



E-BOOK · EXECUTIVE GUIDE

SAP Implementation

Why most problems are not SAP problems — and what to prepare in the foundation before it becomes the critical path.

The problem is almost never the software

When an SAP implementation goes wrong, the conversation usually starts by blaming the system. But anyone who watches go-lives from up close sees the same pattern every time: the software does what it promises — what fails is the **foundation of the IT environment** around it. Data nobody cleaned up, integrations that never ran end to end, an environment sized for the demo instead of for the operation.



Figure 1 — The tower is what shows up on the slide; the excavated foundation is what carries the go-live.

This is not bad luck or incompetence. It is the consequence of how the work is framed: the project plan covers processes, configuration and training — and treats everything that supports those three as 'infrastructure, we will deal with it later'.

The right question — it is not 'will SAP work?', it is 'is the environment around SAP ready for it to work?'. The next pages list what that environment is made of.

The two layers of every project

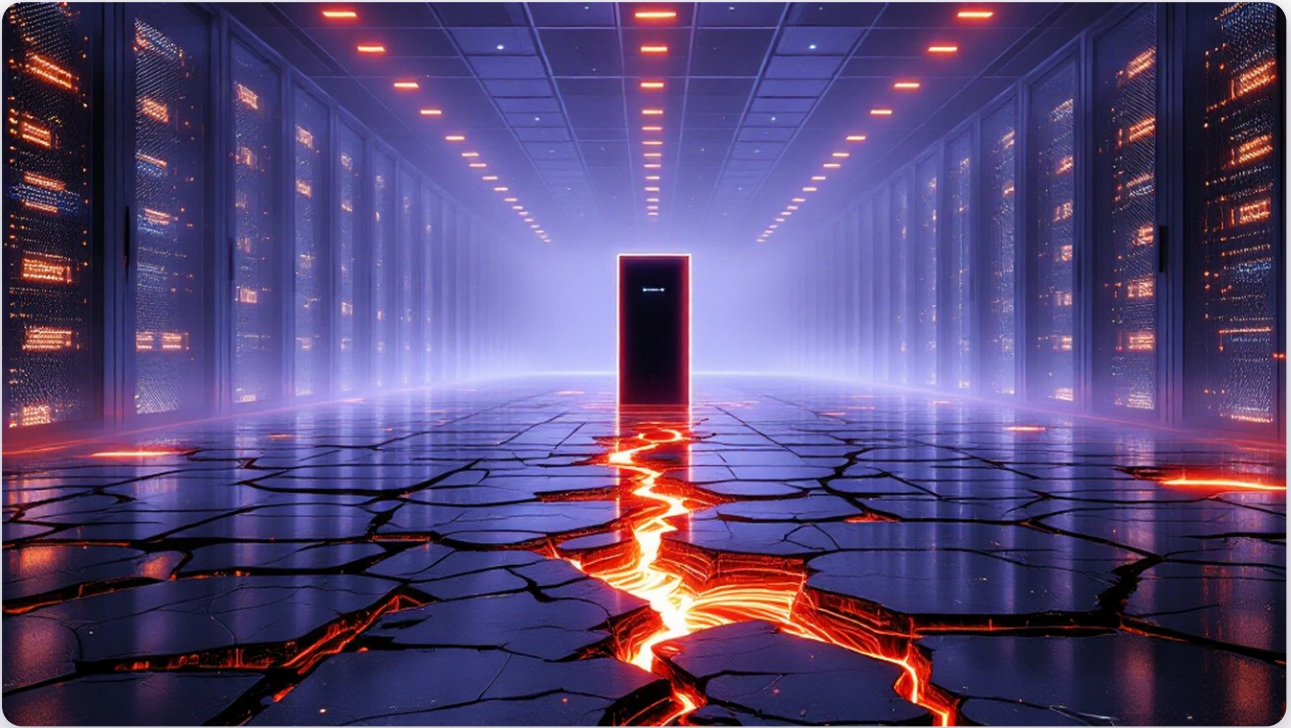


Figure 2 — The flawless environment and, under the floor, the cracks nobody addressed: that is where incidents come from.

