choose a focus and define a practice. Do not require the same level in every skill from every person.

CHAPTER 07

# PROFESSIONAL SELF-ASSESSMENT

**A person may recognize security terms and still need help checking an access rule. The difference appears in what they can demonstrate, not just in their job title.**

The record also helps separate familiarity with a tool from the autonomy to solve a problem. Note the support received and what you would do differently in the next practice.

This exercise is a self-assessment to guide learning; it is not a validated psychometric instrument. Do not use your score to compare people, decide compensation, or infer professional performance.

Choose a skill from the matrix and describe a situation in which you applied it. Consider what you can demonstrate through work, study, or an exercise, and select the highest level for which you find sufficient evidence.

| LEVEL | CRITERION |
|---|---|
| 0 | No knowledge. |
| 1 | Introductory: recognizes terms and explains the purpose with supporting material. |
| 2 | Performs with guidance: completes a defined task with help. |
| 3 | Autonomy in usual situations: performs and explains decisions in a familiar context. |
| 4 | Solves complex problems: adapts the approach under constraints and uncertainty. |
| 5 | Guides others and defines strategy: explains criteria, supports decisions, and revises the approach. |

Record the evidence and its limit: what the context was, what you did, and what you have not yet demonstrated. If there is not enough evidence to distinguish two levels, record the uncertainty and plan an activity to investigate it.

CHAPTER 07 · CONTINUED

# FROM GAP TO ACTION

**Use the criteria below to observe six skills. For each, apply the previous page's scale to the behavior described, without assuming that a single artifact proves complete mastery.**

| SKILL | OBSERVABLE EVIDENCE |
|---|---|
| Risk analysis | Describes the consequence, affected people, and uncertainties of a change. |
| Test strategy | Connects a failure hypothesis to checks and justifies priorities. |
| Automation | Implements a check, explains its result, and identifies necessary maintenance. |
| APIs and contracts | Describes an input condition, the expected response, and a possible error. |
| Security | Identifies an access rule and checks allowed and denied access. |
| Communication | Presents a result, the evidence's limit, and a pending decision to another person. |

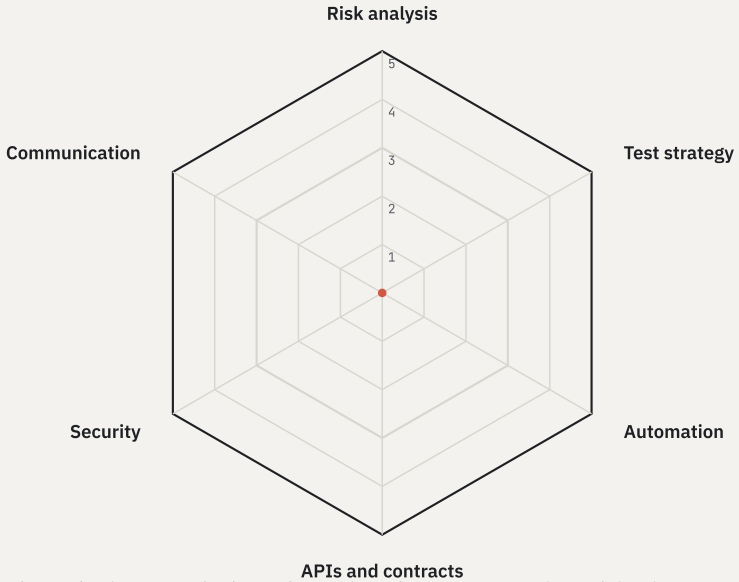
HYPOTHETICAL EXAMPLE

A person identifies that they can describe an access rule but still need guidance to check denied access. They choose a practice API, document the rule, and request a review of the tests they wrote.

To turn a gap into action, choose a skill, define a behavior you want to demonstrate, and create a small outcome. Choose a resource, a review date, and a person or criterion to evaluate the outcome; then record what you learned and your next question.

Use the number as a starting point for conversation. Do not treat the difference between levels as a precise measure of ability.

SELF-ASSESSMENT RADAR · FILL IN FROM 0 TO 5 FIG. 6




Mark your level on each axis, connect the dots and repeat in 90 days. 3 = autonomy in usual situations.

