SAP© PROJECTS

Why (some) of them fail?

A Practical Guide to
recognizing and avoiding
classic SAP Project Problems

Bogdan Gorka | PMBG.EU

Table of Contents

SAP© Projects – why (some) projects fail?

Cover Design © 2025 Bogdan Gorka
Photos:
Source: https://www.flickr.com/photos/jaygalvin/28874178626
https://www.flickr.com/photos/tags/jaygalvin title: Caution Icy Road sign

Edition 1
First published: September 2025

Disclaimer
PMBG Bogdan Górka is an independent consulting practice registered in Poland. This company is not affiliated with, sponsored by, or endorsed by SAP SE, Atlassian Corporation Plc, or any of their subsidiaries. All content is independently developed for informational and educational purposes only.
SAP, the SAP logo, and product names such as SAP S/4HANA, SAP Activate, and SAP Focused Build are trademarks or registered trademarks of SAP SE in Germany and other countries.
Atlassian, the Atlassian logo, Jira, and Confluence are trademarks or registered trademarks of Atlassian Corporation Plc in the U.S. and other countries

Excluded liability
Although every effort has been made to make this publication accurate, the information contained in this publication is distributed without any warranty, either express or implied, and distributed as is at your own discretion. The author assumes no responsibility for any potential mistakes or printing errors made in this publication's content. When you use the content of this publication, including all information and instructions, you agree to do it at your own risk and the author assumes no responsibility for any potential damages caused directly or indirectly as a result from using this publication.

1 Foreword

Hello, and thank you for downloading this white paper. My name is Bogdan Górka, and over the recent years I've worked at the intersection of SAP methodologies, enterprise program delivery, and Atlassian tools. I wrote this publication to help project leaders like you avoid common pitfalls in SAP implementations and I hope these insights will help you get better prepared for your SAP program.

In the era of digital transformation, SAP implementations are seen as critical and strategic initiatives. However, ERP projects like SAP system implementations remain among the most failure-prone undertakings in enterprise IT.

This paper is for CIOs, Program Directors, SAP delivery leads, and Transformation Sponsors who are accountable for results. Through a structured analysis of some of the most expensive and visible SAP Project failures, I wanted to extract universal lessons and provide you practical, governance-oriented suggestions for preventing similar problems in your SAP Project.

Find me on LinkedIn https://www.linkedin.com/in/bgorka/ and let me know what you think about this publication. Visit my website https://pmbg.eu and find out more about what we do.

Wishing you success in your next transformation,

Bogdan Gorka
PMBG.EU

2 Introduction

SAP Projects are among the most complex and strategically important programs an enterprise can undertake. They touch every business process, involve every department, and impact every user. When done well, they become the digital backbone of the organization. When done poorly, they trigger financial losses, reputational damage, and even executive-level fallout.

This white paper introduces a structured approach for understanding ERP failure through the lens of the "Requirements-to-Deploy" (R2D) process framework. R2D process organizes a SAP implementation project into six core process areas:

1. Scope Planning
2. Delivery Management
3. Release Management
4. Status Reporting
5. Test Management
6. Documentation Management

Each section of this paper maps specific project breakdowns to these areas, providing clarity on where and how things typically may go wrong.

I draw on over a dozen real-world SAP failure cases — including Lidl, LeasePlan, SPAR, Revlon, and National Grid — to demonstrate the patterns and early warning signs that repeat across industries, geographies, and company sizes. You will find the links to sources at the end of this publication.

My recommended approach to avoiding typical SAP Project pitfalls is grounded in hands-on experience with SAP methodologies, expert interviews, and in-depth analysis of publicly available case studies and publications. By combining real-world failure analysis with structured, actionable recommendations, this guide aims to help decision-makers understand common risks and thus equip them with practical methods and tools to better plan, scope, govern, and execute ERP programs, particularly SAP implementations.

This is not a technical paper. It is a paper about awareness, leadership, accountability, and decision-making in complex transformation environments. I refer often to recommended SAP Activate Methodology and Implementation Roadmaps offered by SAP AG. Check this address for reference: https://me.sap.com/roadmapviewer (User ID is required)

3 Anatomy of Failure: Executive Overview

Enterprise software failures rarely come down to just "bad software." In nearly every case, the real culprits lie in delivery strategy, project governance, and implementation execution. Poor decisions made early in discovery and preparation phases, cascade into irrecoverable misalignments. The ERP platform then becomes the scapegoat for what is ultimately a failure of transformation leadership.

What we can observe in high-profile SAP failures is not a flaw in the technology, but a breakdown in the ecosystem around it:

- Misaligned expectations and assignment of duties between IT and business
- Lack of clarity in the operating model and future process state
- Over-reliance on external consultants without strong internal ownership
- Executive turnover and shifting project priorities

- Unrealistic timelines and inadequate readiness

SAP offers standardization, integration, and scale—but these strengths are undermined when organizations try to bend the system to fit legacy processes rather than adapting their operations to the new platform.

