# Trusted Data: Why It Breaks & How to Fix It

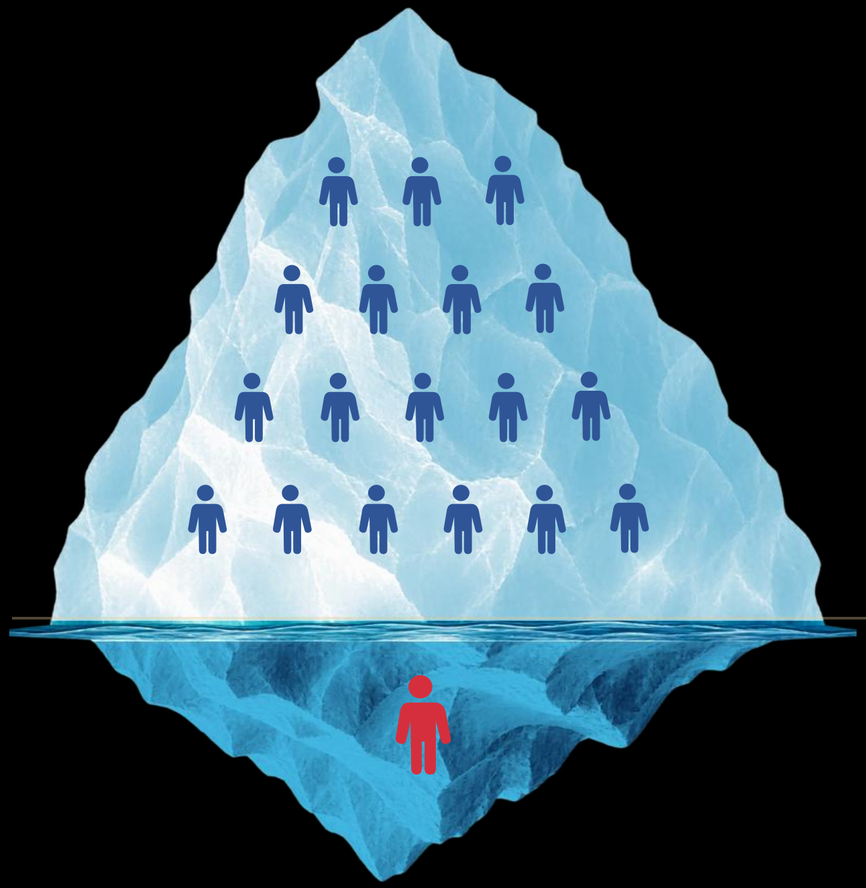


**Etza Lukmanto**

DATA VALUE LEAD, SKYCITY

WHY IT BREAKS

# A good data-quality result can still hide serious risk



**99%** records **passed** the data-quality checks

**Overall result looked healthy**

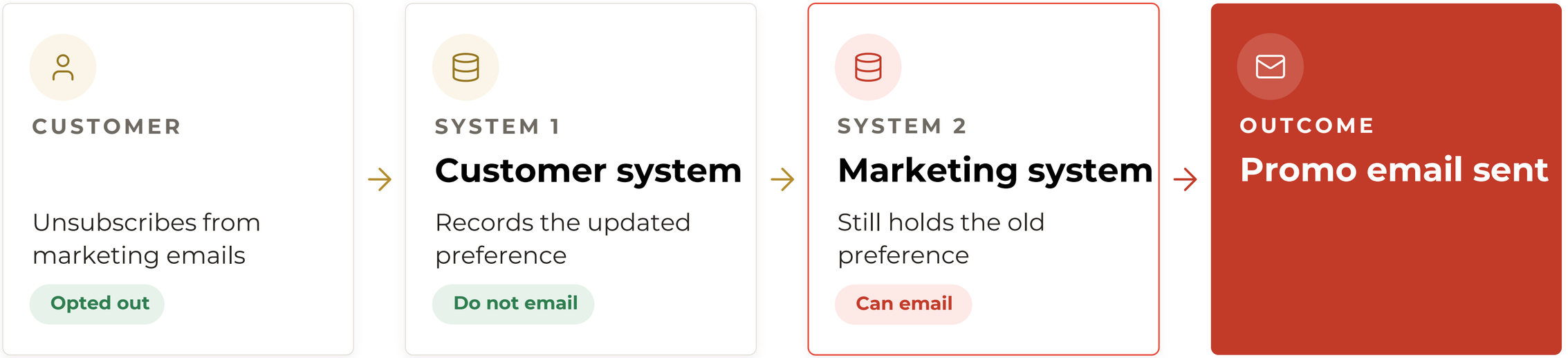
**1%** **The real risk was hiding in the 1%**

**The result looked excellent, but what mattered was hidden below the surface.**

WHY IT BREAKS

# What was hiding inside the 1%?

A small data error can lead to the wrong customer action




**One customer. Two different answers.**

## Not every data-quality failure has the same impact.

Sometimes 1% carries the biggest compliance risk.

