

NORDIC TESTING DAYS 2026 · WORKSHOP

# The 70% Problem

Reclaiming Testing's Intellectual Core With Agentic Quality Engineering

**Lalit Bhamare** · with a live guest spot from Dragan Spiridonov

10:30 – 15:30 · Tallinn, Estonia · Hands-on workshop



## AGENDA

# Run of Show

<b>10:30</b>	Welcome & Part 1 — The Diagnosis	<i>Lalit</i>
<b>11:15</b>	AQE intro— Dragan (live video call)	<i>20 min</i>
<b>11:35</b>	Part 2 — The Architecture: QCSD Swarms	<i>Lalit</i>
<b>12:30</b>	Lunch	<i>60 min</i>
<b>13:30</b>	Re-entry & environment check	<i>buffer</i>
<b>13:40</b>	Part 3 — The PACT Framework	<i>Lalit</i>
<b>14:10</b>	Part 4 — Hands-On Orchestration	<i>Lalit</i>
<b>15:05</b>	Personal Adoption Roadmap	<i>Lalit</i>
<b>15:20</b>	Wrap-Up & Q&A	<i>Lalit</i>



PART 1

# Welcome & The Diagnosis

10:30 – 11:15 · why the problem persists



## THE DIAGNOSIS

# The Faster Horse Problem

*Same trap. Different century.*

### 1900s — THEN

- Problem: need faster transport
- Solution offered: faster horses
- Missed: the automobile

### TODAY — AI ERA

- Problem: testing can't keep pace
- Solution offered: more test cases, more AI automation
- Missing: quality consciousness, value, business impact



# Record AI adoption. Same old metrics.

**56%**

**measured on test coverage**

*not on business impact*

**70%**

**of AI usage goes to test-case creation**

*the faster horse, automated*

**4.5%**

**measured on NPS / business value**

*the number that matters*

*The industry upgraded its tools but not its thinking — still rewarded for volume, not value.*



## THE DIAGNOSIS

# The 70% Problem

*Where does tester time actually go?*

# 70%

CLERICAL BURDEN

- Documentation & artifact management
- Repetitive analysis & reporting
- Automation maintenance
- Status updates & test reporting
- Insights from failing tests/defects

# 30%

EXPERT WORK

- Critical thinking
- Risk judgment
- Testability assessment
- Deep exploration
- Context analysis
- Strategic testing

*Agentic QCSD flips this ratio.*



INTERACTIVE · ~10 MIN

# Exercise: Map your own organisation

- Plot your team's testing activity on the clerical ↔ expert spectrum.
- Where does your 70/30 actually sit? Be honest.
- What would you stop doing tomorrow if no one was counting it?
- Pairs discuss, then a few share out.



AQE INTRO

# Dragan Spiridonov

11:15 – 11:35 · live video call



LIVE VIDEO CALL · DRAGAN

# Introduction to Agentic QE

- Live walkthrough of the framework through website along with story telling of how Dragan built it from scratch. - from idea to iterations – to final product.



PART 2

# The Architecture: QCSD Swarms

11:35 – 12:30 · why it's designed this way



# The tester, amplified — not replaced

## THE TESTER

- Critical thinking
- Risk judgment
- Testability
- Deep exploration
- Business context
- Ethical judgment
- Creative what-if
- Domain expertise

AMPLIFY ▶

### Ideation Swarm

Risk Storming · Testability Review

13 agents · 9 skills

### Refinement Swarm

Product Factors · Coverage Forecasting

8 agents · 6 skills

### Development Swarm

Test Design · Automation Scripting

21 agents · 7 sub  
· 11 skills

### CI/CD Swarm

Quality Gates · Regression Risk

11 agents · 6 skills

### Production Swarm

Prod Telemetry · ML Defect Prediction

20 agents · 11 skills

## PART 2 · ARCHITECTURE

# One coherent system

60+

QE Agents

90+

QE Skills

13

DDD Contexts

11

Platform Integrations

*Skills activate by swarm context, not keywords. Agents coordinate through shared memory under hierarchical orchestration.*

Numbers keep growing — latest at [agentic-qe.dev/agents](https://agentic-qe.dev/agents)



LIVE DEMO

# The QCSD lifecycle in motion

Epic → Ideation → Refinement → Development → Verification → Production → Learning

Watch the swarms surface risk before code — and the human gates at every step: agents inform, humans decide.