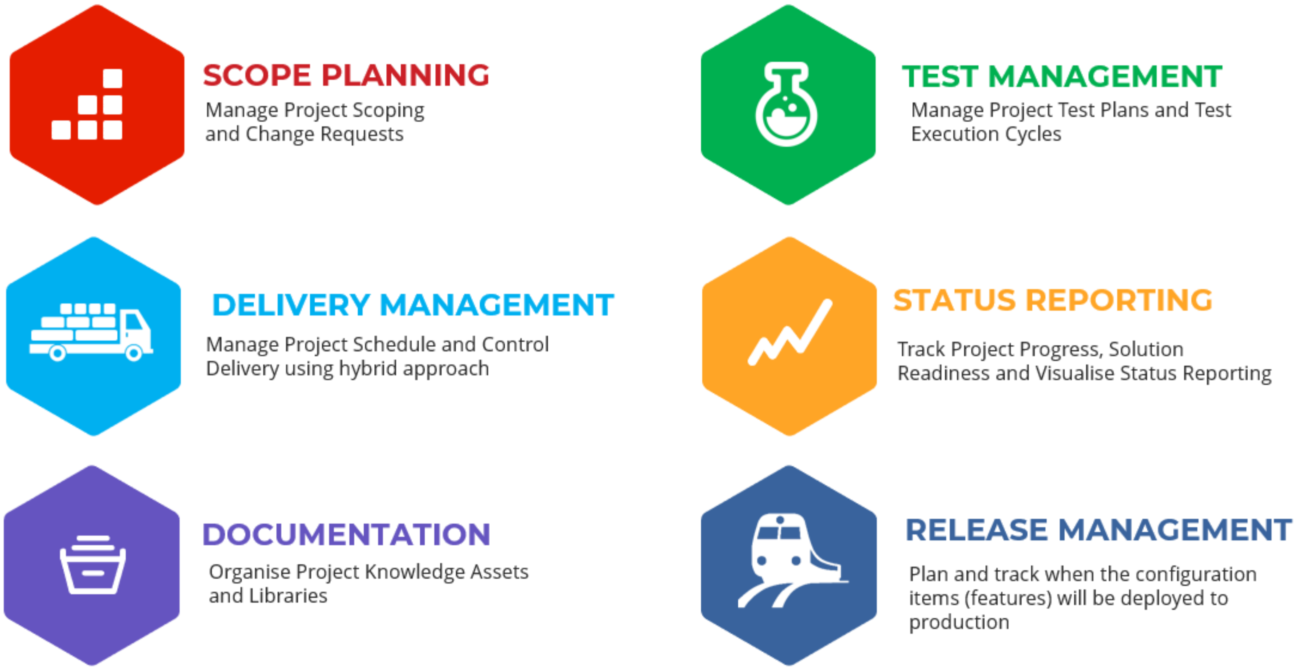
Failures like Lidl and LeasePlan demonstrate that even with a best-in-class ERP platform, the absence of organizational readiness and strategic fit will doom the project. Conversely, smaller projects with clear scope, strong sponsorship, and aligned teams often succeed even under tight constraints.

The purpose of this paper is not to critique SAP system and implementation partners, but to expose the organizational behaviors and program patterns that lead to failure. Because if SAP system is the engine, the problem is usually in the driver's seat.

Read the full story why some SAP Projects fail.

4 Where things may go wrong

The R2D (Requirements-to-Deploy) framework helps precisely identify where and why SAP implementation projects fail. It divides the SAP Project organization into six interdependent areas which are integrated under six Phases of the SAP Activate project lifecycle. Each one is a critical success factor—and also a potential failure zone. What follows in this document is a breakdown of these six areas, supported by real case examples.



SCOPE PLANNING
Manage Project Scoping and Change Requests

TEST MANAGEMENT
Manage Project Test Plans and Test Execution Cycles

DELIVERY MANAGEMENT
Manage Project Schedule and Control Delivery using hybrid approach

STATUS REPORTING
Track Project Progress, Solution Readiness and Visualise Status Reporting

DOCUMENTATION
Organise Project Knowledge Assets and Libraries

RELEASE MANAGEMENT
Plan and track when the configuration items (features) will be deployed to production

5 Scope Planning

Scope Planning failures originate when organizations misjudge the fit between their existing business model and the standard SAP capabilities. Often, companies underestimate how much their internal processes need to change—or overestimate how much the software should adapt to them.

5.1 Typical pitfalls

- **Refusing to standardize processes**
  Organizations frequently insist on maintaining outdated or localized processes that don't align with SAP's standardized model. For example, SAP Retail is built around valuation based on retail price, yet Lidl attempted to preserve its legacy purchase-price valuation logic. Such resistance can necessitate heavy customizations, increasing cost and risk. Standardization is a key design principle in SAP Activate, especially during the Explore phase, where Fit-to-Standard workshops help assess which legacy processes should be adapted.

- **Attempting to customize SAP into legacy shapes**
  Instead of transforming their processes to match SAP's industry best practices, some companies try to bend the system to mirror their old ways of working. This not only erodes the value of the investment but also creates technical debt that makes upgrades harder and long-term sustainability weaker. In SAP Activate, the Fit/Gap Analysis is a critical checkpoint to identify where exceptions are truly needed—and where they are not. Any decisions taken at this time should be recorder in your project management tool. In this application, tracking items such checkpoints or quality gate decisions should be available. For this purpose Atlassian Confluence wiki-like tool is a good option to be considered.

- **Misjudging the size of a requirements gap**
  Organizations often assume the functional gap between their current state and the SAP solution is minor, only to discover deep incompatibilities later. This underestimation leads to rework, delays, and budget overruns. Proper scoping during the Discover and

Prepare phases of SAP Activate—using tools like the Business Process Master List (BPML) and Solution Scope Document—is vital to avoid such surprises. To assess the required level of functional gap you will need a good reporting tool capable of summarizing the results of the Fit/Gap Analysis.

- **Lack of alignment between IT and business stakeholders**
  When business leaders are disengaged or IT leads the project in isolation, the resulting scope may reflect technical feasibility but not business priorities. This gap creates friction during delivery and user adoption. SAP Activate encourages early organizational alignment and stakeholder engagement in the Prepare phase, with clearly defined roles and responsibilities in the project governance structure. This structure should be based on formally reviewed and approved Project Charter and Project Management Plan documents. The decisions recorded in these documents should guide the whole project

5.2 Case Study

Lidl: Lidl rejected SAP's inventory valuation method and heavily customized the system. After €500M+ spent, the solution was abandoned. SAP was not at fault—the strategic choice to preserve legacy logic was.