Every SAP program has two layers. The top one is visible: scope, process design, custom development, training, the go-live date on the slide. The one underneath rarely has a declared owner:

- **Environments** — how many, at what size, refreshed from production how often. Testing on an environment smaller than the real one is rehearsing the wrong play.
- **Master data** — material, customer, vendor, cost center. The quality of the legacy records defines the migration effort and the amount of errors on day 1.
- **Integrations** — banks, tax authorities, WMS, e-commerce, payroll. Each one is a contract between systems that has to be tested at volume and with simulated failure.
- **Access and SoD** — roles designed during the project, not improvised the night before; segregation of duties settled before the auditor asks.
- **Performance** — load testing with the real volume of the operation, not with ten demo orders.
- **Cutover** — the switchover sequence, timed and rehearsed.

Rule of thumb — if any of these six items has no named owner in your plan, it is already a risk. The good news: all six can be addressed early, when they cost little.

The foundation runs across every phase

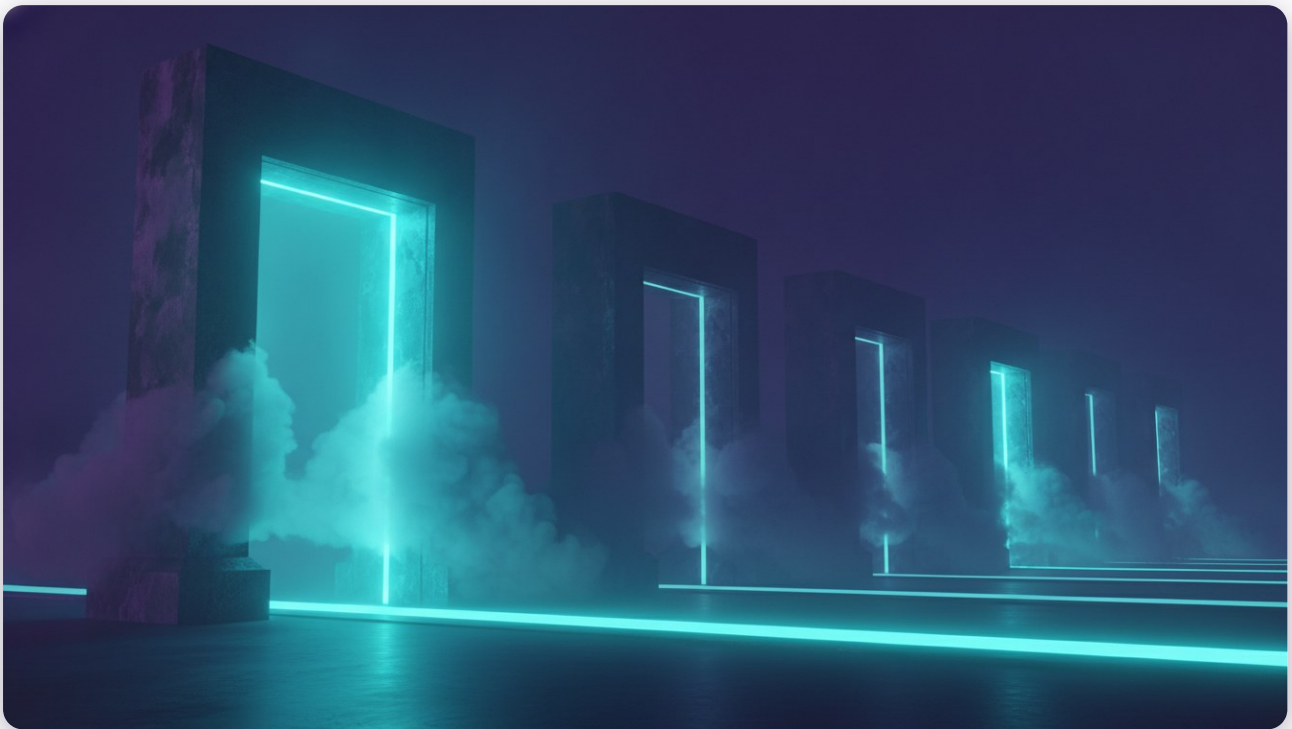


Figure 3 — Five gateways, a single beam of light crossing all of them: the foundation does not belong to one phase.

