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

MAIN TRACK TALK

AI Meets API:
Transforming Automated Testing Processes

Madrid, May 2026

JOHAN PEARSON

SENIOR QA AUTOMATION ENGINEER

- 15+ years in test automation, architecture & consulting
- Backend & API testing, CI/CD-driven quality strategies
- Led automation in energy, retail & enterprise platforms
- Works closely with dev & product teams to improve reliability
- Drives innovation in scalable automation & emerging technologies

ADHA HRUSTO

AI ENGINEER

- PhD in Computer Science (Lund University, Sweden)
- Specializes in ML & GenAI for software quality
- Built agentic AI chatbots & AI-driven test frameworks
- Turned cutting-edge research into real-world solutions
- Published & presented at international conferences

SYSTEM VERIFICATION

IMPROVING DIGITAL LIFE WITH
INNOVATIVE QUALITY ASSURANCE

MOTIVATION

- Embrace new technology or fall behind
- Faster releases and more uncertainty with AI-driven development
- QA engineers must handle increasing testing demands without compromising quality
- Traditional approaches don’t scale anymore

THE PROBLEM

The automations engineer's reality...

- **80% boilerplate** — same patterns, different endpoints, over and over
- **Maintenance treadmill** — API changes break tests, unplanned rework
- **Coverage gaps** — edge cases missed until production incidents
- **Requirements disconnect** — specs say one thing, tests validate another
- **Doesn't scale** — more endpoints = more engineers, linear cost growth

What if we could...

- **Eliminate** the boilerplate entirely — focus on what matters
- **Stay in sync** — spec changes trigger automatic test updates
- **Bridge the gap** — requirements map directly to test coverage
- **Minutes, not weeks** — from spec to running tests

30+
endpoints per API

1 day
per endpoint (manual)

weeks
of engineering effort

minutes
with SpecOps

KEY DESIGN DECISIONS

Seamless adoption for **QA engineers**

No reliance on generated **code quality**

Clear **traceability** between generated test cases and API Specs

Structured **test models** designed for easy reading and analysis

QA engineers can **review generated test cases**

Easy maintenance and updates of the test suite as APIs evolve

THE SOLUTION

SpecOps

An AI-driven solution that generates executable NUnit tests from OpenAPI specs

Automated Generation

AI generates contract and system tests from your API spec

Self-Healing

Failures analyzed by AI, test models auto-corrected

Incremental Updates

Detect spec changes, regenerate only what's affected

Domain-Aware

Inject business rules and glossary into AI prompts

CONTRACT & SYSTEM TESTS

CON_ Contract Tests

Single-endpoint validation

- Happy path, error cases, and boundary conditions
- Auth and authorization scenarios
- Schema validation against OpenAPI spec
- Status code and response body matching