LeasePlan: The business model (leasing-as-a-service) did not align with SAP's monolithic architecture. Consolidating 35 systems into one proved unfit for a rapidly evolving, decentralized operation.

5.3 How to prevent problems

- **Begin with business process re-engineering**
  Before jumping into system design, companies must critically examine their existing processes to determine what should be preserved, redesigned, or retired. Using SAP Best Practices Explorer and industry model accelerators during the Discover phase can help re-imagine value streams rather than replicate inefficiencies.
  You can use tracking tools such as Atlassian Jira for creating lists of processes that that you intend to explore and review.

- **Use proof-of-concept to test fit before committing**
  Developing a pilot or sandbox system using SAP's Model Company or baseline configurations allows teams to visualize how standard SAP supports their needs. This testing, conducted during or before the Explore phase, helps expose gaps early and minimizes costly surprises later. This testing may be less formal with consultants demonstrating or more formal with specific Test Cases. If you want to go more formal than a tool for managing testing will be useful. One of the available solutions is XRAY for Jira application or testing module in CloudALM.

- **Challenge legacy process assumptions**
  Ask not "how can we replicate this in SAP?" but "why are we doing it this way at all?" During Fit-to-Standard workshops in the Explore phase, facilitators should encourage process owners to evaluate whether exceptions are justified. Many "must-have" legacy features are actually workaround artifacts from outdated systems. The results of these facilitation discussions should be recorded so that is not forgotten. You can do it in the forms of comments added to Scope Items tracked in Jira or CloudALM.

- **Build governance that includes both IT and functional leadership**
  Establish a steering committee and process governance teams composed of both IT leads and senior business stakeholders. Ensure decisions are vetted through both lenses—technical feasibility and business value. In SAP Activate, this is formalized in the Project Governance Framework, usually created during the Prepare phase. As part of the selected approach in the Project Governance, a toolset needs to be selected to support the whole project initiative. The most popular options are SAP Cloud ALM and Atlassian Jira.

6 Delivery Management

Delivery Management failures often occur when the implementation plan is overly ambitious, under-resourced, or outsourced with insufficient oversight. This is where strategy meets execution—and often breaks down.

6.1 Typical pitfalls

- **Overly aggressive rollout timelines**

  Tight delivery windows driven by board expectations or financial cycles often push teams to skip essential readiness activities such as integration testing, cutover rehearsals, or role-based training. This false sense of progress creates technical go-lives without operational readiness. In SAP Activate, the Realize phase should include sufficient time for test cycles and defect resolution — not just technical build. One of the ways Project Sponsors and Project Managers may protect the project success is to base their decisions not on opinions contained in PowerPoint slides but on data that can be regularly checked on Solution Readiness Dashboards. Make sure that your tool supporting the project has progress data that you can check without asking anybody for assistance.

- **Big bang deployments with minimal stabilization windows**

  Launching all modules and geographies at once, especially under tight timelines, leaves no room for progressive learning, phased training, or issue isolation. The result is chaos: overwhelmed support teams, cascading errors, and system paralysis. SAP Activate recommends Release Planning and phased deployments to reduce systemic shock—particularly for large enterprises or distributed teams. Make sure that your project governance strategy supplies the process for delivering the project in smaller increments like relases and sub-releases (called Waves in SAP Activate Methodology)

- **Lack of internal capability or dependency on vendors**

  Many clients rely heavily on system integrators without building internal knowledge or process ownership. This leads to low continuity, knowledge gaps, and delayed resolution when issues arise post-go-live. Without a strong internal PMO and business engagement, decisions are outsourced—often with limited context or accountability. This is probably one of the most critical pitfalls. Outsourcing everything to the services vendor puts you in a high risk of project failure. Make sure that you have access to live reports that show you the true progress reporting and do not rely on PowerPoint slides prepared by consultants just for the purpose of Gate Meetings.

- **Misaligned or inadequate resourcing across business and IT**
  Projects fail when critical roles are backfilled inadequately, when functional leaders split time between delivery and operations, or when IT has capacity but the business side does not. In SAP Activate's Prepare phase, the Project Staffing Plan should include full-time commitments from key business users, process owners, and test leads—not just IT. With properly configured deliverables tracking tool the demand for resources can be at least estimated based on estimated data and then displayed in one of the reports.

6.2 Case Study

SPAR: SPAR rushed the rollout of SAP S/4HANA without sufficient training or process readiness, leading to a regional supply chain collapse and over $100M in lost revenue. The problem wasn't the technology—the problem was how the technology was introduced into a "living" business structure.

Revlon: Integration of SAP post-acquisition of Elizabeth Arden resulted in $64M in lost orders and shareholder lawsuits due to chaotic delivery and system instability.

National Grid: Went live after Hurricane Sandy under financial pressure. Payroll and vendor processing collapsed, with $75M in settlement costs.

6.3 How to prevent problems

- **Design phased rollouts with clear stabilization criteria**
  Instead of a single go-live across all business areas, consider wave-based delivery. Each wave should include formal entry/exit criteria: stable system performance, trained users, and support coverage. SAP Activate allows this through structured release planning and staging configurations in deployment units that match business operations. Some tools like CloudALM or Jira offer integration with SAP systems so that decisions made in the tracking tool have their direct influence on configuration transfer from system to system.

- **Conduct readiness assessments across people, data, and processes**
  Use structured assessments (e.g. cutover rehearsal success, open defects, training

completion rates) before moving to next phases. In SAP Activate, this aligns with the Quality Gate Reviews before exiting Realize and again before go-live in the Deploy phase. A checklist-driven go/no-go decision, involving both IT and business, is essential. It is worth mentioning that Quality Gate Reviews can be based on data directly visible in the project management supporting tool. It is much more reliable than tables copied from Excel to PowerPoint, assuming that project team members actually track their work in this supporting tool.

