Reporting & Analytics

Data Migration

A Comprehensive Guide to Planning, Executing, and Succeeding with Your Software Transition

bqe

Table of Contents

Introduction: "The Only Thing I'm Worried About"

A principal at a 20-person civil engineering firm spent seven months trying to implement Oracle NetSuite. He had a third-party consultant, a project plan, and full executive team buy-in. Seven months and a significant sum of money later, he pulled the plug.

When he eventually called us at BQE, the first thing he said wasn't "tell me about your features." It was: "*The only thing I'm worried about is that we don't lose time with your implementation*."

He'd been through it. He knew what a bad migration felt like — the false starts, the data cleanup that takes three times as long as anyone estimated, the staff who quietly return to their spreadsheets while the new system sits half-configured. He wasn't afraid of switching software. He was afraid of doing it poorly again. He couldn't afford another slip-up.

That fear is legitimate. It's also the most common reason A/E firms delay switching platforms long after they've decided their current one isn't working. But don't let this hold you back from switching to the software platform that can drive operational improvements. Migration doesn't have to be painful or chaotic

This guide is written for firms at that inflection point — already convinced a change is necessary, not yet convinced they can pull it off without chaos. Whether you're evaluating platforms, mid-negotiation, or already committed and staring down the implementation ahead of you, what follows is a practical account of what a software migration actually involves for architecture and engineering firms.

The generic migration guides that come packaged with most enterprise software are written to reassure you. They describe a clean, linear process — data in, data out, go-live on schedule. Meanwhile, the firms who've actually been through it describe something messier: decisions that nobody told them they'd have to make, internal time commitments that weren't in the original scope, and a parallel running period that stretched longer than anyone planned.

The gap between what's promised and what's experienced is exactly where migrations go sideways. This guide tries to close that gap.

What follows covers the full arc: how to choose the right migration scope for your firm, what data actually transfers and what doesn't, what realistic timelines look like by firm size, who does the work on each side, and what the first 90 days after go-live look like in practice. Where relevant, we draw on our experience implementing BQE CORE with hundreds of A/E firms — but the principles here apply to any platform transition in the architecture, engineering, consulting, and professional services space.

The firms that migrate successfully aren't the ones with the cleanest data or the most IT resources. They're the ones who went in knowing what they were actually signing up for. Then making the decisions that set them up for success throughout implementation and beyond.

Why A/E Migrations Go Wrong

Failed software implementations at architecture and engineering firms aren't random. They follow a pattern — and the pattern almost never starts with bad data. It starts with the wrong tool. NetSuite was built for product companies. Deltek software was originally built for government contractors and later adapted for A/E with minimal investment over time. SAP was built for manufacturers. Each of these platforms has been sold to A/E firms, and each has left a trail of expensive, half-finished implementations behind it — not because the software is bad necessarily, but because it was designed around workflows that have nothing to do with how a design firm actually operates.

Phases, fee structures, reimbursables, consultant markups, hourly billing against fixed-fee projects — these aren't edge cases in A/E practice. They're the core of it, and a platform that treats them as customizations is going to fight you at every step. This is why trying to force your data into a system not designed to work the way you do is challenging.

The second failure point is the implementation team. When a large software vendor sells you a platform, they typically send a partner — a third-party consultant who has been through a certification program and knows the software in the abstract. What they don't know is your firm, your data structure, or the specific ways A/E project accounting diverges from standard financial software assumptions. You end up paying consulting fees to educate someone who is supposed to be educating you. By the time the implementation is visibly off track, you're six months in, and the consultant has moved on to their next engagement.

The third factor is scope — specifically, overscope. The instinct when switching platforms is to bring everything: every project, every invoice, every time entry, going back to the firm's founding. It feels like the responsible thing to do. In practice, it adds months to the implementation timeline, inflates cost, and often produces a cluttered system that staff quietly resist using. The data that seemed essential during planning turns out to be historical records nobody opens once the new system is live. This is something you control, and it is worth carefullty considering exactly what data you need for your new software to be most effective. Defaulting to, “all of it,” isn’t usually the best path for most firms.

Wrong tool, wrong team, and wrong scope account for the vast majority of A/E migration failures — and all three are decisions made before the first data export runs.

Why Migration Is Complex

- The complexity of a data migration is driven by several factors that are unique to your organization and the systems involved:
- Data volume and variety: Large datasets, or datasets with many different record types, take longer to process and validate.
- Data quality: Years of accumulated records often contain duplicates, incomplete entries, inconsistent formatting, or outdated information.
- Structural differences: The source and destination systems may organize data very differently, requiring transformation and mapping.
- Business continuity: Migration must be planned around your operational schedule to minimize disruption.
- Regulatory and compliance requirements: Certain data may be subject to retention, privacy, or audit requirements that affect how it is handled.
- Multiple data sets: if you have migrated between software in the past, you probably have multiple data sets that would need new mapping to the latest software system.

