Legacy Migration Checklist v3.1

Info

System Name: __

Team: __

Date: __

How to Use This Checklist

This checklist is designed to work with the Legacy Migration Canvas during a 4-8 hour workshop. Questions are prioritized to help you focus on what matters most:

- **P1 (Very Important)**: Should be addressed in workshop - drives canvas completion
- **P2 (Important)**: Should address if time permits, or assign as follow-up tasks
- **P3 (Additional)**: Additional considerations that could be important - depends on the system

> **Facilitator Note:** Start with P1 questions to ensure canvas completion. Use P2/P3 to deepen analysis where time allows or create actionable follow-up tasks.

Business Case & Value Proposition

Why are we doing this & what do we gain?

- ☐ 1.1 **P1** Have we clearly articulated the business goals for this migration beyond technical improvements?
- ☐ 1.2 **P1** Have we identified specific customer/user pain points this migration will address? What are they?
- ☐ 1.3 **P1** Have we conducted a neutral assessment of the current system's strengths and weaknesses before defining a specific migration approach?

- [ ] 1.4 **P1** Do we understand how a potential migration could support the company's long-term strategy?
- [ ] 1.5 **P1** Have we identified new business capabilities that will be enabled by the migration?
- [ ] 1.6 **P1** Have we aligned with different stakeholders on their specific migration objectives (revenue generation, operational efficiency, risk reduction)?
- [ ] 1.7 **P2** Have we identified product lines or features that are unprofitable and could be discontinued rather than migrated?
- [ ] 1.8 **P2** Have we quantified potential time/cost savings for the business and customers?
- [ ] 1.9 **P2** Do we have metrics in place to measure business value before and after the migration?
- [ ] 1.10 **P2** Have we considered using structured goal-setting frameworks like OKRs to align migration objectives with measurable business outcomes?

OKRs (Objectives and Key Results) connect high-level business objectives with specific, measurable key results that can be tracked throughout the migration.

- [ ] 1.11 **P2** Have we secured executive sponsorship with alignment on business objectives?

Facilitator Note: Focus on tangible business value, not just "technical debt reduction." Help the team think like product owners, not just engineers.

Current System Analysis

What we're working with

- [ ] 2.1 **P1** Have we thoroughly analyzed and documented the current system architecture and dependencies?
- [ ] 2.2 **P1** Do we understand the data models and their business relevance?
- [ ] 2.3 **P1** Do we know which legacy features are still used vs. obsolete?
- [ ] 2.4 **P1** Have we identified cross-cutting concerns (logging, security, etc.) that need redesign?
- [ ] 2.5 **P2** Do we understand integration points with other systems and their migration

impacts?

- ☐ 2.6 **P2** Have we analyzed current operations, monitoring, and support processes for the legacy system?
- ☐ 2.7 **P2** Have we identified technical debt that should be addressed during migration?
- ☐ 2.8 **P2** Have we evaluated security risks in both the legacy and target systems?
- ☐ 2.9 **P2** 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.

- ☐ 2.10 **P2** 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.

- ☐ 2.11 **P3** Have we analyzed how current business processes are shaped by legacy system constraints?
- ☐ 2.12 **P3** 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.

> **Facilitator Note:** Don't get lost too deep into technical details in the workshop. Focus on understanding the big picture and identifying what needs deeper analysis.

Stakeholder Needs

Who's affected & what they actually need

- ☐ 3.1 **P1** Have we identified all relevant stakeholder groups (end users, business departments, IT, etc.)?

- [ ] 3.2 **P1** Have we gathered input from end customers on their needs and pain points?
- [ ] 3.3 **P1** Have we engaged with sales/customer service teams to understand customer expectations?
- [ ] 3.4 **P1** Do we have a communication plan to keep stakeholders informed throughout the migration?
- [ ] 3.5 **P1** Have we established feedback channels for continuous stakeholder input?
- [ ] 3.6 **P2** Have we conducted interviews or observation sessions with frontline employees?
- [ ] 3.7 **P2** Have we analyzed support tickets and common user complaints?
- [ ] 3.8 **P2** Have we engaged with finance teams to understand revenue impacts of different products/features?
- [ ] 3.9 **P2** Have we defined communication cadence and key milestone updates for different stakeholder groups throughout the migration?

