

9ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



15 septembre 2026
17h à 23h



ETSI, Valbonne



Soirée du
test logiciel



Sophia Antipolis

From Metric Definitions to Automation: Building a Quality Data Model for AI



Andrea TIPA
Principal Quality Assurance Engineer
amADEUS



The journey to the standardization



why change?





Why Change? Key KPI Challenges

Lack of a Common Measurement Baseline

- *No standardized metrics to assess maturity*
- *Difficult to measure adherence to best practices*
 - *Limited ability to set meaningful targets*

Lack of a Common Language

- *Inconsistent metric and KPI definitions*
- *Different interpretations across teams*
- *Difficult benchmarking and comparison*

Limited Automation

- *Heavy reliance on manual calculations*
- *Time-consuming reporting activities*
- *Reduced efficiency and scalability*



The journey to the standardization



why change?



Definitions

Definitions

Metrics

Metrics are raw measurements.

- A **quantified measurement** directly observed or calculated
- **Value without interpretation by itself**
- At this stage, no objective is set

Example:
Invalid defects = 5



Indicators

Indicators are metrics with meaning (direction, intent).

- One or more **metrics interpreted in context**
- Tied to a **question or concern**
- Gives a **signal** about health, trend, risk, or maturity

Example:
Defect Rejection Ratio (DRR): evaluates the percentage of invalid defects over the total defect inflow



KPIs

KPI is an indicator elevated to steering & accountability.

- An **indicator**, not a raw metric
- Selected because it is **key**
- Has targets, thresholds, ownership
- Used to **decide, steer, or report**

Example:
 $DRR \leq 10\%$



Purpose

Measure

Understand

Steer

Typical question

"What is the value?"

"What does this tell us?"

"Are we on track vs objectives?"



Indicators types

Leading indicators (internal)

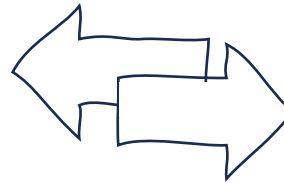
Leading indicators track processes, behaviours, and maturity (**what's being done**) helping predict or influence future outcomes reflected in lagging KPIs.

A subset of these becomes KPIs to steer internal performance and improve lagging KPIs

- Leading KPIs are **enablers** not outcomes
- Leading indicators → show **capability, behaviour, and maturity**

Example:

Defect Rejection Ratio
(Are we good at identifying real issues?)



Lagging indicators (outcome)

Lagging indicators measure results and final impact (**what happened**). They provide a retrospective view of performance and are typically used for **customer communication, executive reporting, and business accountability**.

Some serve as KPIs for customer communication and management reporting.

Lagging indicators show **results and impact**

Example:

Defect Leakage
(Did we miss real issues?)

Time horizon

Short / near-term

Medium to long-term

Change speed

Fast

Slow

Control

High (team-controlled)

Low (result of many factors)