- **Strengthen internal PMO capabilities before launch**
  Invest early in a Program Management Office that understands both SAP delivery and business transformation. The PMO should manage dependencies, enforce scope governance, and serve as the escalation point for unresolved risks. Activate provides the Project Governance Structure during Prepare to formalize this layer. Also, one of the critical decisions the PMO or Project Manger will make is selecting the project supporting tool, that is tailored to selected project governance model and delivery approach (Waterfall, Hybrid or Agile)

- **Limit dependencies on external integrators by building internal continuity roles**
  Create internal process owners, functional leads, and solution architects who work alongside consultants—not below them. SAP Activate encourages knowledge transfer planning starting in the Realize phase. Formalize this in workstream plans to reduce reliance on vendor continuity. What is also important is the value of frequent collaboration events suggested by SAP called "Show and Tell" when consultants are supposed to regularly demonstrated the results of their work. Such events are like micro-trainings and after a few sprints and sub-releases build up the necessary and solid knowledge about the business processes and the system.

7 Release Management

Release Management failures happen when go-live decisions are made for the wrong reasons—driven by budget deadlines, seasonal pressures, or executive optics rather than true operational readiness based on data and not on opinions.

7.1 Typical pitfalls

- **Go-live dates set based on financial calendars, not system stability**

  Many organizations choose their go-live date to align with the fiscal year, a major reporting event, or seasonal business cycles—without validating whether the system, users, or data are actually ready. This results in launching systems underprepared and not fully tested. In SAP Activate, project delivery should use readiness-based milestones, not fixed external dates, for the Deploy phase. If the system is not ready than the deployment should not take place.

- **No formal quality gates between phases**

  Skipping structured phase-end reviews means the project progresses without addressing unresolved issues, test gaps, or user concerns. This increases the likelihood of go-live with known risks. SAP Activate defines formal quality gate reviews (Q-Gates) at the end of each phase (e.g. Realize → Deploy), which must be completed with defined entry/exit criteria signed off by key stakeholders.

  These formal quality gates (Q-Gates) reviews should be described in the Project Governance Plan and scheduled upfront in calendars of executives. The data and reports about the solution and business readiness progress should come directly from the project management tool. CloudALM solution offers a pre-defined Solution Readiness dashboard reports. Conversely with Jira and some additional apps installed you can flexibly adapt your project reporting to specific needs.

- **Lack of rollback plan or cutover simulation**

  Without practicing the go-live transition in a real-time simulation (cutover rehearsal), teams don’t know and do not have clear sequence of actions during production launch. If things go wrong, they often have no way to safely reverse the deployment. In SAP Activate, the Cutover Plan is a mandatory activity in the Deploy phase, ensuring operational transitions are practiced and validated.

7.2 Case Study

Hershey: Hershey's go-live in Q3 1999 aimed to hit the Halloween candy season. The system failed, leaving $100M in unfulfilled orders and a stock price drop.

National Grid: Under pressure to avoid a $50M cost overrun, National Grid launched a new ERP system days after Hurricane Sandy, resulting in catastrophic financial processing issues.

7.3 How to prevent problems

- **Introduce go/no-go criteria based on functional readiness**, **use calendar as guideline**
  Develop a checklist of readiness indicators across functional, technical, and business dimensions. These may include open critical defects, data load validation, training coverage, and cutover rehearsal results. SAP Activate recommends defining these criteria in the Deploy phase plan and formalizing decisions through a cross-functional go-live committee.

- **Use mock cutovers and dry runs to validate execution plans**
  Conduct at least one full cutover simulation using production-like data and real teams. Track timing, data availability, task ownership, and incident handling during the dry run. SAP Activate's Cutover Checklist, used during Deploy, helps ensure technical and business cutover tasks are coordinated and executable under time constraints.

- **Avoid major go-lives during peak business periods**
  Plan go-lives during operational low seasons and not fiscal closings, seasonal peaks, or regulatory deadlines. This ensures resources are available and business impact is minimized. SAP Activate recommends integrating go-live planning with the customer's broader business calendar and risk profile, reviewed in Steering Committee meetings.

- **Include contingency plans and rollback options in every release plan**
  The project can have major and minor releases. Every major release with the go-live deployment to production system should include a documented fallback procedure with clear criteria for triggering rollback. This could include rolling back transports,

reverting interfaces, or activating business continuity processes. In Activate, this is addressed through the Deployment Strategy and Risk Mitigation Plan.

8 Status Reporting

Failures in this area occur when decision-makers lack an accurate, timely, or complete picture of project health based on live-data and not on opinions or declarations. Poor status reporting as part of the Solution Readiness control masks problems until they become unmanageable and risks become imminent.

8.1 Typical pitfalls

- **Overly optimistic reporting from consulting partner or PMs**
  Employees from the Consulting Partner and/or internal Project Manager may avoid fully disclosing issues to maintain stakeholder trust or stay aligned with delivery schedule ignoring quality standards and risks monitoring. When reports reflect progress based on timelines and milestones alone —without factoring in testing results, defect trends, end-user feedback, or operational readiness—leaders are misled into thinking the program is on track. In SAP Activate, this undermines the Solution Readiness Review that is expected during phase transitions, especially Realize → Deploy.

- **Suppressed red flags or whistleblower warnings**
  Delivery teams or functional leads may raise early warnings—but these are sometimes filtered, ignored, or politically suppressed. Also, when the team atmosphere does not encourage openess then the warnings may not surface in time. When legitimate concerns don't reach executive ears, the organization misses its opportunity to correct the situation. The SPAR case illustrates the cost of disregarding early alerts. SAP Activate recommends establishing issue and risk escalation protocols within the project governance model in the Prepare phase.