A reflection tool, not a validated psychometric instrument.

CHAPTER 08

# A 90-DAY PLAN

**A long course list can occupy months without producing evidence of applicable learning. A plan needs to connect a specific gap to a practice another person can examine.**

When choosing a project, prefer a small change with a verifiable result and reproducible conditions. The plan should fit your available routine and allow review before adding new activities.

Choose a path and an assessment gap before starting. Adapt the activities to the time available and keep the project small enough to explain and review.

| DAYS 1-30<br>UNDERSTAND AND DEFINE THE SCOPE | DAYS 31-60<br>VERIFY AND RECORD | DAYS 61-90<br>COMMUNICATE AND REVIEW |
|---|---|---|
| GOAL | | |
| recognize the problem and formulate a risk hypothesis. | produce evidence for the chosen hypothesis. | turn evidence into an explainable decision and a portfolio artifact. |
| ACTIVITIES | | |
| read the chosen reference, describe the context of a practice system, and identify affected people, expected behavior, and uncertainties. Choose a scenario that can be checked without real data. | implement or use a practice API, prepare fictional data, and create checks for allowed and denied access. Record execution conditions, results, and limitations. | review the checks, document a gap, request review when someone is available, and describe decision options. Remove private data and secrets before sharing any material. |
| RESOURCES | | |
| OWASP API Security Top 10 [R02] and official documentation for the chosen tool. | OWASP API Security Top 10 [R02] and official Postman or Playwright documentation. | OWASP API Security Top 10 [R02] and official documentation for the platform used to host the project. |
| DELIVERABLES | | |
| problem description, access rule, and prioritized list of questions. | reproducible checks and a record of results. | project documentation, evidence record, limitations, and next steps. |
| HANDS-ON PROJECT | | |
| design a practice API with resources associated with different people and specify who can access them. | exercise the API rule with distinct identities and resources. | finish the practice API and present the access rule, tests, and reasoning behind the choices. |
| COMPLETION CRITERION | | |
| another person can read the rule, identify the consequence of a violation, and understand what has not yet been tested. | the tests can be reproduced and their checked behavior and out-of-scope behavior explained. | a reader can reproduce the exercise, distinguish results from hypotheses, and understand what the project does not demonstrate. |

CHAPTER 08 · CONTINUED

# OVERVIEW AND CHECKLIST

90-DAY DEVELOPMENT ROADMAP FIG. 7

DAYS 1–30

## UNDERSTAND AND DEFINE SCOPE

— Read the chosen reference
— Describe context and uncertainty
— Pick a scenario with no real data

DELIVERABLE
problem, rule and questions

DAYS 31–60

## VERIFY AND RECORD

— Prepare fictional data
— Test allowed and denied access
— Record conditions and limits

DELIVERABLE
reproducible checks

DAYS 61–90

## COMMUNICATE AND REVIEW

— Document a gap
— Ask for a review
— Describe decision options

DELIVERABLE
project with evidence and limits

Completion criterion for each cycle: the deliverable exists and someone else can verify it.

PROFESSIONAL DEVELOPMENT CHECKLIST FIG. 8

PROGRESS CARD DATE: ____ / ____

- [ ] I chose a skill and recorded my starting point
- [ ] I described the problem and the people potentially affected
- [ ] I recorded the rule, hypotheses, and uncertainties
- [ ] I prepared fictional data and reproducible checks
- [ ] I saved results and execution conditions
- [ ] I documented gaps and possible next steps
- [ ] I reviewed the material before sharing it
- [ ] I reassessed the skill based on observable evidence

Use the plan as an adaptable reference. If the project scope changes, renegotiate the deadline and preserve the completion criterion.

CLOSING

# WHAT REMAINS OURS

**A tool can return a convincing result while an important decision still has no owner. The team needs to know who interprets the evidence and who takes responsibility for the choice.**

A tool can produce suggestions, run checks, and organize results. Responsibility for interpreting these signals and deciding what to do with uncertainty remains a human matter.

Quality work gains clarity when we can connect a possible consequence to a question, evidence, and a decision. This includes recognizing when the evidence does not answer the question and when another person needs to take part in the choice.