POST **/Customer/register**
→ 200 ✓ schema valid ✓ body matches

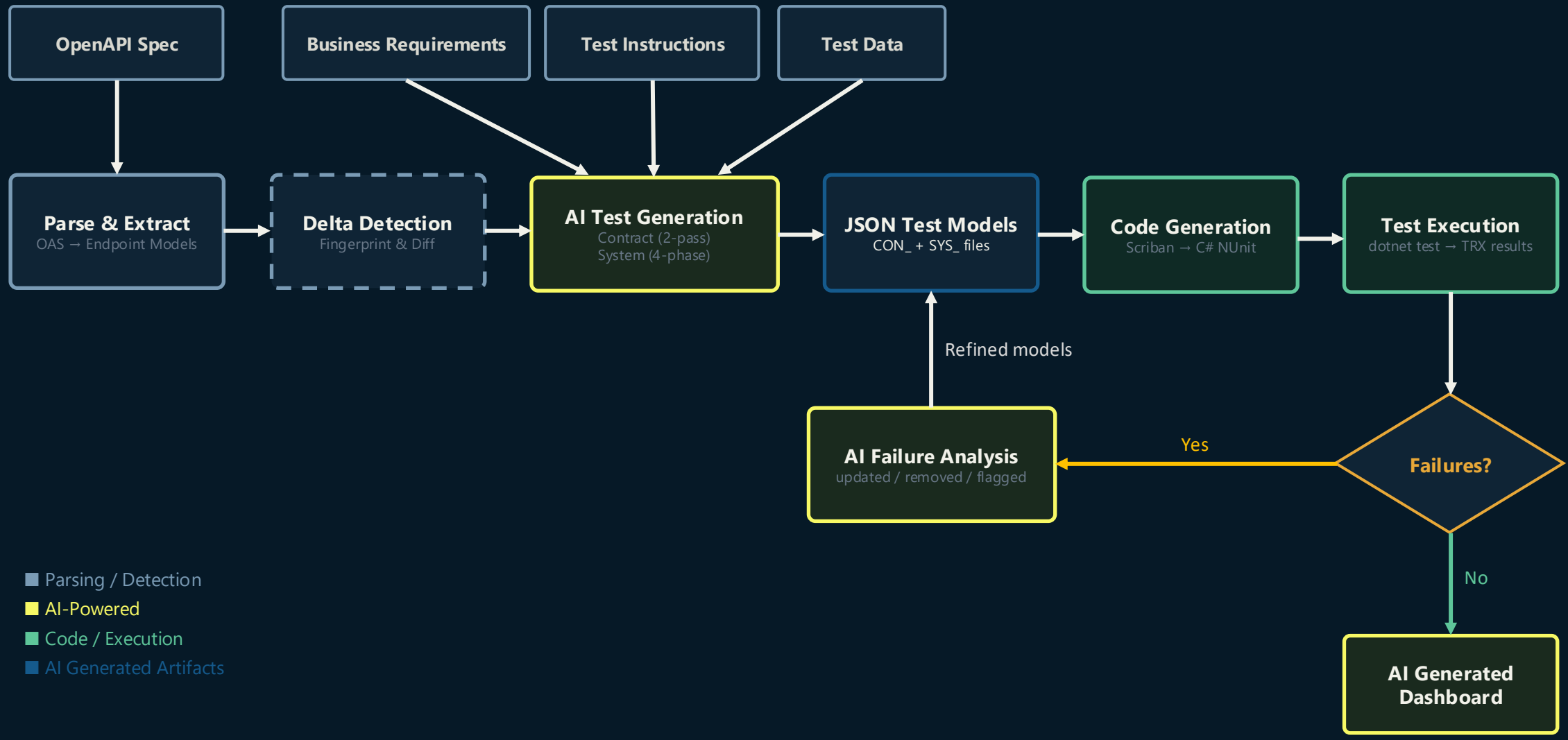
SYS_ System Tests

Multi-step E2E workflows with output chaining

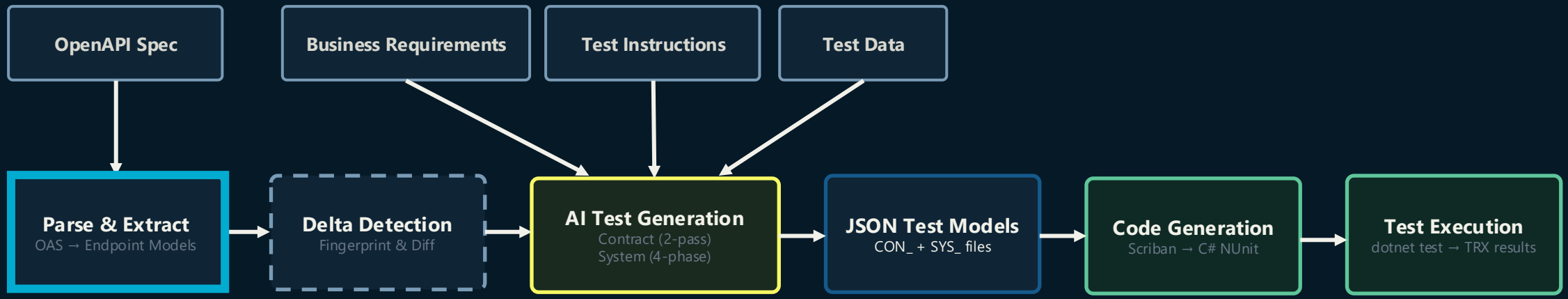
- Realistic business flows with data passing
- Response values captured and chained between steps
- AI-mapped endpoint relationships drive flow design

