Legacy Migration Checklist for Architects

Business Value & Alignment

□ Have we clearly articulated the business goals for this migration beyond technical improvements?

□ Have we identified specific customer/user pain points this migration will address?

□ Have we quantified potential time/cost savings for the business and customers?

□ Have we identified new business capabilities that will be enabled by the migration?

□ Do we have metrics in place to measure business value before and after the migration?

□ Have we secured executive sponsorship with alignment on business objectives?

□ Do we understand how this migration supports the company's long-term strategy?

□ Have we aligned with different stakeholders on their specific migration objectives (revenue generation, operational efficiency, risk reduction)?

□ Have we identified product lines or features that are unprofitable and could be discontinued rather than migrated?

Stakeholder Engagement

□ Have we identified all relevant stakeholder groups (end users, business departments, IT, etc.)?

□ Have we conducted interviews or observation sessions with frontline employees?

□ Have we gathered input from end customers on their needs and pain points?

□ Have we analyzed support tickets and common user complaints?

□ Have we engaged with sales/customer service teams to understand customer expectations?

□ Do we have a communication plan to keep stakeholders informed throughout the migration?

□ Have we established feedback channels for continuous stakeholder input?

□ Have we engaged with finance teams to understand revenue impacts of different products/features?

□ Have we identified off-system workarounds (spreadsheets, access databases, etc.) that have evolved around legacy limitations?

□ Have we involved stakeholders in identifying meaningful ways to slice the migration?

Work with business experts to break down the system into logical segments that can be migrated independently, focusing on business value rather than technical boundaries.

Technical Assessment

□ Have we thoroughly analyzed the current system architecture and dependencies?

□ Do we understand the data models and their business relevance?

□ Have we identified technical debt that should be addressed during migration?

□ Do we know which legacy features are still used vs. obsolete?

□ Have we evaluated security risks in both the legacy and target systems?

□ Do we understand integration points with other systems and their migration impacts?

□ Have we identified cross-cutting concerns (logging, security, etc.) that need redesign?

□ Have we traced data flows to original sources rather than assuming the legacy system is the source of truth?

Often better data exists at original source systems that was lost or degraded when passed to legacy systems.

□ Have we identified any critical aggregators (reporting functions crucial to running the business)?

Reports or data aggregation processes that executives rely on to run the business, which often become bottlenecks in migrations.

□ Have we analyzed how current business processes are shaped by legacy system constraints?

□ Have we explored potential event interception points (messaging, APIs, databases) to enable incremental migration?

Identify locations where you can intercept data flows between systems to gradually redirect processing to new components.

Migration Strategy

□ Have we designed a phased approach that reduces risk?

□ Have we identified high-value components to prioritize for early migration?

□ Do we have a data migration strategy that ensures business continuity?

□ Have we planned for parallel operations during transition if needed?

□ Have we defined rollback procedures in case of migration issues?

□ Do we have a plan for handling legacy system maintenance during migration?

□ Have we established a testing strategy for verifying functionality post-migration?

□ Have we explored applying the Strangler Pattern to gradually replace functionality?

A pattern where new functionality gradually takes over from legacy code by intercepting calls and redirecting them.

□ Have we considered tackling critical aggregators (essential reports/functions) early rather than last?

Consider replacing critical reports first rather than leaving them until the end where they can block complete migration.

□ Have we designed necessary transitional architecture components with clear plans for their eventual removal?

Temporary components needed during migration that should be removed once they're no longer needed.

□ Have we avoided the feature parity trap by focusing on business needs rather than replicating all existing functionality?

Resist the temptation to simply recreate the existing system with newer technology.

□ Have we explicitly ruled out big bang approaches in favor of incremental migration?

□ Have we explored different slicing approaches?

Consider options like migrating by product line, user group, business capability, or user journey to find the most effective approach.

□ Have we implemented frequent delivery practices from the beginning of the migration to validate future delivery capabilities?

Human Factors & Change Management

□ Have we identified potential resistance points among user groups?

□ Do we have a plan to address fears about job security or role changes?

□ Have we accounted for training needs for different user groups?

□ Do we understand how daily workflows will change and how to support that transition?

□ Have we communicated the benefits of the new system to all affected parties?

□ Do we have champions in each business unit to help promote the change?

□ Have we considered how to manage workload during the transition period?

□ Have we prepared users and stakeholders for potential temporary disparities in user experience during phased migration?

During incremental migration, users may experience different interfaces when moving between old and new system components.

□ Have we addressed the organizational behaviors that led to the legacy situation in the first place?

Consider what patterns of decision-making or organizational culture contributed to the legacy situation and how to change them.

Resource Planning

□ Do we have the right mix of technical skills for both legacy and target technologies?

□ Have we allocated product management resources to guide the migration?

□ Have we budgeted for potential unforeseen technical challenges?

□ Do we have contingency plans for timeline extensions if needed?

□ Have we accurately estimated the total cost of ownership for the new system?

□ Do we have access to subject matter experts for critical legacy components?

□ Have we considered external expertise needs for specialized migration tasks?

□ Have we allocated resources for implementing and eventually removing transitional architecture components?

□ Have we accounted for the time needed to collaborate with business on identifying migration slices?

Ensure the schedule includes dedicated time for workshops to analyze and define meaningful migration increments.

□ Have we budgeted for the potential parallel running of critical systems during transition phases?

Delivery Approach

□ Are we "building as we mean to continue" with the same practices we want post-migration?

If the goal is to release every two weeks post-migration, start releasing every two weeks during migration.

□ Have we established short feedback loops for early validation of migration approaches?

□ Are we regularly demonstrating business value through incremental delivery?

□ Have we implemented frequent releases to prove our ability to deliver quickly post-migration?

□ Have we designed test automation that supports rapid, confident changes?

□ Do we have a clear ownership model for components during transition and after migration?

□ Have we avoided heavyweight change processes that contradict our future delivery goals?

Don’t use bureaucratic change management processes during migration if you aim for agility afterward.

System Consolidation (if applicable)

□ Have we mapped feature parity requirements between the systems being consolidated?

□ Do we understand the different user experiences and expectations for each system?

□ Have we identified potential conflicts in business processes between systems?

□ Do we have a strategy for data reconciliation between disparate systems?

□ Have we established decision-making criteria for resolving conflicting requirements?

□ Do we understand the organizational impacts of merging user communities?

□ Have we documented terminology differences to ensure consistent understanding?

□ Have we re-evaluated the assumption that all systems need to be consolidated rather than some retired?

Post-Migration Planning

□ Do we have a maintenance and support plan for the new system?

□ Have we defined how to measure the success of the migration?

□ Do we have plans to collect user feedback after implementation?

□ Have we established a process for addressing issues and enhancements post-launch?

□ Do we have knowledge transfer plans to operational teams?

□ Have we documented architectural decisions for future reference?

□ Have we scheduled a retrospective to capture lessons learned?

□ Have we established processes to ensure we don’t accumulate technical debt in the new system?

□ Have we implemented continuous delivery practices that were promised in the business case?

□ Have we validated that the promised benefits (faster time to market, lower cost of change) are actually realized?

□ Do we have a plan to eventually decommission any remaining legacy components?