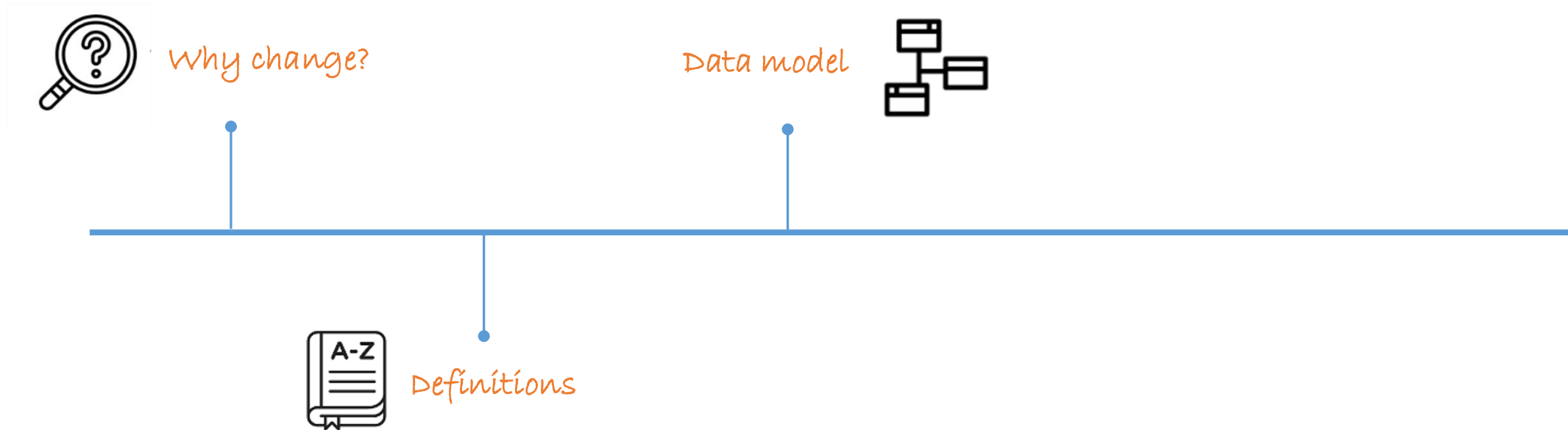
Typical question

"Are we doing the right things?"

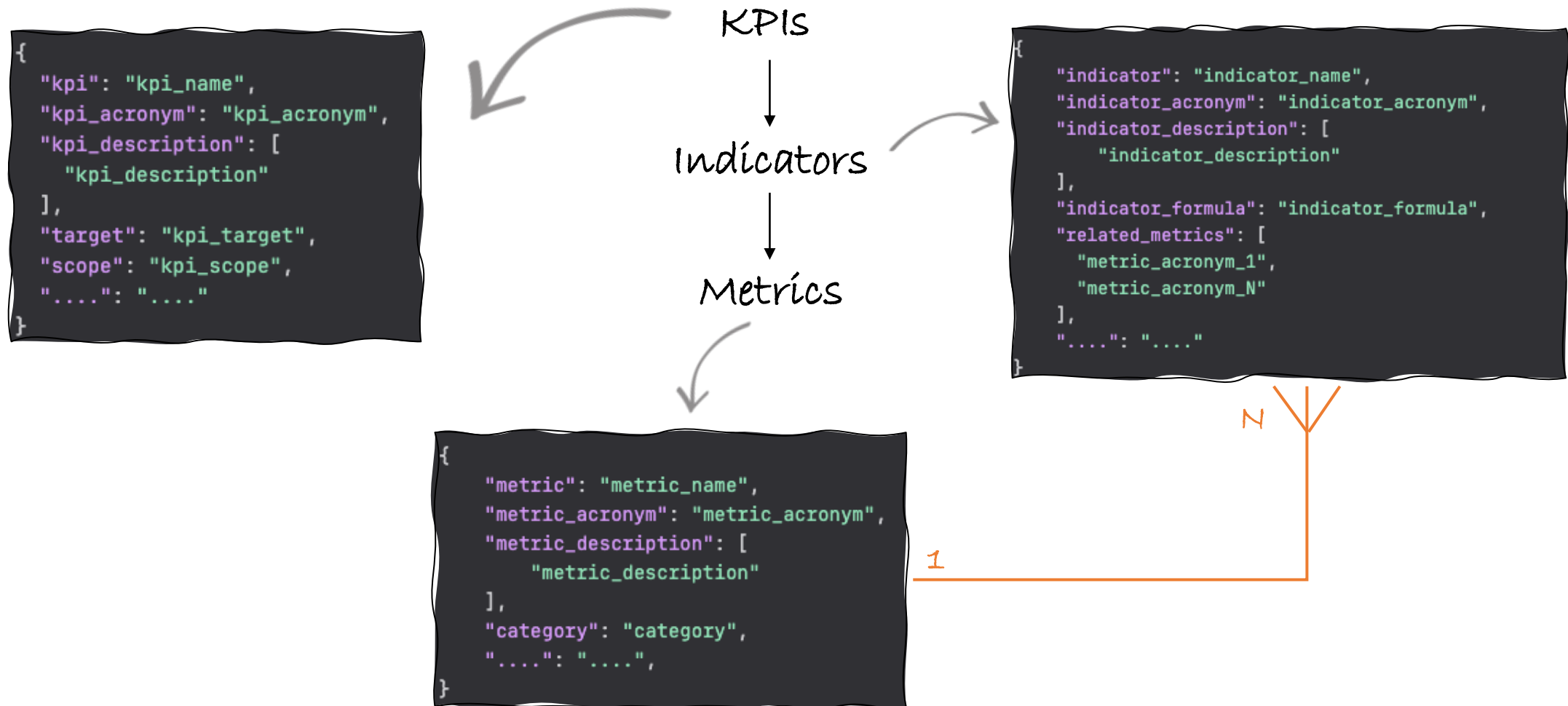
"Did it work?"