- **KPIs focused on effort or cost, not business readiness or value**
  Projects often measure what’s easy: percent complete, budget burn, story points delivered. But these say little about whether the business is actually ready to run on the

new solution - the forgotten human factor. SAP Activate encourages a blend of technical and business KPIs, including training completion, data quality benchmarks, UAT satisfaction, and open issues or defects by severity—especially as part of Solution Readiness Assessments.

- **Disconnect between Solution Readiness dashboards and on-the-ground realities** Central PMO reports often miss real delivery issues—such as regional teams not following the process, or end-users skipping training. This "reporting illusion" results in Project Sponsors being blindsided. SAP Activate recommends embedding embedded quality assurance (QA) roles and independent status verification mechanisms to avoid this problem.

8.2 Case Study

SPAR: An internal whistleblower warned executives of the risk in 2021—over a year before the failed SAP rollout in KZN. The alert was ignored by three board members.

LeasePlan: Executives were pitching ERP-driven efficiency during company roadshows while the system was still struggling in feasibility testing. The disconnect undermined credibility.

8.3 How to prevent problems

- **Include data-driven business readiness KPIs in Project Status Reports**
  Go beyond technical metrics by tracking also organizational and process indicators: training coverage, super-user feedback, cutover rehearsal success, master data load quality, and user acceptance. These indicators are part of the Solution Readiness Dashboard in SAP Activate's project life cycle. This dashboard (one or many) can be constructed Jira with additional apps that offer visually attractive reporting capabilities. Also CloudALM offers such dashboard. However, unlike in Jira, the customization of this dashboard is not possible.

- **Establish a risk review board independent of delivery teams**
  Set up a small group of cross-functional leaders (including business, IT, and QA) tasked with reviewing unresolved risks weekly. This Risk & Readiness Review Board should

operate outside the delivery chain of command and report directly to project sponsors. SAP Activate encourages this via the project governance charter and steering committee practices defined in phase Prepare. Also here functionalities of Jira (risk register, risk review dashboard) can support the meeting effectiveness. Jira (risk register, risk review dashboard) can support the meeting effectiveness.

- **Encourage escalation through structured governance paths**
  Make it easy and safe for concerns to be escalated—formally. Use issue log register that can easily be configured in Jira, anonymous risk reporting channels, and routine check-ins where risks are reviewed openly. SAP Activate suggests using the Issue and Risk Management Plan, including ownership, thresholds, and escalation paths.

- **Review reporting cadence and accuracy regularly with Executive Sponsors**
  Dashboards and reports should be actively reviewed by sponsors and challenged when needed— not just read and accepted. Schedule periodic “deep dives” into key metrics with delivery leads, test leads, PMO, and selected business users. SAP Activate supports this through executive stakeholder engagement and recurring status checkpoint meetings during each phase transition. Such meetings should be facilitated using the project management toolset that offers real data from all involved parties.

9 Test Management

Test Management failures undermine the very confidence that makes go-live possible. A well-configured SAP system can still fail if its processes, data, and integrations are not validated in realistic and multiple business scenarios. Testing must confirm not just system correctness—but operational fitness. The absence of a structured, business-driven test strategy is a leading indicator of the post-go-live instability.

9.1 Typical pitfalls

- **Incomplete or biased test scripts (e.g. only confirming success)**
  Many teams design test scripts that only validate the “happy path”—ideal, error-free scenarios. However, real-world operations include negative cases, user mistakes,

interface failures, and unexpected data combinations. When these are excluded, testing creates false confidence. SAP Activate's Test Planning and Execution workstream during the Realize phase emphasizes the critical role of well prepared test management strategy, test cases and testing cycles organization.

- **Untrained or unengaged testers**
  Testers are often pulled in late, have limited understanding of the new solution, or are assigned without clear roles. This results in superficial testing ("just-do-it"), missed defects, and poor documentation of test outcomes. In SAP Activate, key users and business SMEs should be involved from the Fit-to-Standard workshops onward so they can act as credible, engaged testers during User Acceptance Testing (UAT). The most practical way is involving key users regularly during the delivery in so called "Touch and Feel" sessions when they can learn the system as it is being built.

- **Not cleaned, missing, or duplicated data during test cycles**
  Test environments that lack clean, representative master and transactional data make it impossible to validate business processes accurately. This is particularly dangerous for SAP implementations where downstream processes (e.g. inventory valuation, invoicing, reporting) depend on valid inputs. SAP Activate includes Data Migration Validation and Mock Data Loads as part of the Realize phase, which should be completed before testing begins. The status of data cleansing objects and data loads objects should be closely monitored in project management toolset. In Jira we can do it using specific Data Objects register.

- **No performance or volume testing before launch**
  Functional correctness does not guarantee performance under load. Systems that work for 10 users in a test lab may fail with 1,000 concurrent users post go-live. When performance testing is skipped or limited to technical teams, issues emerge too late. SAP Activate prescribes Load and Stress Testing in the Realize phase, and this should be executed using production-sized data volumes and usage patterns. This aspect is less valid, however, when you go with the SAP Cloud solution

9.2 Case Study

Target Canada: Data entry errors corrupted 70% of inventory records. Test cycles did not surface issues until stores launched with empty shelves.

SPAR: Data inconsistencies between SAP and legacy systems led to under- and overstocking, delayed orders, and inventory chaos.

9.3 How to prevent problems

- **Treat testing as a business activity, not just a technical one**
  Testing is not just about system logic—it's about end-to-end operational readiness. Business process owners must review test cases, validate expected results, and sign off on business acceptability. In SAP Activate, Business Process Testing should be part of the solution validation strategy, with SMEs leading UAT and signing functional readiness. Testing is a complex process on its own but with specialized tools like XRAY for Jira, we you can gain the desired transparency during testing in the Realization Phase.

- **Use real business data for validation**
  Test environments should be populated with anonymized but production-like master and transactional data. This ensures validation of pricing, tax, logistics, financial postings, and approvals across integrated modules. SAP Activate includes Test Data Preparation and Management as a key activity in Solution Validation.