The implementation method is well known and it works. What separates calm projects from heroic ones is **when** the foundation enters the conversation:

Prepare

Environments defined and provisioned before configuration starts. Sizing done with the volume of the operation — including the history that will come over from the legacy system.

Explore

Real fit-to-standard: every gap becomes a deliberate decision between changing the process and building something. This is also where the quality of the data to be migrated gets measured.

Realize

Configuration, integrations and migration move together. An integration tested only at the end is an integration that was not tested.

Deploy

Integration testing at real volume, with real users and with exception scenarios — the invoice that gets rejected, the payment that fails, the file that arrives malformed.

Run

Rehearsed cutover and reinforced support in the first weeks, when questions outnumber defects by a wide margin.

Tipping point — when the foundation shows up late, it stops being preparation and becomes the critical path: the project starts waiting for an environment, for data and for access.

Cutover: rehearse until nothing surprises you

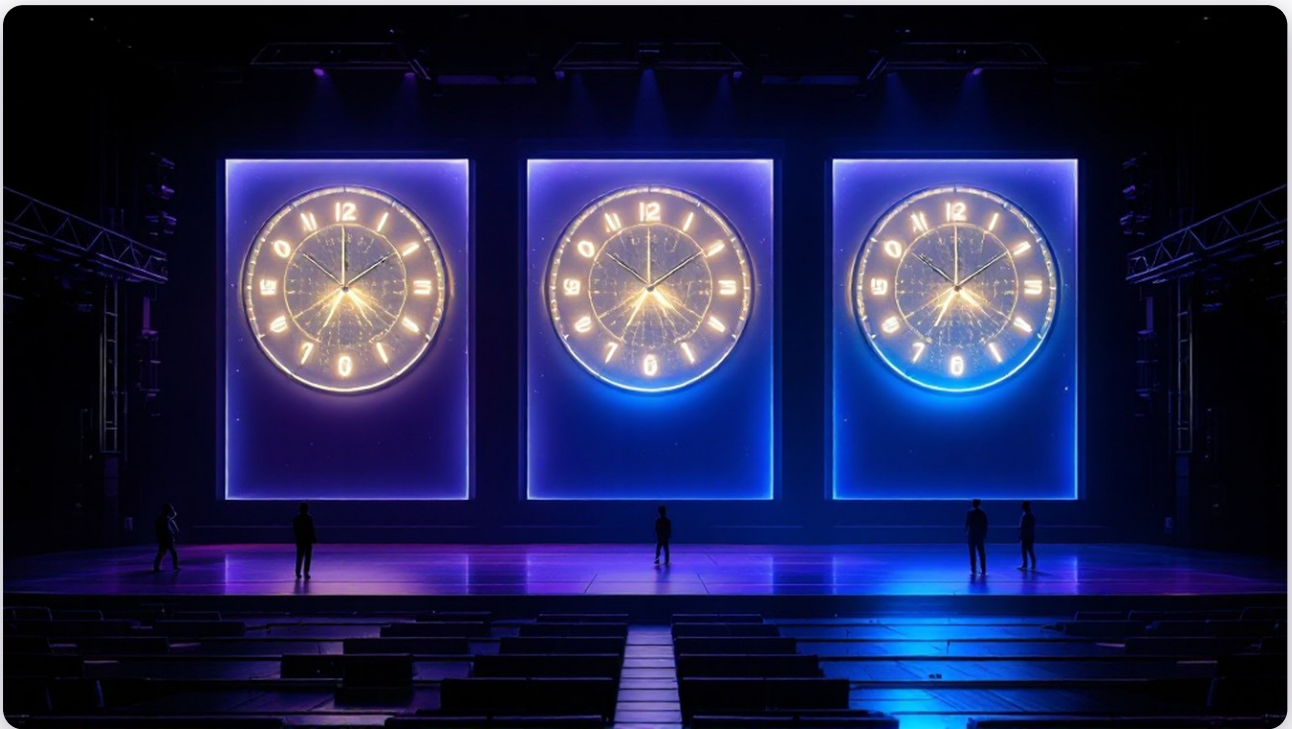


Figure 4 — Three identical stages and the clock above them: the switchover rehearsed until it becomes routine.