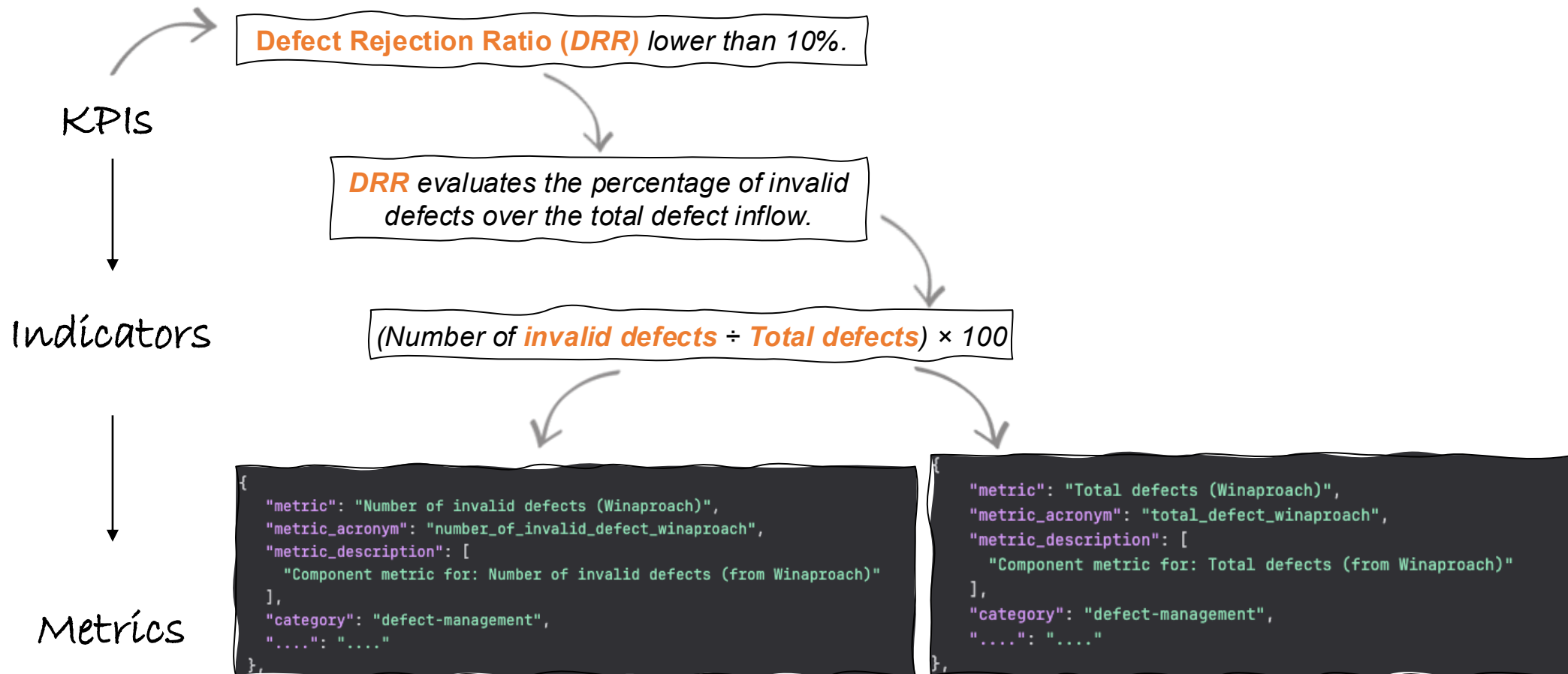
The journey to the standardization



Data model

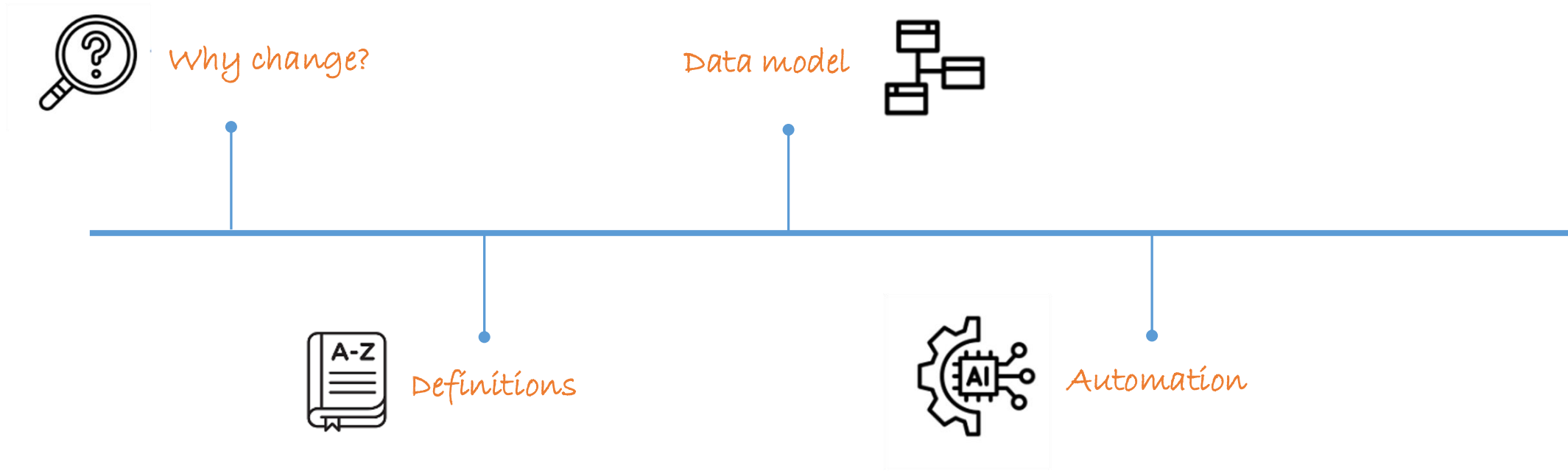


Mapping a real case





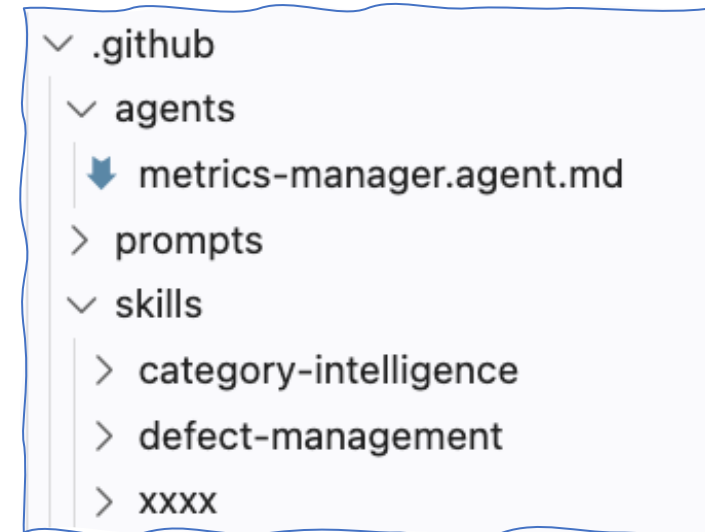
The journey to the standardization

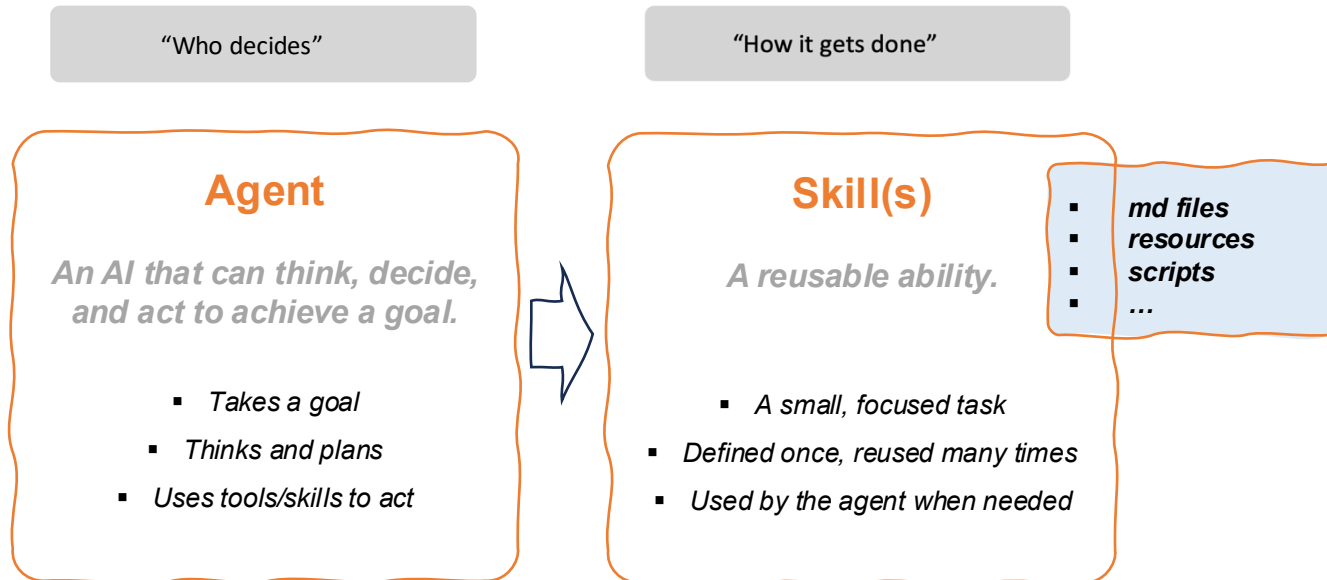




Metrics manager framework powered by AI

- No need to write in JSON
- Focus shifted from visualizations and dashboard to KPIs, indicators and metrics definitions
- Standard way to calculate metrics and indicators
- Improve cross-team KPI comparability







Dedicated skills by category

- *Product Quality*
- *Testing Efficiency*
- *Change Efficiency*
- *AI Maturity*
- *Defect Management*

Approach

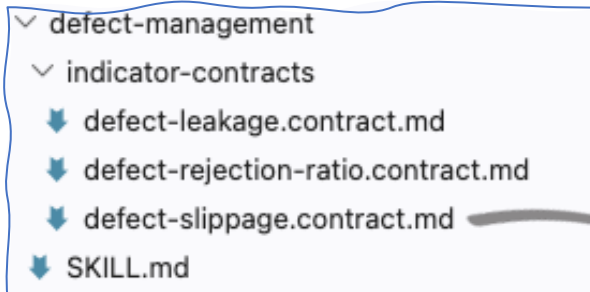
- *Generic core skill*
- *Family-specific contracts*
- *Indicator-agnostic framework*
- *Source-agnostic defaults*

Benefits

- *Maintainable*
- *Core skill reusable*
- *Independent indicator evolution*

Maturity of defect handling