Figures are for illustrative purposes only

WHY IT BREAKS

# How do we check data quality at scale?

Checking one customer is easy. What about 500,000?




**500k**

**Customer records**

**100**

**Validation checks**

**Data changes over time**

## How do we check all of this and keep checking?

Figures are for illustrative purposes only

WHY IT BREAKS

# Data-quality checks need more than data. They need context.

**CUSTOMER SYSTEM**

STORED VALUE

1

DEFINITION

Customer is active

**MARKETING SYSTEM**

STORED VALUE

1

DEFINITION

Customer is inactive

## Two systems. Same value. Different meaning.

**TECHNICAL**
What does each stored value mean?

**BUSINESS**
What makes a customer Active or Inactive?

**Together: Agree on what the check should validate.**

**AI can build the checks faster. But it needs the right context first.**

Figures are for illustrative purposes only

HOW TO FIX IT

# From a spreadsheet to a working solution

How I used AI to build reusable data-quality checks

TRADITIONAL APPROACH *(ESTIMATE)*

~3 weeks

Build data-quality validation scripts and reporting

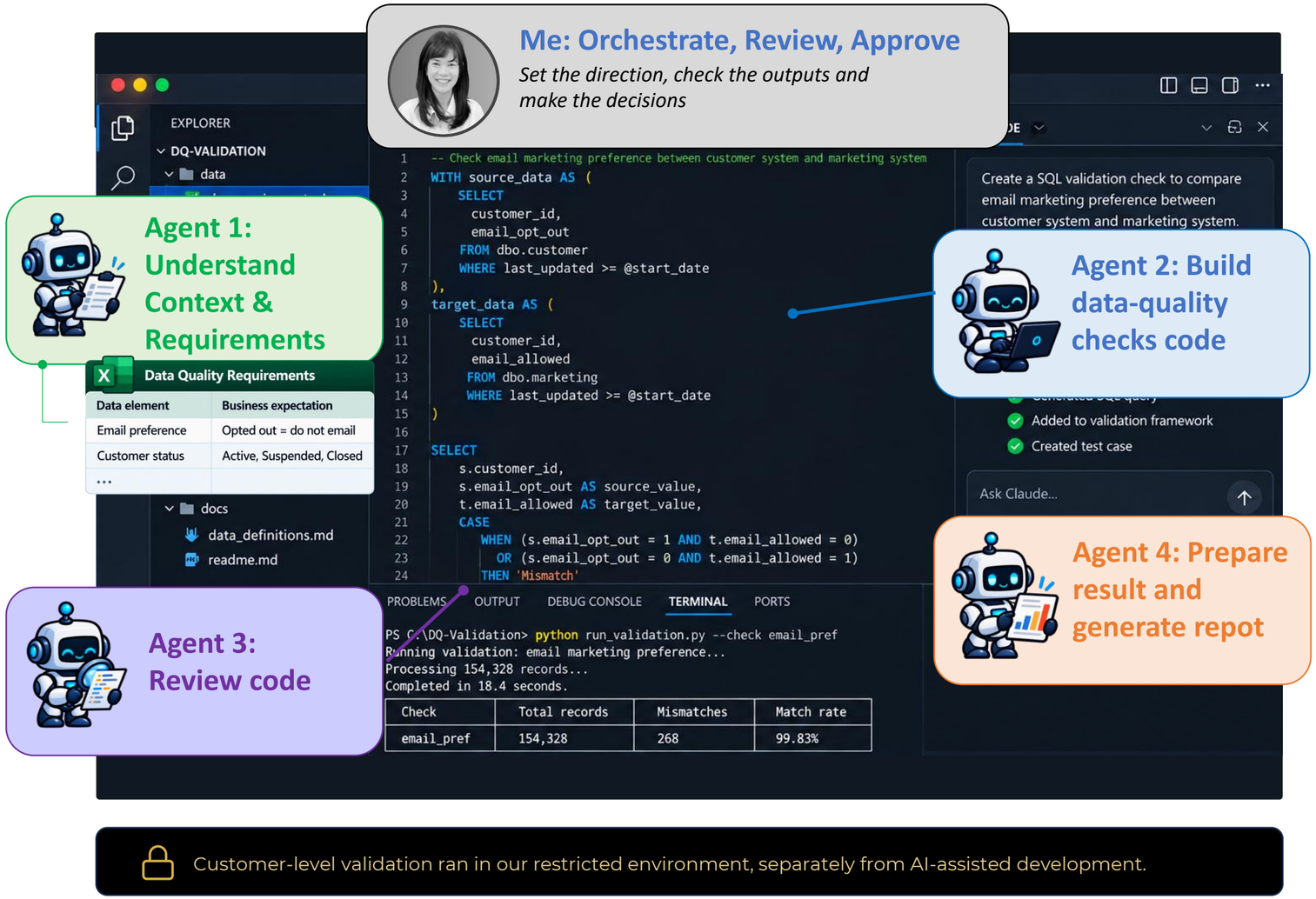
AI-ASSISTED APPROACH (ACTUAL)