Cutover is the moment when everything that was built has to happen in sequence, inside a fixed window, with the business standing still. It is the point of lowest tolerance for improvisation — and still the one most often underestimated.

- **Rehearsal 1** — finds what is missing: a step with no owner, a script that does not exist, a forgotten dependency.
- **Rehearsal 2** — measures the real duration of each step. This is where the window stops being an estimate and becomes a number.
- **Rehearsal 3** — runs exactly like the real day, with the people who will be on call, including the *rollback* decision.

Go criteria — the decision to go live should not depend on optimism. Define the objective criteria up front (data migrated and reconciled, integrations green, the window fitting inside the rehearsal) and the point of no return.

Add to that a **hypercare** plan: in the first weeks, a direct channel for users, reinforced monitoring and someone watching the operational indicators — invoices going out, payments running, integrations flowing.

The mistakes that keep repeating

1. Treating data migration as a technical task

Migrating means deciding: what comes over, what stays behind, what gets fixed first. That is a business decision with technical support — and it needs time from the people who know the records.

2. Testing with toy volumes

A scenario that runs with ten documents can freeze with ten thousand. Without load testing at real volume, performance is discovered by the end user.

3. Leaving access for the end

A badly designed role creates two opposite problems: users blocked on day 1 and, in the rush to unblock them, permissions so broad that the auditor will raise them later.

4. Ignoring the legacy that stays alive

Almost never does everything migrate at once. The systems that stay need integration, coexistence and — sometimes — access to archived history.

5. Setting the date before the capacity

A date chosen by the fiscal calendar or by the board, never checked against the real capacity of the team and the environment, is debt taken on day one.

The pattern behind all five — none of them is a software problem. All of them are preparation problems — which is exactly why all of them can be avoided ahead of time.

Executive checklist

Questions to ask at the next status meeting — the answers tell you more than the percentage complete:

- ✓ Are our environments sized for the real volume of the operation — or for the demo?
- ✓ Who owns master data quality? Have we measured how many records need fixing?
- ✓ Have all integrations already run end to end, with simulated failure?
- ✓ Are access roles and segregation of duties designed — or will they be sorted out the night before?
- ✓ Have we run performance tests at real volume, with the reports the business actually uses?
- ✓ How many cutover rehearsals are in the plan — and is the window measured or estimated?
- ✓ Who looks after the foundation (environments, security, cost, performance) while the functional team looks after the process?

How to use it — if three or more answers are 'not yet', the risk is not in the functional scope: it is in the foundation. It is cheaper to deal with it now than on go-live weekend.

Who looks after the foundation of your program

An SAP implementation has two halves: the functional one (processes, configuration, training) and the **IT foundation that carries the program**. That foundation is where Inove Solutions works, with people who know the environment from the inside:

- **Environments and infrastructure** — sizing, provisioning and refresh windows dimensioned for the real testing cycle.
- **Data** — quality, migration and volume (including archiving, when history weighs).
- **Integrations** — everything that talks to the ERP, tested at volume and with simulated failure.
- **Cybersecurity and access** — roles and SoD designed during the project, with an audit trail.
- **Cloud and FinOps** — the cost of multiple environments under control from start to finish.
- **Application management** — our 'life insurance' model, from hypercare to steady-state routine.

We serve companies such as **Petrobras, Nespresso and Qualicorp** — and we treat the foundation with the same rigor the functional side gets.

Our vision — we want our clients to have no IT headaches and to be happy. Including on go-live weekend.

Does your SAP program already have an owner for the foundation?

Ask for a foundation assessment — environments, data, integrations, access and performance — before it becomes the critical path.

To go deeper, at the Inove Academy: quick guide **S/4HANA: brownfield vs. greenfield** · e-book **SAP Archiving** (less data, shorter project) · infographics on implementation and SAP practices — inovesolutions.com/en/inove-academy-en

inovesolutions.com/en/contact · +55 11 3554-9450