▼ defect-management
▼ indicator-contracts
 ↘ defect-leakage.contract.md
 ↘ defect-rejection-ratio.contract.md
 ↘ defect-slippage.contract.md
 ↘ SKILL.md



SKILL.md

- *Purpose*
- *Core Defect Definitions*
 -

Core Defect Definitions

These base definitions are owned by this skill and may be refined by submodules.

Valid defect: defects raised minus rejected/invalid defects.

Invalid/rejected defect: a defect that has been excluded from the valid count. The criteria for rejection are source-dependent and defined per source section below.

Submodule skills may extend or restrict these definitions for their specific calculation context (e.g. phase-scoped, source-scoped).

defect-slippage.contract.md

- *Purpose*
- *Formula*
 -

Formula

$$DSR = (\text{valid defects in phase2} / (\text{valid defects in phase1} + \text{valid defects in phase2})) * 100$$

valid defects in phase = defects raised in phase – defects rejected in phase

Core metric components should always be expressed in terms of defects raised and invalid/rejected defects, not as standalone valid-defect metrics.



"What you ask"

"Who decides"

"How it gets done"



Prompt

The text you write to tell the AI what to do.

- Tells the AI what you want
- Can be one-time or repeated
- Guides the behaviour

Agent

An AI that can think, decide, and act to achieve a goal.

- Takes a goal
- Thinks and plans
- Uses tools/skills to act

Skill(s)

A reusable ability.

- A small, focused task
- Defined once, reused many times
- Used by the agent when needed

- *md files*
- *resources*
- *scripts*
- ...





"What you ask"

"Humans decide"

"How it gets done"



Prompt

The text you write to tell the AI what to do.

- Tells the AI what you want
- Can be one-time or repeated
- Guides the behaviour

Agent

An AI that can think, decide, and act to achieve a goal.

- Takes a goal
- Thinks and plans
- Uses tools/skills to act

Skill(s)

A reusable ability.