- **Test also failure scenarios (not just "happy paths")**
  Create test cases that deliberately simulate wrong data, invalid sequences, and system unavailability. Run end-to-end exception paths like "invoice mismatch," "over-delivery," or "batch failure." These negative tests help identify whether users and systems respond appropriately to errors.

- **Allocate time and budget for multiple test cycles, including UAT and regression**
  Plan for multiple iterative test cycles, not just one round of UAT. Include regression tests after configuration changes and defect fixes. Ensure capacity for retesting known

defects. In SAP Activate, this is part of the Test Plan and Test Execution Schedule, developed in the Realize phase.

10 Documentation

Poor documentation creates downstream confusion, knowledge silos, and dependency on consultants. It’s the hidden failure that haunts post-go-live operations.

10.1 Typical pitfalls

- **No living process documentation during build**
  Documentation often begins too late—or worse, is treated as a final deliverable instead of an ongoing activity. As configuration evolves during the Realize phase, decisions are made rapidly, and if they aren't recorded in real time, the final documentation becomes outdated or inaccurate. SAP Activate recommends that process flows, configurations, and integration points be documented continuously as part of the Solution Documentation workstream.

- **Lack of decision logs and change history**
  Without a traceable record of key design decisions, scope changes, or process deviations, organizations struggle to understand *why* something was built a certain way. This weakens governance, complicates support, and often results in costly rework. SAP Activate includes a Change Log and Decision Log template that should be maintained from Explore through Deploy phases. Maintaining such logs can perceived as an administrative burden. That is why it is better to keep it in the context of the work that is being delivered. Jira offeres the commenting and labeling functionality that allow for keeping track of decisions and changes right in the object being delivered - Deliverable or Feature.

- **Training materials developed too late (or not at all)**
  Training is often deprioritized in favor of configuration or testing. When training materials are rushed or based on outdated configurations, end-users enter go-live unprepared, fueling adoption resistance. SAP Activate’s Organizational Change

Management stream requires that training needs analysis and content development begin in Realize, not left to the final weeks of deployment. The Hybrid delivery approach may help in such a way that some user manuals for core functionalities can be produced earlier as they will be not changed in next iterations.

- **Inconsistent handover from implementation partners to internal teams**
  Projects frequently conclude with limited knowledge transfer. If implementation partners hold key documentation—or fail to deliver a structured handover—internal teams remain dependent and unempowered. This creates long-term vendor lock-in and support delays. SAP Activate specifies a Transition to Operations package, which includes technical documentation, test evidence, admin guides, and support procedures.

10.2 Case Study

Lidl: As processes evolved and SAP was customized beyond recognition, core documentation was no longer relevant. New staff couldn’t onboard or support the platform.

National Grid: Vendor handoffs were incomplete. Knowledge gaps prolonged the stabilization period and increased the cost of recovery.

10.3 How to prevent problems

- **Embed documentation tasks into every sprint or milestone**
  Treat documentation as a deliverable within each work package—not a final phase. Include process diagrams, configuration rationales, integration specs, and testing notes. In Agile SAP Projects, assign documentation responsibilities within each scrum team and review progress during sprint retrospectives. You can use any tracker for progress monitoring but in Jira or the CloudALM, the work item can be directly linked with the documentation that relates to the work item’s outcome.

- **Require a formal handover package from implementation partners**
  Before contract closure, require partners to submit a full Knowledge Transfer and Operational Readiness Kit. This should include admin guides, technical documentation,

test results, and unresolved issues. Validate delivery through internal QA or PMO reviews during the Deploy phase.

- **Invest in structured training and self-service support content**
  Develop role-specific training materials, process simulations, and quick reference guides that align to real system behavior. SAP Activate recommends building a training catalog linked to business roles and publishing it through company's Learning Management System or simply on internal Confluence platform.

- **Maintain a centralized knowledge repository for design, configuration, and decisions**
  Use a common workspace (e.g. SharePoint, Confluence, or SAP Solution Manager/ALM) to store and organize all critical documentation. Access should be structured by role, lifecycle phase, and system component. SAP Activate recommends building this repository starting in Prepare, then continuously populating it through each phase with inputs from workstream leads.

11 Patterns and root causes: the failure matrix

Across the 12 SAP Project failures analyzed, several recurring themes emerged. These can be visualized in a Failure Matrix, mapping each case to one or more of the six Project Governance Areas. The matrix shows the density of breakdowns and reveals the systemic nature of these problems.

11.1 Top 10 root causes of failure

1. Legacy process overprotection
2. Over-customization instead of standardization
3. Overreliance on implementation partners
4. Poor executive alignment and sponsorship
5. Rushed go-lives driven by calendar, not by business readiness
6. Underinvestment in organizational change management and training

7. Not cleansed, incomplete, or misaligned data
8. Lack of go/no-go criteria and rollback planning
9. Disconnected status reporting and KPI misalignment
10. No documentation discipline or knowledge transfer plan

11.2 Typical causes of failure by Project Governance Area

Project Governance Area	Typical causes of failure
Scope Planning	• Refusing to standardize processes • Attempting to customize SAP into legacy shapes • Misjudging the size of a requirements gap • Lack of alignment between IT and business stakeholders
Delivery Management	• Refusing to standardize processes • Attempting to customize SAP into legacy shapes • Misjudging the size of a requirements gap • Lack of alignment between IT and business stakeholders
Release Management	• Refusing to standardize processes • Attempting to customize SAP into legacy shapes • Misjudging the size of a requirements gap • Lack of alignment between IT and business stakeholders
Status Reporting	• Refusing to standardize processes • Attempting to customize SAP into legacy shapes • Misjudging the size of a requirements gap • Lack of alignment between IT and business stakeholders
Test Management	• Incomplete or biased test scripts • Untrained or uninterested testers • Low quality of data during test cycles • No performance or volume testing before launch

Project Governance Area	Typical causes of failure
Documentation	• Refusing to standardize processes • Attempting to customize SAP into legacy shapes • Misjudging the size of a requirements gap • Lack of alignment between IT and business stakeholders

12 How to prevent failure - Summary

12.1 Become a "Responsible Buyer" — don't outsource your ownership

Organizations often assign delivery accountability to implementation partners or solution vendors. This is the root of many failures. SAP Project success demands internal ownership.