1: **POST /register** → {customerId}
2: **POST /login** → {authToken}
3: **POST /purchase** → {ticketId}
4: **GET /ticket/{ticketId}** → ✓

HOW IT WORKS


OpenAPI Spec
Business Requirements
Test Instructions
Test Data
Parse & Extract
OAS → Endpoint Models
Delta Detection
Fingerprint & Diff
AI Test Generation
Contract (2-pass)
System (4-phase)
JSON Test Models
CON_ + SYS_ files
Code Generation
Scriban → C# NUnit
Test Execution
dotnet test → TRX results
Refined models
AI Failure Analysis
updated / removed / flagged
Yes
Failures?
No
AI Generated Dashboard
Parsing / Detection
AI-Powered
Code / Execution
AI Generated Artifacts

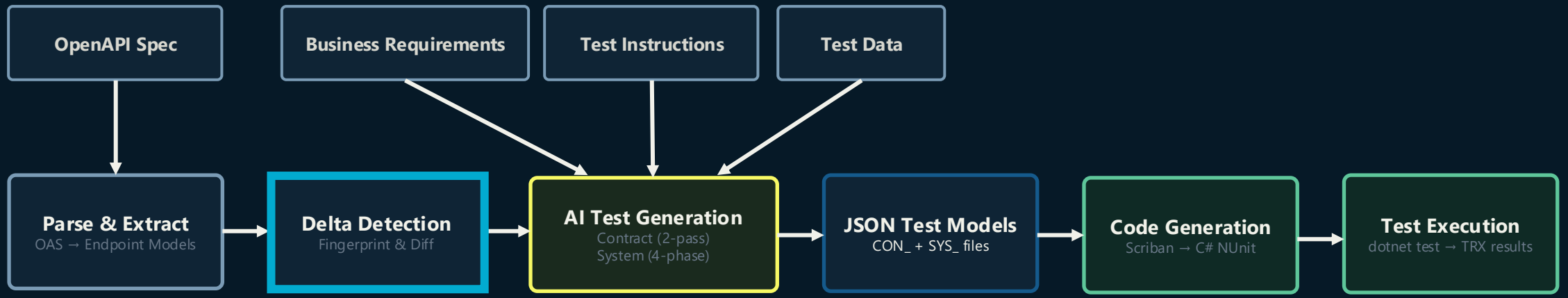
HOW IT WORKS



Parse & Extract

- Supports Swagger 2.0, OpenAPI 3.0.x, and 3.1.x via Microsoft.OpenApi
- Extracts parameters, request/response schemas, security requirements
- Computes SHA-256 fingerprints per endpoint for change detection
- Generates quality-report.json scoring documentation completeness