Different stakeholders need different update frequencies - executives might need monthly updates while end users need more frequent communication during their migration phases.

- [ ] 3.10 **P2** Have we established escalation protocols for communicating migration issues or delays to stakeholders?

Define when and how to communicate problems, who needs to be informed first, and what level of detail different groups require.

- [ ] 3.11 **P3** Have we identified off-system workarounds (spreadsheets, access databases, etc.) that have evolved around legacy limitations?
- [ ] 3.12 **P3** 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.

- [ ] 3.13 **P2** Do we have a plan for celebrating migration milestones and communicating wins to maintain stakeholder engagement?

Migration projects can be long - plan for recognizing progress and maintaining momentum through regular success communication.

Facilitator Note: Technical teams often underestimate stakeholder complexity. Help them understand that "users" includes many different groups with different needs.

Organizational Constraints

Reality check

- ☐ 4.1 **P1** Do we have the right mix of technical skills for both legacy and target technologies?
- ☐ 4.2 **P1** Have we allocated product management resources to guide the migration?
- ☐ 4.3 **P1** Have we budgeted for potential unforeseen technical challenges?
- ☐ 4.4 **P1** Do we have access to subject matter experts for critical legacy components?
- ☐ 4.5 **P2** Do we have contingency plans for timeline extensions if needed?
- ☐ 4.6 **P2** Have we accurately estimated the total cost of ownership for the new system?
- ☐ 4.7 **P2** Have we considered external expertise needs for specialized migration tasks?
- ☐ 4.8 **P2** Have we budgeted for the potential parallel running of critical systems during transition phases?
- ☐ 4.9 **P3** Have we allocated resources for implementing and eventually removing transitional architecture components?
- ☐ 4.10 **P3** 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.

Facilitator Note: Be realistic about constraints. Technical teams tend to be optimistic about timelines and underestimate the complexity of organizational coordination.

Quality Goals

What matters most

- ☐ 5.1 **P1** Have we identified the top 3-5 most important quality goals for this migration?
- ☐ 5.2 **P1** Have we aligned quality goals with key stakeholder concerns and business drivers?
- ☐ 5.3 **P1** Have we established measurable criteria for each quality goal?
- ☐ 5.4 **P1** Have we prioritized quality goals when they conflict with each other?
- ☐ 5.5 **P2** Have we defined how to measure the success of the migration?
- ☐ 5.6 **P2** Do we have plans to collect user feedback after implementation?
- ☐ 5.7 **P2** Have we established a process for addressing issues and enhancements post-launch?
- ☐ 5.8 **P2** Can we validate that the promised benefits (e.g. faster time to market, lower cost of change) are actually realized?
- ☐ 5.9 **P3** Have we established processes to ensure we don't accumulate technical debt in the new system?
- ☐ 5.10 **P3** Do we have a plan for implementing continuous delivery practices that might have been promised in the business case?

> **Facilitator Note:** Quality goals should be specific and measurable, not generic "better performance." Help the team connect quality attributes to real business outcomes.

Technical Constraints

Technical stuff we can't change

- ☐ 6.1 **P1** Have we identified legacy interfaces that must be maintained during and after migration?
- ☐ 6.2 **P1** Do we understand technology stack requirements or limitations?
- ☐ 6.3 **P1** Have we mapped integration points with external systems?
- ☐ 6.4 **P1** Do we know infrastructure and deployment constraints we must work within?

- ☐ 6.5 **P2** Have we evaluated what regulatory or compliance requirements constrain our technical choices?
- ☐ 6.6 **P2** Do we understand data residency or sovereignty requirements?

> Where data must be stored, how it can be processed, and which laws apply to it based on geographic location.

- ☐ 6.7 **P2** Have we identified any vendor lock-in situations that limit our options?
- ☐ 6.8 **P2** Do we know what existing licenses or contracts affect our technology choices?
- ☐ 6.9 **P3** Have we evaluated the impact of existing monitoring and operational tooling on our choices?
- ☐ 6.10 **P3** Do we understand network and security constraints that may affect the new architecture?