- A small, focused task
- Defined once, reused many times
- Used by the agent when needed

- *md files*
- *resources*
- *scripts*
- ...





DSR Acceptance — Winaproach Scenario

Use this file as the single source of assumptions for DSR from Winaproach data.

Assumptions



hey metric manager, calculate DSR for winaproach using the dedicated prompt

+ metrics-manager Auto

- ...

+ metrics-manager Auto

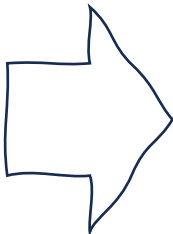
}

Post-Process Rules

Apply the following deterministic flow for DSR components using two parallel queries:

- 1. Query Winaproach using mandatory fields plus the following filters:

•



Defect Slippage Ratio Report

Generated at (UTC): 2026-09-14T07:58:21

Valid Defect Definition

A valid defect is a raised defect that is not rejected/invalid.

Valid Defect Calculation Rule

valid defects = raised defects - invalid defects

DSR Component Summary

Component	Value
Defects raised in phase1	106
Defects raised in phase2	41
Invalid defects in phase1	12
Invalid defects in phase2	4
Valid defects in phase2	37

DSR Calculation

DSR = valid_phase1 / (valid_phase2 + valid_phase1)

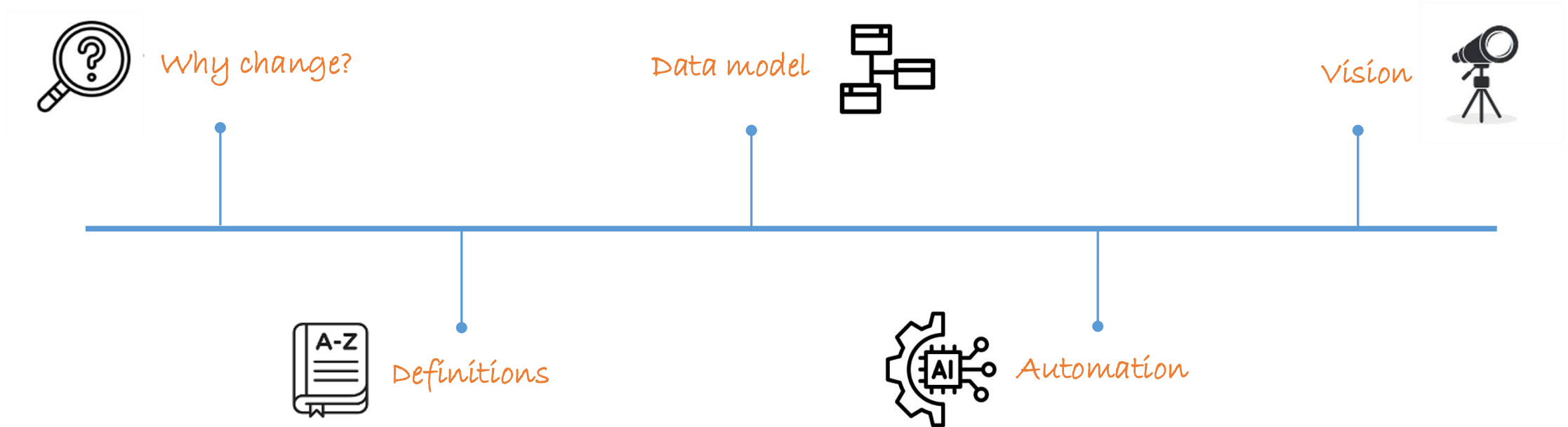
- DSR ratio: 0.717557
- DSR percentage: 71.76%

Raised Defects in Acceptance

Record ID	Status	Summary
12345678	XX	Timeout raised in phase 1



The journey to the standardization





Vision

Governance

- *Skills Definition Ownership*
- *Metrics Definition Ownership*
- *Shared Skills Governance*

Architecture

- *Maintainable*
- *Modular*
- *Supporting Incremental Growth*

Benefits

- *Reusable catalog of standardized metrics*
- *Harmonized indicator and metric definitions*
- *Improved cross-team KPI comparability*

9ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



15 septembre 2026
17h à 23h



ETSI, Valbonne



Soirée du
test logiciel

☒ Sophia Antipolis

SCAN ME



**Merci de
votre écoute !**

Votre avis nous intéresse