# Grounded in production reality

- A real enterprise case where the QCSD swarm approach was applied.
- What the swarms surfaced that conventional testing had missed.
- Where the system needed human judgement to interpret, not just act.
- The honest trade-offs — what worked, what didn't.



LUNCH

**Back at 13:30**

12:30 – 13:30 · the hands-on half is after lunch — keep your laptop ready.



# Environment check before the lab

- Confirm Node.js 20+, Git, and the framework install on every machine.
- Warm up the framework; clear technical snags now, not mid-lab.
- Pair up — at least one fully-working setup per pair as a safety net.
- Doubles as the technical / changeover buffer.



PART 3

# The PACT Framework

13:40 – 14:10 · evaluate any agentic system



# PACT: four interrogation tools

## Proactive

Anticipates risk — not just reacting to prompts.

## Autonomous

Genuine autonomy vs. sophisticated prompt-following.

## Collaborative

Real coordination vs. parallel execution that never connects.

## Targeted

Sharp and context-fit vs. merely broad.



# Apply PACT — don't take the agent's word

- Score the swarm outputs and case-study evidence from Part 2 against PACT.
- Turn it on outside AI-testing platforms you already use or evaluate.
- Form your own assessment — not the presenter's conclusion.
- Build a portable evaluative instinct you keep after today.



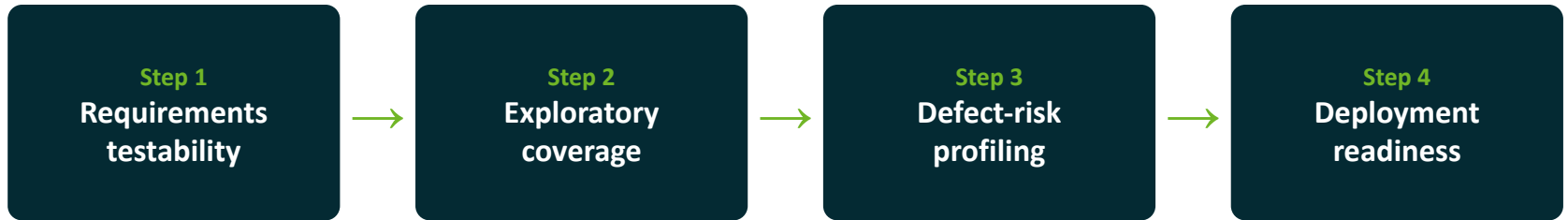
PART 4

# Hands-On Orchestration

14:10 – 15:05 · this is where it lives



# The lab arc — across the SDLC



*Configure and run multi-agent pipelines live on real code; apply PACT to interrogate the outputs.*



# Agents inform. Humans decide.

## HUMANS

- Set quality strategy & risk appetite
- Explore ambiguity, challenge assumptions
- Risk-based decisions at the gates
- Interpret outcomes, drive business impact

## AGENTS

- Testability scoring, product-factor analysis
- Test generation & coverage analysis
- Quality gates, flaky-test hunting
- Production intelligence, deployment readiness



# A starting point for your context

## 1 Install & Explore

- `npm install -g agentic-qe`
- `aqe init` in your project
- Browse the 90+ skills library

## 2 Map One Agent to One Pain

- Pick your biggest clerical burden
- Try `qe-qx-partner`, `qe-flaky-test-hunter`, `qe-a11y-ally`
- Measure time saved in week one

## 3 Build the Swarm Gradually

- Add agents as trust grows
- Month 1: agent suggests, human decides
- Month 4: agent handles low-risk autonomously

**“If we are to save costs and time, and still deliver quality software, then it is always cheaper to do things right the first time.”**

— Jerry Weinberg, Quality Software Management



## WRAP-UP

# What you take with you

- Expert thinking, accessible — without years of investment.
- A configured, MIT-licensed framework — nothing held back.
- PACT as a portable, vendor-neutral evaluation instinct.
- A personal adoption roadmap for your context.

## RESOURCES

[agentiqe.dev](https://agentiqe.dev)

[github.com/proffesor-for-testing/agentiqe](https://github.com/proffesor-for-testing/agentiqe)

[forge-quality.dev](https://forge-quality.dev)

[teatimewithtesters.com](https://teatimewithtesters.com)



THE FASTER HORSE WON'T SAVE US

**But an amplified tester might.**

