
expoQA®26
MADRID 26th, 27th & 28th May
expoqa.eu

expoQA 2026 · Madrid · May 27, 2026 · AI Testing Track

Classical QE + Agentic Principles:
Building Bridges, Not Burning Them

Dragan Spiridonov

Founder & Agentic Quality Engineer · Quantum Quality Engineering · Agentics Foundation Ambassador for Serbia

ABOUT THE SPEAKER

Dragan “Profa” Spiridonov

Founder & Agentic Quality Engineer,
Quantum Quality Engineering, Petrovaradin, Serbia

- 30 years in IT (1996 → today)
- 12 years QA/QE leadership
- Founded Quantum QE in October 2025
- 3 open-source agentic testing platforms
- Board Secretary, Agentics Foundation
- Ambassador, Serbian Agentics Foundation Chapter
- The Quality Forge blog — forge-quality.dev

AGENTICS [FOUNDATION]

Quantum QE

The noise you've been hearing:

"AI will replace QA!"

https://www.functionize.com/blog/the-death-of-traditional-qa

"TDD is dead!"

https://neonwatty.com/posts/tdd-is-dead/

"Autonomous agents make exploratory testing irrelevant!"

https://www.thunders.ai/articles/autonomous-software-testing-and-ai-agents-the-new-era-of-quality-assurance

The question this talk will try to answer:

Why do teams that abandon classical foundations consistently underperform teams that evolve from them?

— And what do you do about it on the next working day.

THE UNIFYING FAILURE MODE

Completion Theater

Agents optimizing for appearing done rather than being done.

— Dragan Spiridonov, 2025 (independently validated: Chevrot et al., Type III Errors in Autonomous Testing)

Goodhart's Law with an API

Measure a proxy for quality.

Agents optimize the proxy.

"Task complete" becomes the goal.

Quality evidence disappears.

Classical testing already knows this

You don't ask the system under test whether it passed.

You design an independent check.

This is the oracle problem.

It doesn't get a new name for AI.

FAILURE MODE 1

Tests that confirm instead of challenge

What we tried

Let AI write both tests AND implementation simultaneously.

Faster? Yes.

Better? No.

Result: a test suite that confirmed the implementation rather than challenged it. Grammatically correct. Contextually meaningless. Confidently wrong.

What actually works

Use AI to generate edge cases you didn't think of.

But still write the failing test first.

The red-green-refactor discipline is not a ceremony. It is what forces you to specify correct behaviour before implementation exists.

AI amplifies coverage. TDD maintains design quality.

FAILURE MODE 2

Autonomous exploration without strategy

"Test checkout under poor network" → agent generated 47 network condition variations

Good. Charter-driven.

Pure autonomous exploration

Hundreds of issues found. Mostly noise.

Without human strategic intent, agents explore randomly. Signal drowns in volume. The team spends more time triaging false positives than fixing real problems.

Charter-driven agent exploration

Human testers write the charter.

Agents execute the variations.

The intelligence is in the charter, not the agent. Exploratory testing's strategic discipline scales — at machine speed.

Failure Mode 3

Green is not evidence.

Green is a hypothesis.

Treating a passing CI pipeline as a result is the oldest mistake in testing.
AI doesn't fix it. AI amplifies it.

The mistake

Ship based on green pipeline. No independent oracle. No adversarial check. Agent certified itself.

What breaks

Integration failures that passing tests don't reach. Agents that tested isolated units and missed the gaps between them.

Classical fix

Independent verification. Oracle separation. A check that runs whether anyone is watching — and cannot be influenced by the thing it verifies.

What Works

— and why it scales

Agent suggests. You specify tests. Agent implements.

1

Agent generates edge cases

Ask it for boundary conditions, unusual inputs, error scenarios you didn't think of.

2

You write the failing test

RED. You specify what correct behaviour looks like. The agent doesn't own this step.

3

Agent implements minimally

GREEN. The smallest code that makes the test pass. Nothing more.

4

Agent refactors with guidance

REFACTOR. You guide the agent to refactor, and your test suite catches regressions.

Human intuition guides strategy. Agents handle scale.

The Charter (you write this)

"Test checkout flow under degraded network conditions across mobile and desktop devices, focusing on error handling, recovery behaviour, and state consistency after reconnection."

The Agent (executes this)

- 47 network condition variations
- 3G / 2G / packet loss / timeout scenarios
- Mid-checkout disconnection states
- Recovery after reconnect
- Cart state persistence testing
- Mobile vs desktop divergence

Your context-driven oracles become agent evaluation criteria.

What makes a good test in your context? Write it down. Then program agents to follow those principles.