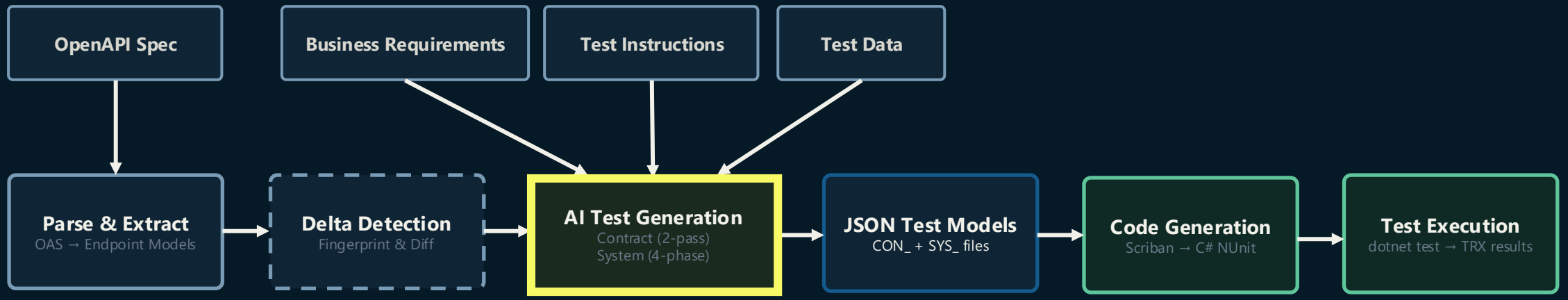
HOW IT WORKS



Delta Detection

- DeltaTracker compares current fingerprints against the manifest
- Classifies endpoints as added / modified / removed / unchanged
- For modified endpoints, SchemaDiffer produces field-level diffs
- In incremental mode, only changed endpoints proceed to AI generation

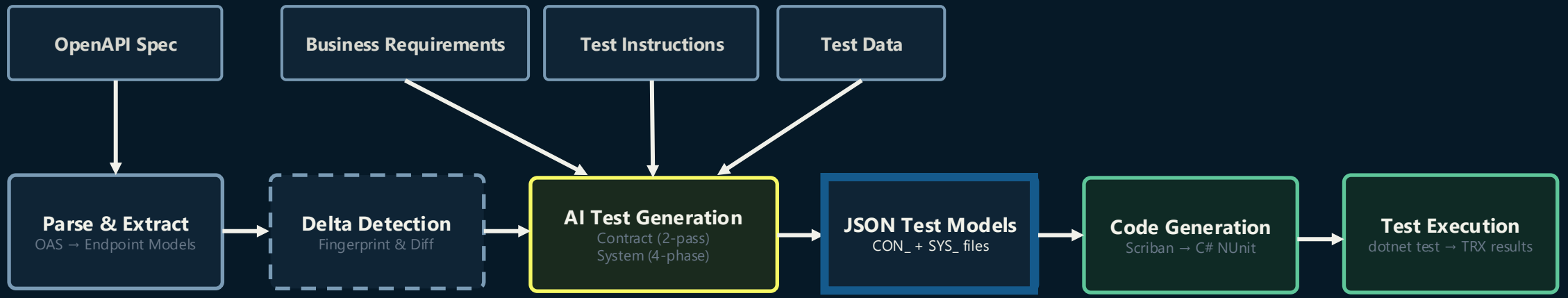
HOW IT WORKS



AI Test Generation

- Contract tests: Two-pass (generate + refine) per endpoint
- System tests: Four-phase (reduce → dependencies → skeletons → enrich)
- Prompt templates with placeholder injection for context-aware generation
- Self-correction: Up to 2 retries on invalid JSON output
- Produces dependencies.json mapping endpoint relationships

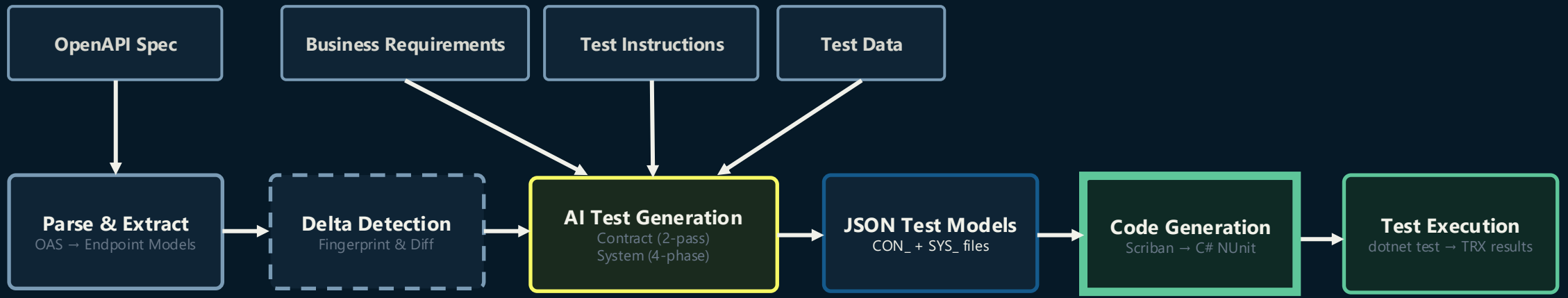
HOW IT WORKS



JSON Test Models

- Contract tests: CON_ prefix — single endpoint scenarios
- System tests: SYS_ prefix — multi-step with output chaining
- Required keys: testCaseName, testCaseID, description, steps
- <SKIPPED> markers for dynamic values that skip comparison