Facilitator Note: Technical constraints are often the most comfortable area for technical teams, but don't let them get stuck here. Keep the focus on constraints that truly limit options.

Migration Strategy

How we're going to do this

- ☐ 7.1 **P1** Have we designed an iterative/incremental approach that reduces risk?
- ☐ 7.2 **P1** Have we identified suitable candidates for early migration?

> Considering both high-value components and low-hanging fruit, with clear criteria for what constitutes meaningful early wins (e.g. 10% revenue threshold)?

- ☐ 7.3 **P1** Do we have a data migration strategy that ensures business continuity?
- ☐ 7.4 **P1** Have we explicitly ruled out big bang approaches in favor of incremental migration?
- ☐ 7.5 **P1** 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.

- ☐ 7.6 **P1** Have we planned for parallel operations during transition if needed?
- ☐ 7.7 **P1** Have we considered tackling critical aggregators (essential reports/functions) early rather than last? Which?

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

- ☐ 7.8 **P1** 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.

- ☐ 7.9 **P1** Have we considered implementing frequent delivery practices from the beginning of the migration to validate future delivery capabilities?
- ☐ 7.10 **P2** Have we defined rollback procedures in case of migration issues?
- ☐ 7.11 **P2** Do we have a plan for handling legacy system maintenance during migration?
- ☐ 7.12 **P2** Have we established a testing strategy for verifying functionality post-migration?
- ☐ 7.13 **P2** Have we planned for performance and load testing at each migration phase to ensure the new system can handle production workloads?

	Consider testing both individual migrated components and the overall system performance as load shifts between old and new systems.

- ☐ 7.14 **P2** Do we have a strategy for testing data integrity and consistency between old and new systems during parallel operations?

	Include plans for automated data reconciliation checks and handling of data discrepancies during transition.

- ☐ 7.15 **P3** Have we explored applying the Strangler Fig and Bridge to the New Town patterns to gradually replace functionality?

	Patterns where new functionality gradually takes over from legacy code by

	intercepting calls (Strangler Fig) or through intermediate connecting layers during transition (Bridge to the New Town). When deciding for a pattern like Strangler Fig, also consider that while functionality migration may be straightforward, data migration complexity can be significantly higher.

- ☐ 7.16 **P3** 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.

Facilitator Note: Migration strategy often generates the most debate. Focus on establishing principles (incremental, value-driven) rather than detailed execution plans in the workshop.

Key Architectural Decisions

Critical decisions that shape the migration

- ☐ 8.1 **P1** Have we made key technology stack decisions for the target architecture?
- ☐ 8.2 **P1** Have we decided on the overall migration pattern (Strangler, Big Bang, Event Interception, etc.)?
- ☐ 8.3 **P1** Have we chosen our approach to data migration and synchronization?
- ☐ 8.4 **P1** Have we decided how to handle integration with systems that aren't being migrated?
- ☐ 8.5 **P1** Have we established short feedback loops for early validation of migration approaches?
- ☐ 8.6 **P1** Are we regularly providing business value through incremental delivery?
- ☐ 8.7 **P1** Do we have a clear ownership model for components during transition and after migration?
- ☐ 8.8 **P2** Have we chosen patterns for maintaining data consistency during migration?
- ☐ 8.9 **P2** Have we decided on our testing and validation approach for migration increments?
- ☐ 8.10 **P2** Can we implemented frequent releases to prove our ability to deliver quickly

post-migration?

- ☐ 8.11 **P2** Have we considered test automation that supports rapid, confident changes?
- ☐ 8.12 **P2** Have we documented architectural decisions for future reference?
- ☐ 8.13 **P3** Have we decided on our approach to handling transitional states and dual-system operations?
- ☐ 8.14 **P3** Have we decided on monitoring and observability approaches for the migration process?
- ☐ 8.15 **P3** Have we chosen our approach to feature flags or configuration management during transition?
- ☐ 8.16 **P3** 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.

- ☐ 8.17 **P3** Do we have a plan to eventually decommission any remaining legacy components?
- ☐ 8.18 **P3** Have we designed our testing approach to validate that migrated functionality performs equivalently to legacy systems under real-world conditions?