Throughout this guide, the same access problem appeared as a quality dilemma, a security investigation, and a release decision. The perspective changed; the need to protect affected people and make the limits of verification explicit remained.

Return to the assessment after completing a practice and record what you can demonstrate. A useful portfolio presents context, choices, observed results, and gaps, without turning an exercise into experience that did not happen.

Use tools to broaden investigation when their results can be reviewed. Do not transfer to them responsibility for accepting a risk they do not understand.

SOURCES

# REFERENCES

Each [R..] mark in the text points to the source below. A reference supports the point cited, not necessarily the whole section. Format: ABNT NBR 6023:2018 (Brazilian standard).


R01 DAVILA, Jonas. **AIMA 2.0**: método de Engenharia de Qualidade. Preview Edition, release 0.9. [S. l.], 2026. Available at: https://aima20.dev/. Accessed: Nov 17, 2026.

R02 OWASP FOUNDATION. **API1:2023 – Broken Object Level Authorization**. In: OWASP API Security Top 10 – 2023. [S. l.]: OWASP, 2023. Available at: https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/. Accessed: Nov 17, 2026.

R03 OWASP FOUNDATION. **OWASP Top 10 for LLM Applications 2025**. [S. l.]: OWASP GenAI Security Project, 17 nov. 2024. Available at: https://genai.owasp.org/llm-top-10/. Accessed: Nov 17, 2026.

R05 INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD. **ISTQB releases Certified Tester AI Testing (CT-AI) syllabus version 2.0**. [S. l.]: ISTQB, 2026. Available at: https://istqb.org/istqb-releases-certified-tester-ai-testing-ct-ai-syllabus-version-2-0/. Accessed: Nov 17, 2026.

R07 INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 25010:2023**: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. 2. ed. Genebra: ISO, 2023. Available at: https://www.iso.org/standard/78176.html. Accessed: Nov 17, 2026.

R08 ALVIDREZ, Marc. Embracing Risk. In: BEYER, Betsy et al. (ed.). **Site Reliability Engineering**: how Google runs production systems. Sebastopol: O'Reilly Media, 2016. cap. 3. Available at: https://sre.google/sre-book/embracing-risk/. Accessed: Nov 17, 2026.

R09 DORA. **DORA's software delivery performance metrics**. [S. l.]: Google Cloud, updated Jan 5, 2026. Available at: https://dora.dev/guides/dora-metrics-four-keys. Accessed: Nov 17, 2026.



JONAS DAVILA DA SILVA

AUTHOR

# ABOUT THE AUTHOR

**Jonas Davila da Silva is a Software Quality Engineering professional and Senior QA/QE, with approximately 18 years of experience in IT and approximately 10 years in QA/QE.**

He has experience in test strategy and architecture, automation, APIs, performance, and continuous integration and delivery. His background includes banks, fintechs, SaaS, consultancies, and technology companies, including software quality for financial systems. His experience includes technical collaboration, mentoring, and process improvement.

He is the author of the AIMA 2.0 project and studies application security and DevSecOps from his QE experience. His education includes IT Management and ISTQB CTFL certification. He is pursuing postgraduate studies in Software Quality and Testing Engineering at PUC Minas.

## RESOURCES

— AIMA 2.0
— Jonas Davila's portfolio
— GitHub repositories
— Quality Insights newsletter on LinkedIn.

QUALITY INSIGHTS

**QUALITY IS REDUCING RISK WITH EVIDENCE.**

QUALITY.MD 1 CHANGE

\- ~~QUALITY = NUMBER OF TESTS~~

\+ **QUALITY = REDUCING RISK WITH EVIDENCE.**

REVIEW: QUALITY ENGINEERING NEXT STEP: YOUR EVIDENCE

Artificial intelligence expands execution capacity; quality engineering guides decisions; evidence helps justify confidence.

Six career tracks, a chapter on AI for testing and testing AI, risk-based decisions, the AIMA 2.0 method, a skills matrix, a self-assessment and a 90-day plan.

AIMA20.DEV JONASDAVILA.COM.BR GITHUB.COM/JONASQASOFTWARE

CC BY-NC-ND 4.0 · SHARE WITH CREDIT, NO COMMERCIAL USE, NO MODIFICATIONS