

Zero Unexplained Differences: Proving a New MDMS Before Go-Live

Why a go-live parallel should start from the outputs a utility depends on, give both systems the same inputs, and explain every difference, including the legacy system's estimates

PRODUCT	Grid Data Management Platform, with the Grid Data Conversion Testing Tool
AUDIENCE	MDMS replacement program teams and sponsors, billing, finance, regulatory and meter operations leaders, VEE analysts, and the consultants and reviewers who evaluate a new MDMS

The parallel run is where the decision gets made

When a utility replaces its meter data management system, the safest path to go-live is a parallel run. For several billing cycles the new MDMS and the incumbent run side by side on the same production data, and each produces everything it normally would: billing determinants for the CIS, settlement submissions, customer usage data, exception worklists and reports.

The implementing vendor then has to show that switching will not change bills, settlement, customer data or operations in ways the utility has not agreed to, and explain every difference in detailed reports. Those reports are reviewed by many people: billing operations, finance, regulatory, meter operations, IT, program sponsors, outside advisors and sometimes the regulator. Whoever leads the evaluation, each of them needs to be able to check the evidence for themselves.

Many parallel runs never get there. They produce thousands of differences in the first week and end in long arguments about which of them matter.

Why parallel runs go wrong

Nobody agreed what "the same" means. If the run starts by comparing interval reads, it measures what is easy to compare rather than what the utility depends on. The billing team cares about determinants and dollars. Finance cares about settlement. Operations cares about exception workload. Each needs its own standard of "close enough", agreed before the results come in.

Most differences are not about the MDMS at all. The two systems receive slightly different data, a resend reaches one and not the other, master data disagrees, a multiplier or time convention differs, a rule parameter was set differently, or someone edited a read in one system only. Each of these creates differences by the thousand, and none of them says anything about which system processes meter data better.

The two systems are compared at different moments. If one system's output is taken at dawn and the other's an hour later, late-arriving reads alone will make them disagree.

The incumbent is a black box. Legacy configuration is often poorly documented, customized over years, or understood only by people who have since left. When its estimate differs from the new system's, nobody can say how it was produced.

Different is not the same as wrong. When the two systems estimate a gap differently, the new estimate may be the better one. A test that only rewards matching quietly pushes the new system to copy the old system's mistakes. And when the new system is more correct, bills still change, which is a decision for the utility, not for the project team.

Decisions get lost. Why a difference was accepted, who approved it, and whether it should still be accepted next month tends to live in email threads and meeting notes.

Principles of a parallel run that proves something

Define the outputs before comparing anything. Start with an output contract: every downstream use of meter data, from CIS billing and settlement to the customer portal, exception worklists and regulatory reports. Each output gets an owner, a tolerance and an exit criterion, all agreed and locked before the run starts.

Give both systems the same inputs. Both systems should receive the same head-end data, including resends, late files and gaps, with every delivery kept in an archive so any day can be replayed. A daily report should confirm that the inputs really were the same.

Settle the setup before counting differences. Reconcile master data first: meters, channels, multipliers, units, rates and billing cycles, meter exchanges and move-ins. Map every legacy validation and estimation rule and every quality code to its counterpart. Agree conventions such as time stamps and daylight saving, and freeze configuration on both sides for the proving period. Readiness gates make sure none of this is skipped.

Compare from the top down, at the same moment. Compare billing determinants for every account first, then the money they produce, then daily usage, intervals, VEE outcomes and exception workload, timeliness, and the interface files themselves. Both systems are compared as they stood at the same cut-off. The new platform can show what it was serving at any past moment, because every value it serves comes from a decision ledger.

Give every difference exactly one explanation. Each difference is classified as an input difference, a setup difference, an agreed difference in method, a defect in the legacy system, or a defect in the new system. Anything without one of those is unexplained. The exit test is zero unexplained differences, not zero differences.

Never fix a legacy defect silently. When the new system is more correct than the old one, bills will still change. The utility decides, in writing, whether to reproduce the old behavior or to accept the change with customer communication and any regulatory approval it needs.

Treat estimates as a separate test. Where both systems used actual reads, the values should match. Where they estimated, judge the estimates on quality: hold back actual reads, let each system estimate them, and compare with the real values, looking at bias first. Each differing estimate is judged better, the same or worse, with the evidence shown.