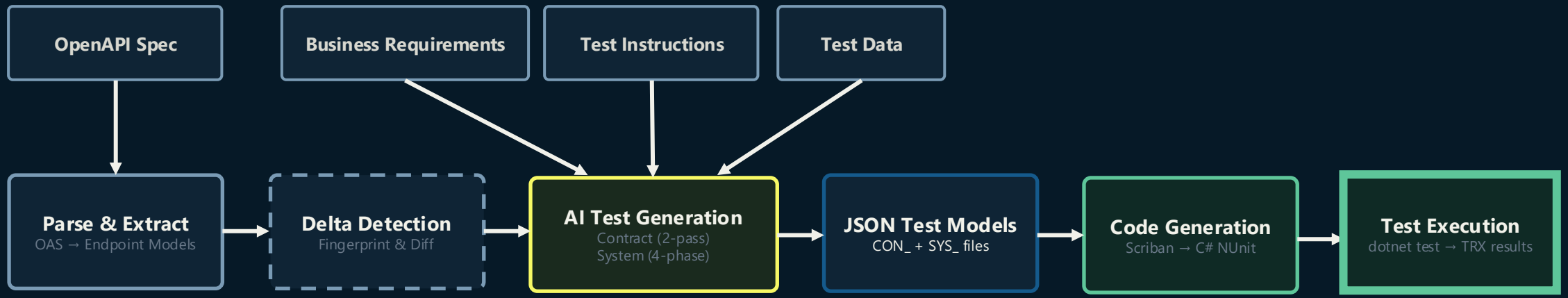
HOW IT WORKS



Code Generation

- Scriban template renders test classes with NUnit attributes
- Each step: configure request → send via HttpClientHelper → validate response
- Tests organized in src/test-gen/SpecOps.Tests/{ApiName}/
- Same JSON input always produces identical C# output