Beyond functional testing, ensure performance, reliability, and user experience match or exceed legacy system capabilities.

- ☐ 8.19 **P3** Have we considered approaches for parallel performance measurement and validation (such as tools like GitHub Scientist) to compare old and new system behavior under real conditions?

Facilitator Note: Focus on decisions that can't be easily changed later. Avoid getting bogged down in implementation details that can be refined during execution.

Risks & Mitigation

What could go wrong & how we'll handle it

- ☐ 9.1 **P1** Have we identified the top technical risks that could derail the migration?
- ☐ 9.2 **P1** Have we identified business continuity risks during the migration process?
- ☐ 9.3 **P1** Do we have data migration risks and verification strategies identified?
- ☐ 9.4 **P1** Have we planned contingencies for our most critical dependencies?
- ☐ 9.5 **P1** Have we identified potential resistance points among user groups?
- ☐ 9.6 **P2** Do we have a plan to address fears about job security or role changes?
- ☐ 9.7 **P2** Have we accounted for training needs for different user groups?
- ☐ 9.8 **P2** Do we understand how daily workflows will change and how to support that transition?
- ☐ 9.9 **P2** 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.

- ☐ 9.10 **P2** Do we have champions in each business unit to help promote the change?
- ☐ 9.11 **P3** Have we communicated the benefits of the new system to all affected parties?
- ☐ 9.12 **P3** Have we considered how to manage workload during the transition period?
- ☐ 9.13 **P2** Have we avoided heavyweight change processes that contradict our future delivery goals?
- ☐ 9.14 **P3** 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.

Facilitator Note: Risk identification often reveals gaps in earlier analysis. Use risks to validate previous sections and identify areas needing more detailed planning.

System Consolidation (if applicable, not in canvas)

Special considerations when merging multiple legacy systems

- ☐ 10.1 **P1** Have we mapped feature parity requirements between the systems being consolidated?
- ☐ 10.2 **P1** Do we understand the different user experiences and expectations for each system?
- ☐ 10.3 **P1** Have we identified potential conflicts in business processes between systems?
- ☐ 10.4 **P1** Do we have a strategy for data reconciliation between disparate systems?
- ☐ 10.5 **P2** Have we established decision-making criteria for resolving conflicting requirements?
- ☐ 10.6 **P2** Do we understand the organizational impacts of merging user communities?
- ☐ 10.7 **P2** Have we documented terminology differences to ensure consistent understanding?
- ☐ 10.8 **P3** Have we re-evaluated the assumption that all systems need to be consolidated rather than some retired?

> **Facilitator Note:** System consolidation adds significant complexity. If applicable, ensure these questions get P1 attention as they fundamentally impact migration strategy.

Post-Migration Success (if applicable, not in canvas)

Ensuring we actually deliver the promised business value

- ☐ 11.1 **P1** Do we have a maintenance and support plan for the new system?
- ☐ 11.2 **P1** Do we have data/metrics to measure the success of the migration?
- ☐ 11.3 **P1** Do we have knowledge transfer plans to operational teams?
- ☐ 11.4 **P1** Have we documented architectural decisions for future reference?
- ☐ 11.5 **P2** Do we have plans to collect user feedback after implementation?
- ☐ 11.6 **P2** Have we established a process for addressing issues and enhancements post-launch?

- ☐ 11.7 **P2** Have we validated that the promised benefits (faster time to market, lower cost of change) are actually realized?
- ☐ 11.8 **P2** Have we implemented continuous delivery practices that were promised in the business case?
- ☐ 11.9 **P2** Have we scheduled a retrospective to capture lessons learned?
- ☐ 11.10 **P3** Have we established processes to ensure we don't accumulate technical debt in the new system?
- ☐ 11.11 **P3** Do we have a plan to eventually decommission any remaining legacy components?

> **Facilitator Note:** Post-migration success is where many projects fail. Technical completion ≠ business success. Ensure the team commits to measuring and validating actual business outcomes.

Workshop Summary & Next Steps

- **Key Decisions Made:**
- **Critical Risks Identified:**
- **Immediate Follow-up Actions:**
- **Important Follow-up Actions:**
- **Future Considerations:**