**3 days**

Built the initial reusable validation and reporting solution

**One person. AI-assisted development.**
**Repeatable data-quality checks.**

*"AI helped me build it faster. But I still had to make sure it was checking the right things."*




Customer-level validation ran in our restricted environment, separately from AI-assisted development.

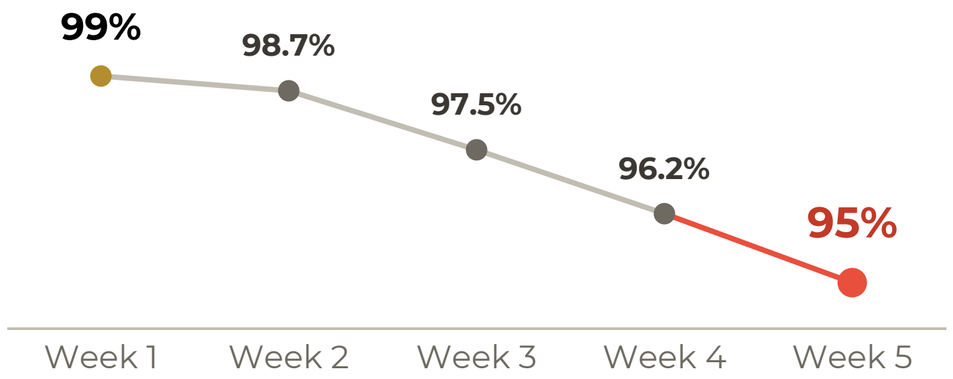
HOW TO FIX IT

# Data doesn’t stand still. Neither should our checks.

Keep checking as data changes. Expand coverage as our needs grow

## OVERALL DATA-QUALITY TREND

% of records that passed the checks




## WHERE SOURCE & TARGET DON’T MEET EXPECTATION

Number of records not meeting expectations

| Information checked | Wk 1 | Wk 2 | Wk 3 | Wk 4 | Wk 5 |
|---|---|---|---|---|---|
| Account number | 0 | 0 | 0 | 0 | 0 |
| Customer status | 0 | 0 | 0 | 0 | 0 |
| **Email preference** | 1 | **2** | **3** | **4** | **6** |
| **Phone preference** | 1 | 1 | **2** | **3** | **4** |

**Check**
Run the same checks regularly

→

**Investigate**
Understand what’s different and what’s the impact

→

**Act**
Resolve issues or update the checks for new scenarios

→

**Check again & evolve**
Repeat existing checks. Add new ones as needed

**Detect changes. Understand the impact. Take action**

Figures are for illustrative purposes only

WHO IS RESPONSIBLE

# Detecting the problem is one thing. Preventing it is another.

The data-quality checks tell us what's wrong. People need to make sure it doesn't happen again

**We keep finding the same issue. Who makes sure it gets fixed?**

BUSINESS & COMPLIANCE

## Agree what's correct

Define the expected behaviour and what matters

DATA & TECHNOLOGY

## Find and fix the cause

Investigate and correct the underlying issue

ACCOUNTABLE PROCESS OWNER

## Make sure it stays fixed

Ensure safeguards are in place, issues are followed up, and fixes are checked

# Better technology. Not necessarily better data.

What needs to change?

1 **Understand what matters, not just the percentage**
Know the business impact and agree on what correct data should look like.

2 **Make data-quality checks ongoing**
Use AI to build solution faster, monitor data regularly, and improve the solution as needs change.

3 **Make sure people take responsibility**
Investigate issues, fix the underlying causes, and prevent them from happening again.

**That's how we build data we can trust.**

# Thank you