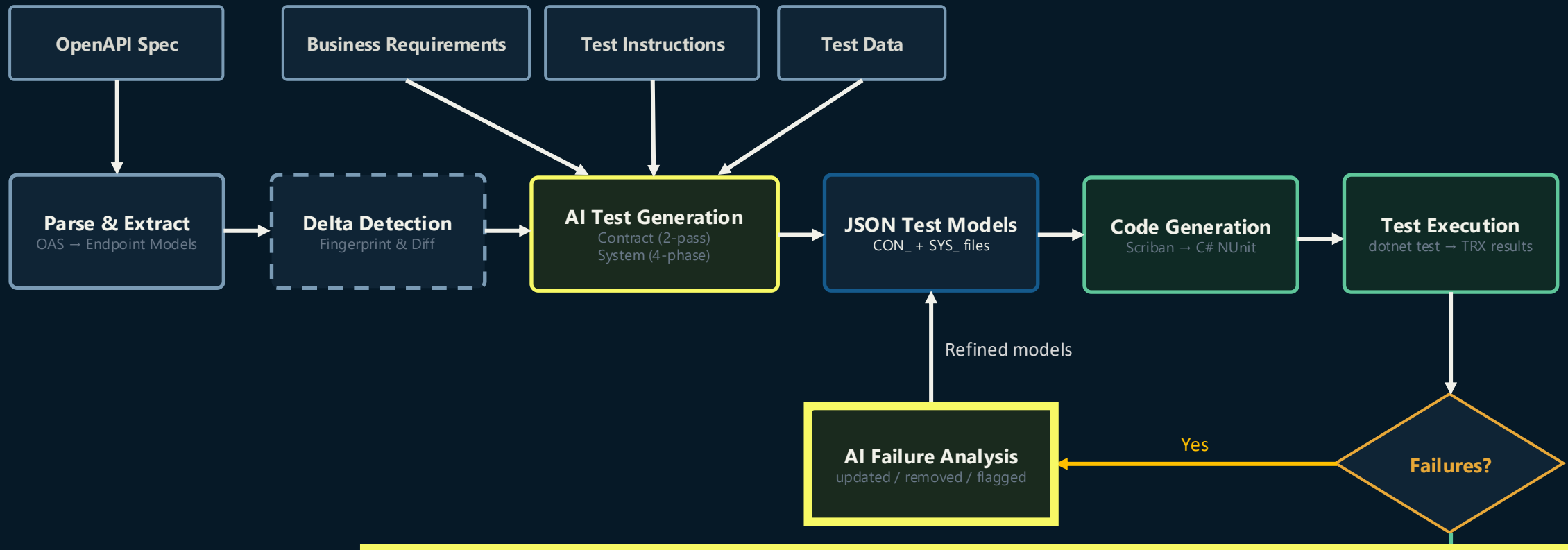
HOW IT WORKS



Test Execution

- Standard dotnet test producing TRX result files
- HTTP client with per-API instances and auth resolution
- Response validation: status codes, JSON schema, body values
- Output chaining captures values between multi-step tests

HOW IT WORKS



AI Failure Analysis & Refinement

- TrxParser groups failures by test model file
- AI classifies each failure: updated (corrected data), removed (invalid), flagged (needs review)
- --apply saves corrections and regenerates C# files
- Produces refinement-report.json

TRANSPARENCY & CONTROL

Every AI decision is documented, every pipeline step is visible, and you stay in the driver's seat

Every decision documented

Quality report — spec completeness scores, missing schemas flagged

Validation report — requirements mapped to API coverage

Refinement report — why each test was corrected, removed, or flagged

Dependency map — AI-detected endpoint relationships

Full pipeline visibility

Endpoint models — parsed OAS structures, readable JSON

Manifest — SHA-256 fingerprints for change tracking

Dashboard — aggregated metrics in a single HTML view

Diff output — exactly what changed since last run

You stay in control

JSON contract — review and edit AI output before code gen

Test instructions — guide AI with domain knowledge

Flagged items — AI escalates ambiguous cases for human review

Custom templates — Scriban templates you own and modify

INPUTS & CODE GENERATION

Pipeline Inputs

OpenAPI Spec

The source of truth — Swagger 2.0, OAS 3.0.x, 3.1.x

Business Requirements

Natural-language requirements for coverage validation

Test Instructions

Domain knowledge, glossary, excluded endpoints

Test Data

Realistic values for request bodies and parameters

JSON → C# (Deterministic)

```json
{
  "testCaseName": "Register Customer",
  "testCaseID": "CON_TC_001",
  "steps": [{
    "endpoint": "/Customer/register",
    "method": "POST",
    "expectedStatusCode": 200
  }]
}
```

▼ Scriban template ▼

```
[TestCase(TestName = "Register Customer")]
public async Task CON_TC_001()
{
    var setup = SetupStep(new SetupModel {
        Endpoint = "/Customer/register",
        Method = "POST",
        ExpectedStatusCode = 200 });

    var response = await SendRequestAsync(setup);
    await ValidateResponseAsync(response, setup);
}
```

SpecOps Demo

AI-powered framework that transforms OpenAPI specifications into comprehensive, executable NUnit test suites, automatically.

.NET | AI-Powered | NUnit | OpenAPI

Terminal

Full Pipeline | Analyze | Diff | Status | Validate | Incremental | Gen Tests | Refine | Dashboard

SpecOps CLI

```
$ specops analyze --api TicketingSystem
Parsing: ticketing-system-oas.json (OpenAPI 3.0.1)
Extracting endpoints...
  ▸ POST /Customer/register
  ▸ POST /Customer/login
  ▸ GET /Product
  ▸ POST /Purchase
  ▸ GET /Ticket/{ticketId}
  ... and 13 more
  18 endpoints extracted
  quality-report.json saved (18 endpoints · 2 info · complexity: Medium)

Computing fingerprints for 18 endpoints...
Comparing against previous manifest...
  18 new endpoints (first run)

$ specops generate models --api TicketingSystem --type all
Loading test instructions...
Generating contract tests (2-pass per endpoint)...
  [████████████████████] 18/18
  47 contract test cases generated
Generating system tests (4-phase pipeline)...
  Phase 1: Reducing OAS...
  Phase 2: Extracting dependencies...
  dependencies.json saved
  Phase 3: Generating skeletons...
  Phase 4: Enriching 4 test cases...
  12 system test cases (4 workflows)
```