Learn how the legacy system really estimates. Working only from the incumbent's own output, each legacy estimate can be explained in plain language, and across the population those explanations add up to the rules the legacy system actually applies. VEE experts confirm or reject each one, and the result sits beside the legacy documentation and the new configuration.

Agree how manual edits are handled. Analysts keep editing reads during the run. Either mirror each legacy edit into the new system, or compare edited reads as a separate group. Pick one up front.

Test the hard cases on purpose. Some situations will not happen often enough during the run to prove anything: a duplicate resend, late data after the bill, a register rollover, reverse flow, a mid-cycle meter exchange, a long communication outage. Scripted scenarios on test meters create them deliberately, in both systems, and check the result at every level.

Report findings so every stakeholder can check them. A daily reconciliation report, difference detail down to the account, meter and interval, the input and master-data reconciliations, the legacy rules as observed, the estimate verdicts and the dollar impact should all be available to everyone reviewing the program, in formats they can open and filter themselves.

Sign off on evidence. Exit criteria are measured every day. Readiness is signed by the output owners, and the evidence pack behind the decision is tamper-evident, so anyone can confirm later that it has not changed since it was signed.

AN ILLUSTRATIVE EXAMPLE (HYPOTHETICAL)

A mid-sized utility is replacing its MDMS. Before the run starts, billing operations, finance, regulatory and meter operations agree the output contract: which outputs matter, how close is close enough, and what must hold before go-live. Both systems receive the same head-end data, and each morning the previous day's findings go to those owners, the program sponsor and the consultant advising the program.

In the first week, most differences trace to setup. A group of commercial meters carries a multiplier in one system only. A set of accounts moved in mid-cycle in one system's master data but not the other's. Each is fixed at the source, the affected days are compared again, and the differences disappear.

With setup settled, the remaining differences are mostly estimates. For each legacy estimate, reviewers can open a plain-language explanation of what most likely produced it. The utility's VEE expert confirms most of the legacy rules found this way. One finding surprises everyone: the legacy system sometimes based new estimates on days that were themselves estimated, which the utility's own estimation policy does not allow. That is recorded as a defect in the legacy system, and billing operations decides to accept the corrected behavior, with a note prepared for customer service.

The hold-back test shows the new system's estimates are less biased for long gaps. The VEE expert approves that as an agreed difference in method, and the dollar impact stays within the billing tolerance. Scripted scenarios run on test meters, including a duplicate resend, late data after the bill and a mid-cycle meter exchange, and both systems handle them as expected.

Over the following billing cycles, the daily report shows zero unexplained differences and every exit criterion met. The output owners sign the readiness decision. The evidence pack goes into the program record, and the utility cuts over knowing where the new system matches the old one, where it differs, and why.

Questions to ask any MDMS vendor or integrator

- Will the parallel run start from an agreed list of outputs, each with an owner, a tolerance and an exit criterion?
- How do you make sure both systems receive the same inputs, and can any day be replayed?

- Are master data, rules, quality codes and conventions reconciled before differences are counted?
- Are both systems compared at the same moment, from billing determinants down to intervals?
- Does every difference get exactly one explanation, and is "unexplained" visible until it does?
- When the new system is more correct than the old one, who decides, and is the decision recorded?
- How are estimates judged, and can you explain how our current MDMS produced its estimates?
- Will every stakeholder get detailed reports they can check for themselves, down to the account and meter?

Proof, not parity

Matching the old system is not the goal. Being right, and agreeing every change, is. A parallel run should leave the utility and its regulator with proof that the new MDMS produces the same answers or better ones, with every difference explained and every decision on record.

The Grid Data Conversion Testing Tool puts this method into practice for Grid Data Management Platform replacements. It manages the output contract, keeps both systems' inputs identical and replayable, reconciles setup, compares both systems from billing determinants down to intervals, explains the legacy system's estimates, and produces detailed reports that every stakeholder can review, whoever leads the evaluation. Readiness becomes a signed, tamper-evident decision. The Grid Data Management Platform supports more than 15 million meters and over 1 billion interval reads per day in production.

Learn more about the [Grid Data Management Platform](#), or [talk to our team](#) about proving your MDMS replacement before go-live.

ABOUT PERINIMBLE

PeriNimble builds mission-critical grid data software for energy and utility companies. Founded in 2022 and headquartered in Bloomington, Minnesota, we support more than 15 million meters and over 1 billion interval reads per day in production.

Built for systems that can't afford failure. · perinimble.com · perinimble.com/contact