Context-driven principle	Encoded as agent criterion
RESTful consistency oracle: Same request → same response structure	Agent auto-checks: response schema matches across all generated test variations
Error message quality: Errors must be actionable, not cryptic	Agent evaluates: does the error message tell the user what to do next?
State consistency: Data written must be readable	Agent pairs every write operation test with a read verification assertion

Every agent output is a claim. Design verification that challenges it.

Separate verifier from verified

The agent that does the work cannot be the agent that verifies the work. Different context, different prompt chain, never the same instance.

Evidence triggers review, not scores

Human review is triggered by the absence of traceable reasoning — not by an arbitrary confidence percentage. If you cannot show why the output is correct, it goes to a human.

Anti-sycophancy by design

Build checks that reject hollow outputs. expect(true).toBe(true) is green and meaningless. Your verification layer should know the difference.

The Judgment Drain

AI handles decisions that humans would otherwise make through effortful judgment.

The hidden cost

Teams that outsource every testing decision to agents lose the ability to evaluate what the agents are doing.

The judgment muscle atrophies.

When something goes wrong at a scale agents didn't anticipate, nobody knows how to investigate it.

The protection

Keep making consequential testing decisions yourself.

Use agents for breadth. Use your judgment for the critical path.

Run regular exploratory sessions without AI assistance. Not because AI is bad — because judgment degrades without exercise.

When a task gets cheaper, we do more of it — not less.

Radiology:

Geoffrey Hinton predicted in 2016 that people should stop training radiologists. Actual data: 30,723 US radiologists in 2014 → 36,024 by 2023, projected 47,119 by 2055. AI lowered the cost of a scan — so clinics ran more scans. Total demand increased.

For Quality Engineering:

AI makes test generation cheaper	→ **teams test more, not the same team does less**
AI makes regression breadth affordable	→ **more regression coverage gets demanded**
AI makes API testing faster	→ **API test suites grow, not shrink**
Strategic testing judgment becomes scarcer	→ **more valuable, not less**

PACTS

The framework for evolution, not revolution

Five dimensions. Each maps a classical practice to its agentic extension.

P	**Proactive**	Classical: Risk-based testing	Agents surface issues before they become failures — continuously, not just at test time
A	**Autonomous**	Classical: TDD specification discipline	Agents operate independently within boundaries YOU define through test specifications
C	**Collaborative**	Classical: Exploratory testing charters	Agents execute charter variations at scale; humans maintain strategic oversight
T	**Targeted**	Classical: Context-driven oracle design	Precision over coverage theater; your heuristics guide agent evaluation criteria
S	**Structured**	Classical: ⚠ The missing dimension	Governance: agent behaviour is observable, auditable, and correctable. Most implementations skip this.

Your next-working-day roadmap

1

Audit your classical foundation

Can your team write good tests without AI? If no — fix that first. AI amplifies existing practice, good or broken.

2

Start agent-assisted, not autonomous

Pair your judgment with AI generation. You design the strategy; agents execute the variations.

3

Encode your testing heuristics

Write down what makes a good test in your context. Make those principles the agent's evaluation criteria.

4

Build confidence through evidence

Every agent output is a claim. Design verification that challenges it independently.

5

Scale where it works — preserve what doesn't need scaling

Agents handle regression breadth. Humans still lead critical-path exploratory sessions.

MONDAY MORNING — ONE TAKEAWAY PER PRACTICE YOU ALREADY OWN

You don't need to change everything. Start with one.

If you do TDD

Use AI to generate edge cases. Hold the red-green-refactor loop. You write the failing test.

If you do Exploratory Testing

Write the charter yourself. Give it to an agent. Let it multiply your session by x.

If you do Context-Driven Testing

Your oracles are now your agent evaluation criteria. Encode them before you run anything.

If you do Risk-Based Testing

Your risk model is your agent targeting strategy. High-risk areas get adversarial gates. Low-risk areas get breadth coverage.

Your classical QE skills are not obsolete.

They are your competitive advantage.

The teams winning the agentic transition are not the ones with the most agents.

They are the ones with the clearest human judgment about what to ask agents to do.

- Completion theater is structural — design independent verification
- TDD discipline + AI edge cases > AI-generated tests alone
- Charter-driven exploration > autonomous exploration
- Classical wisdom encoded > agents without heuristic grounding
- PACTS: Proactive · Autonomous · Collaborative · Targeted · Structured

Thank you.

Questions?

Blog	forge-quality.dev
Framework	agentic-qe.dev
Open source	github.com/proffesor-for-testing/agentic-qe
Community	youtube.com/@AgenticFoundationSerbia

expoQA®26

MADRID 26th, 27th & 28th May

Thank you for attending

expoqa.eu