Terminal

Full Pipeline | Analyze | Diff | Status | Validate | **Incremental** | Gen Tests | Refine | Dashboard

SpecOps CLI

```
$ specops analyze --api TicketingSystem
Parsing: ticketing-system-oas.json (OpenAPI 3.0.1)
  18 endpoints extracted

Comparing against manifest fingerprints...
  Added:     2 endpoints
  Modified:  1 endpoint (schema change)
  Removed:   0
  Unchanged: 15

$ specops generate models --api TicketingSystem --mode incremental
Skipping 15 unchanged endpoints...
Generating 2 new endpoints...
  POST /Refund - 3 test cases
  GET /Refund/{refundId} - 2 test cases
Merging 1 modified endpoint...
  Schema diff for GET /Product:
    + "discountPercentage" (number)
    ~ "price" - now nullable
  Merged into existing 4 test cases
  3 endpoints processed instead of 18

$ specops generate tests --api TicketingSystem
  3 test files updated, 24 unchanged

Test model changes:
  + NEW: CON_Post_Refund.json
  + NEW: CON_Get_Refund_refundId.json
  ~ MODIFIED: CON_Get_Product_productId.json
    + "discountPercentage" (number)
    ~ "price" — now nullable
  24 test models unchanged
```

SpecOps - Pipeline Dashboard

API: TicketingSystem OAS: openapi.json Last Run: 2026-03-26 14:58 Generated: 2026-03-26 15:30

19

ENDPOINTS

Unique API operations discovered in the OpenAPI specification.

100.0%

CONTRACT COVERAGE

Endpoints with at least one contract test validating request and response behavior.

190

TEST CASES

247 steps

Total generated tests across contract and system suites.

85.7%

REQ. COVERAGE

1 partial

Business requirements fully backed by API endpoints.

80.5%

TEST PASS RATE

153 / 190 passed

Latest live execution result for the generated test suite.

Test Coverage

Coverage measures how many API endpoints have generated tests. Contract tests validate individual endpoints, while system tests validate multi-step workflows that chain endpoints together.

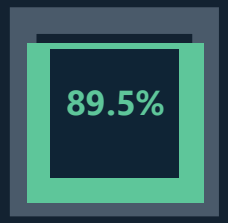
Contract Tests

Validates each endpoint independently against its OpenAPI schema.

19 / 19 endpoints

System Tests

Validates end-to-end workflows across multiple chained endpoints.



17 / 19 endpoints

Test Distribution

	Contract	System	Total
Test Cases	153	37	190
Test Steps	163	84	247

Uncovered Endpoints

These endpoints do not yet have generated tests. This may be intentional (e.g. utility or test-only endpoints) or may indicate gaps that need attention.

Contract

No uncovered endpoints

System

- [PUT] /Notifications/{id}
- [DELETE] /Notifications/{id}

API Specification Quality

Analysis of the OpenAPI specification completeness and best practices. These issues influence generated test quality and expose documentation gaps.

Issue Summary

2 warnings 14 info 19 endpoints

Warnings indicate schema or request contract problems. Informational issues highlight missing security or incomplete error response definitions.