What can be done?

- Appoint a full-time internal SAP Project Lead with both delivery experience and business process fluency as well as strong internal relations with key stakeholders already established.
- Create a responsibility matrix where key responsibilities (design decisions, testing sign-off, readiness validation, business readiness) are owned by internal stakeholders, not vendors.
- Control the roadmap using a business capability model, not a vendor work plan.
- Review and approve backlog items and configuration changes using own Project Governance standards.

12.2 Treat Organizational Change Management seriously

Technology is only half the project**. Behavioral change is the other half**—and it's often ignored until go-live panic sets in.

What can be done?

- Start the Organizational Change Management (OCM) track in the Prepare phase of SAP Activate.
- Run a Stakeholder Impact Assessment for each business function—what's changing, why, and how to support it after the go-live.
- Build a Training Curriculum by Business Role, not by system module. Use real business scenarios that are business goal-oriented. Teaching just technical navigation is not enough.
- Track adoption readiness KPIs like end-user confidence scores, check end-user training completion and test their knowledge.

12.3 Use Agile controls to stay aligned and nimble

SAP Activate supports an agile delivery model. It is not pure SCRUM approach but more of a hybrid with some SCRUM practices during the Realization and Deploy phases. Unfortunately many projects still default to waterfall in practice.

What can be done?

- Plan your project using time windows with specific start and end dates. Try to fit the scope to be delivered to the time available. Use Gantt charts only for high level planning but not on the task level scheduling.
- Organize retrospectives in teams after each completed Sprint. Discuss what worked and what can be done differently during the next Sprint.
- After each deployment Wave or Release, hold a project level retrospective about the delivery process with both IT and business leads. Discuss delivery process variations, process gaps, and required changes to the delivery approach. Follow Activate Methodology when it helps the purpose of your project but do not follow it rigorously like religion. Use the fit-for-purpose approach.

- Before each go-live (regional or functional), hold a Stakeholder Readiness Review including key-users, process owners, and data migration leads. Only proceed when this group jointly confirms readiness.

12.4 Define emergency state criteria

Most SAP failures show signs long before they collapse—but no one wants to pull the brake and stop the delivery. A mature program has control check points, quality control gates and exit paths.

What can be done?

- Set clear kill or pause criteria: e.g., critical defects unresolved after two test cycles, failure in two mock cutovers, organizational business readiness blocked or executive misalignment.
- Use independent (external) SAP QA reviewers to challenge status reports. Their role is to look beyond dashboards and interview actual delivery teams.
- Schedule regular check point meetings (based on release and deployment calendar) where status is reviewed with business context, risks are prioritized, and decisions are made — not just noted.

13 Conclusion: Transformation Requires Ownership

SAP is not just a system—it becomes the backbone of your business operations. When it fails, business stops. Orders don't ship. Invoices aren't being generated. Salaries don't get paid. The cost of failure isn't just technical—it's existential for your business.

An SAP Project isn't "just IT." It fundamentally changes how your company operates.

- Departments that used to work in isolation must now align.
- Manual workarounds must give way to structured flows.
- Decisions become more traceable—and more exposed.

That's why SAP transformation is not a tech project. It's organizational surgery.

13.1 The truths that must be accepted early

- SAP methodology (like Activate) and supporting tools (like Jira or CloudALM) do not prevent failure—they give you a way to detect it early. But only if you use it properly.
- Governance is not a formality. It must be designed to create visibility, invite collaboration, and escalate truth—not to report green statuses.
- If your team is afraid to say “we’re not ready,” you’ve already lost control of the project.

13.2 Your role as Leadership

To lead SAP transformation, you must build a culture of transparency:

- Make it safe to raise concerns without blame.
- Introduce the language of risk and readiness—not just scope and dates.
- Reward clarity over optimism.

This requires from the PMO governance that is active, not ceremonial. Ask hard questions. Visit the testing floor. Listen to key-users. Look beyond dashboards.

If you’ve made it this far, congratulations—that’s a huge step forward, putting you ahead of many SAP Projects that don’t succeed. Now, use these lessons as your guide and take action before the challenges escalate. Your success starts here!

14 Appendix 1: R2D ALM for Jira

Over the course of this paper, we've examined how SAP implementation failures often arise not from technology limitations, but from a lack of structure, alignment, and visibility across key delivery areas.

While the SAP Activate methodology offers a strong foundation, many organizations continue to struggle with operationalizing it in a way that is transparent, flexible, and collaborative.

This is where R2D ALM for Jira enters the conversation.

A familiar problem, a modern approach

SAP's own Requirements-to-Deploy (R2D) process — popularized through tools like Focused Build for Solution Manager — is intended to structure how requirements move through design, build, test, and deployment. But in practice, many organizations find these SAP-native tools:

- Rigid and hard to adapt to hybrid agile-like delivery models
- Lacking user-friendly interfaces that support cross-functional collaboration
- Overly complex for business and functional teams who are not SAP technical experts
- Disconnected from the modern toolchains already used by agile teams (Jira, Confluence, etc.)

As a result, even organizations with strong governance intentions fall back into manual tracking in Excel, fragmented reporting, and low engagement from stakeholders outside of the core IT or SAP teams.

R2D ALM for Jira is a structured, pre-configured implementation of the R2D process — delivered entirely within Jira Cloud and Confluence Cloud — that addresses these challenges head-on.