API Complexity

Low

Metric	Value
Avg Params/Endpoint	0.5
Avg Request Fields	1.2
Max Schema Depth	1
File Uploads	No

Quality Issues

16 issues

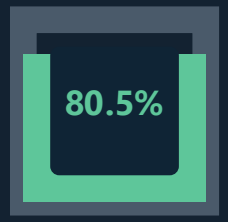
Sev	Code	Endpoint	Message
warning	empty-success-response	[POST] /Test/testdata/generate	No response schema for 201
warning	missing-request-body	[POST] /Test/testdata/generate	Write method without request body schema
info	missing-security	[POST] /Customer/register	No security definition
info	missing-security	[POST] /Customer/{customerId}/wallet/transfer	No security definition
info	missing-error-responses	[GET] /Customer	No error responses (4xx) defined
info	missing-security	[POST] /Customer/login	No security definition
info	missing-error-responses	[GET] /Notifications	No error responses (4xx) defined
...	...	...	...

Test Execution And Refinement

Latest live execution results plus the AI refinement pass that classifies failures as updated, flagged, or removed.

Pass Rate

Percentage of generated tests passing in the latest execution.

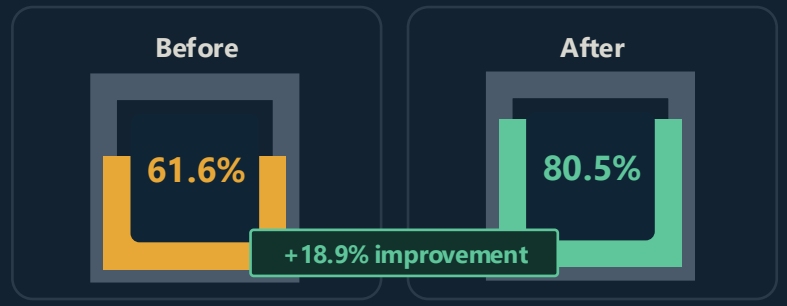


153 / 190 tests passed

Execution Details

Metric	Value
Passed	153
Failed	37
Total	190
TRX File	JohanP_PEARSON-SURFACE_2026-03-26_16_28_58.trx
Timestamp	2026-03-26 15:29

Refinement Result



Pre-refinement

Post-refinement

Refinement Actions

Analyzed 73 failures across 20 models (190 total tests)

44
Updated
Tests corrected

29
Flagged
Potential API issues

0
Removed
Not applicable

- Flagged Issues - Potential API Bugs — 29 flagged
- Updated Tests - AI Corrections — 44 updated
- Removed Tests - Dropped During Refinement — 0 removed

Requirements Validation

Cross-references business requirements against the API endpoints defined in the OpenAPI specification and highlights where coverage is full, partial, or missing.

Coverage Summary

6 Full | 1 Partial

Requirement Coverage Matrix

ID	Requirement	Coverage	Endpoints	Gaps
1	Customer Lifecycle Management	Full	POST /Customer/register POST /Customer/login GET /Customer GET /Customer/{customerId}	-
2	Wallet and Transaction Handling	Partial	POST /Customer/{customerId}/wallet/transfer	There is no explicit endpoint for viewing wallet balance or transaction history. Error responses for insufficient funds or invalid amounts are implied but not directly documented in the endpoints.
3	Product and Event Management	Full	POST /Product GET /Product GET /Product/{productId}	-
4	Purchase Flow	Full	POST /Purchase GET /Purchase/{purchaseId}	-
5	Ticket Management	Full	GET /Ticket/{ticketId} GET /Ticket/validate/{ticketId}	-
6	Notification System	Full	POST /Notifications GET /Notifications GET /Notifications/active PUT /Notifications/{id} DELETE /Notifications/{id}	-
7	Test and Data Management	Full	POST /Test/testdata/generate DELETE /Test/database/clear	-

KEY TAKEAWAYS

Scale and accelerate API testing to match development pace, while preserving quality and empowering QA engineers.

- AI ELIMINATES BOILERPLATE
- COMPREHENSIVE TEST COVERAGE
- INCREMENTAL UPDATES
- SELF-HEALING
- FULL TRANSPARENCY
- YOU STAY IN CONTROL

ASK US QUESTIONS AND REACH OUT!

Website

www.systemverification.com

Email & LinkedIn

Johan Pearson
johan.pearson@systemverification.com

Email & LinkedIn

Adha Hrusto
adha.hrusto@systemverification.com

expoQA®26

MADRID 26th, 27th & 28th May

Thank you for attending

expoqa.eu