Built for SAP Projects. Delivered with Atlassian tools.

R2D ALM for Jira doesn’t replace SAP Activate or R2D process known from Focus Build — it brings them to life in a way that is:

- Easy to use
- Seamlessly integrated
- Fully transparent
- Aligned with governance best practices

The solution includes:

- **Pre-configured Jira Cloud configuration** supporting all R2D work item types: Requirements, Work Packages, Work Items, Test Cases, Defects, Releases, etc.
- **Confluence space example** with embedded documentation structure
- **Xray test management integration** to support structured test case management, planning, and execution
- **Automations** accelerating workflows
- **Dashboards for stakeholders**: PMO, Test Manager, Release Lead, Solution Owner
- **Role-based training guides** and live training sessions
- **Optional read-only demo access** to evaluate the solution before purchase

Designed to prevent what typically goes wrong

The solution was not designed in a vacuum — it was shaped specifically to prevent the types of failures outlined in this paper.

R2D Area	How R2D ALM for Jira Helps
Scope Planning	Standardizes requirement types and traceability; avoids uncontrolled scope changes
Delivery Management	Visual boards and dashboards highlight tasks to be delivered and overdue items as well as capacity issues
Release Management	Built-in release calendars and deployment tracking
Status Reporting	Real-time dashboards with business readiness indicators and risk
Test Management	Xray app integration supports test cycles, coverage, and reporting

Documentation	Pre-built Confluence structure supports collaborative editing and traceability

Who will benefit from this solution?

R2D ALM for Jira is intended for:

- **CIOs and program directors** preparing for S/4HANA transitions
- **PMOs and transformation leads** who want better oversight without SAP Solution Manager
- **SAP CoEs** modernizing their internal tool landscape

Not just a tool – a way of working

The intention behind R2D ALM for Jira is not to introduce another tool — it is to **enable SAP programs to adopt better ways of working**, using tools that stakeholders already understand and enjoy using. This is especially critical in organizations that span agile, hybrid, and waterfall teams, where a single, rigid methodology may not fit all streams.

The solution reinforces the mindset promoted throughout this white paper:

- Clarity over complexity
- Ownership over outsourcing
- Transparency over comfort
- Readiness over deadlines

Learn More

If the pain points in this white paper resonate with your current situation, and you are exploring available toolsets and alternative ways to structure, govern, and deliver your SAP initiatives: visit https://pmbg.eu and find out more about R2D ALM for Jira

A brief discovery workshop and a two-week sandbox demo can help your team assess whether this solution fits your needs — before you commit.

15 Appendix 2: Source Articles

SAP ERP Project Failure Lessons Learned and Mini Case Studies 1

- https://www.iitrun.com/sap-erp-project-failure-lessons-learned-and-mini-case-studies-1/

Shane's Blame Game: Management, Not SAP Retail, Sinks Jewelry Company

- https://ematters.wordpress.com/2009/01/14/shanes-blame-game/
- https://web.archive.org/web/20110811171915/https://www.erpko.com/articles/erp-articles/shane-company-lessons-learned-from-an-erp-implementation-failure/

GovERP debacle: A predictable but valuable lesson

- https://www.innovationaus.com/goverp-debacle-a-predictable-but-valuable-lesson/

12 famous ERP disasters, dustups and disappointments

- https://www.cio.com/article/278677/enterprise-resource-planning-10-famous-erp-disasters-dustups-and-disappointments.html

10 Early signs of ERP disaster

- https://www.cio.com/article/230492/10-early-warning-signs-of-erp-disaster.html

10 common ERP mistakes and how to avoid them

- https://www.cio.com/article/284312/10-common-erp-mistakes-to-avoid.html

11 tips for deploying ERP application

- https://www.cio.com/article/289076/enterprise-software-11-tips-for-deploying-erp-applications.html

5 reasons why SAP Projects fail

- https://onfinity.io/blog/technologies/5-reasons-why-sap-projects-fail/

Press Release where they need to admit higher than expected costs and complexities relating to SAP implementation

- https://www.sec.gov/Archives/edgar/data/804212/000119312510035790/dex99a1.htm

Anatomy of a successful SAP Implementation

- https://www.cio.com/article/246425/anatomy-of-a-successful-sap-implementation.html

5 Most popular ERP failures

- https://goldenowl.asia/blog/erp-system-implementation-failures

Cost of troubled SAP Project

- Cost of troubled SAP project will skyrocket to nearly $1 billion, audit says – Computerworld

7 tips to hire the right SAP consultant

- https://community.sap.com/t5/enterprise-resource-planning-blog-posts-by-members/seven-tips-to-ensure-you-hire-the-right-consultant/ba-p/12897229

How SAP niearly broke SPAR

- https://currencynews.co.za/how-sap-nearly-broke-spar/

8 Main reasons SAP Projects Fail

- https://targpoint.com.br/en/reasons-why-sap-projects-fail/

Case Study 12: Lidl's €500 Million SAP Debacle

- https://www.henricodolfing.com/2020/05/case-study-lidl-sap-debacle.html

Scope Creep - Maintaining SAP Project Control

- Scope Creep: Maintaining SAP Project Control | LinkedIn

Inside the Massive $100M SAP S/4HANA Failure

- Inside the Massive $100M SAP S/4HANA Failure at Spar Group | LinkedIn

Project Failure Case Study - Leaseplan

- https://www.henricodolfing.com/2020/01/project-failure-case-study-leaseplan-sap.html

Failed SAP implementation costs LeasePlan €100 million

- https://www.consultancy.uk/news/22850/failed-sap-implementation-costs-leaseplan-100-million

Lidl's $600 Million SAP Disaster: What Went Wrong and What Every Business Can Learn

- https://www.thirdstage-consulting.com/lidls-600-million-sap-disaster/