Common Migration Scenarios

Migration Guide	Description	Typical Complexity
System Replacement	Retiring one platform and moving all data to a new one.	Medium to High
Cloud Migration	Moving from on-premise systems to a cloud-based solution.	Medium to High
System Consolidation	Merging data from multiple systems into a single platform.	High
Version Upgrade	Moving from an older to newer version of the same software.	Low to Medium
Partial Migration	Migrating only a subset of data or modules.	Low to Medium

Recognizing the Warning Signs

Wrong Tool: The clearest signal is a demo that doesn't look like your work. If a vendor walks you through a scenario built around inventory management, retail sales orders, software product development, manufacturing workflows, or is a general tool built for all small businesses — and then tells you it works "the same way" for A/E firms — that's not helpful. It probably means their software won't be the right fit. Purpose-built A/E software handles phase-based billing, reimbursable tracking, project and resource scheduling, and consultant markups as native features, not configurations. If those require add-ons, workarounds, or professional services to implement, factor that into the real cost of the platform and whether migrating your data into their system will work for you.

Wrong Implementation Team: Before signing with any vendor, ask directly: does the implementation team work for the software company, or are they a certified third-party partner? If it's a third party, make sure you speak to that company and the individual people who will be working with you. Ask how many A/E firm implementations they have personally led — not their firm, them. Ask who your primary contact will be after go-live, and whether it's the same person who sets up the system. The answers will tell you more about implementation quality than anything in the contract.

Wrong Scope: Scope decisions made under pressure at the start of a project tend to expand rather than contract. A useful gut-check before finalizing migration scope: in the past 90 days, how many times did you open a project from five or more years ago in your current system? Which reports do you actually run — and how far back does the data underneath them need to go? What would the system need to contain to run your business on day one, versus everything you've ever had? The answers to those three questions should define your scope, not a vendor's default package. And remember, this is one of the areas where often it is the firm's decision that drives additional complexity that makes implementation harder. Carefully decide what you actually need. Don't default to keeping all data just for the sake of it. Only migrate the data you will actually use.

Key Insight

No two migrations are identical. Even if you are migrating from the same source system as another organization, your data history, business processes, and configuration will produce a unique set of challenges. This is why upfront planning and data assessment are so important.

Pre-Migration Planning

The decisions that determine whether a migration succeeds or fails are made in the two or three weeks before any data moves — when the team is assembled, the scope is defined, and the governance rules are set. Firms that treat this phase as administrative box-checking pay for it later. Firms that treat it as the most consequential phase of the entire project almost always have smoother go-lives.

The single most important planning decision is who owns the migration internally. Not a committee, not a shared responsibility across the leadership team — one person with the authority to make data decisions quickly and the bandwidth to stay engaged throughout the process. Call this person the data champion. They are the main point of contact with the implementation team, the person who resolves ambiguities when the data throws a curveball, and the one who signs off before go-live. Migrations stall most predictably when this role is either undefined, filled by multiple people at once, or assigned to someone who is already overcommitted. Protecting that person's time is not a soft recommendation — it is the single most controllable factor in your timeline.

Scope definition deserves the same seriousness. The temptation in early planning conversations is to say yes to everything — bring all the historical data, migrate every module, go live on all divisions simultaneously. It feels thorough. What it actually does is turn a manageable 8-week implementation into a 6-month project that consumes more internal time than anyone budgeted. The right question to ask during scope definition isn't "what do we have?" It's "what do we actually need on day one to run the business?" Everything else can be evaluated against that standard, archived separately, or phased in later.

The planning framework below doesn't need to be elaborate. For most A/E firms, it fits on a few pages of documentation and a single kickoff meeting. The value isn't in the paperwork — it's in forcing the decisions that are easy to defer and expensive to revisit once migration is underway.

Assemble Your Migration Team

A data migration is a cross-functional effort. The following roles should be identified and engaged before migration work begins:

Role	Responsibilities
Project Sponsor	Executive stakeholder who owns the migration initiative, resolves escalations, and ensures organizational alignment. This is the final decision maker.
Project Manager	Coordinates tasks, manages timelines, facilitates communication between internal and vendor teams. This person leads the day to day aspect of the migration process.
Data Owner(s)	Subject matter experts who know the business meaning of your data and can validate accuracy. These folks should be providing the data needed to be migrated, and help determine what data to keep and migrate, and what data could simply be stored in an archive without migrating to the new system.
IT / Systems Administrator	Manages access, security, and technical infrastructure for both source and target systems. They should be able to answer any technical questions and give access to any software or system needed to execute the migration.
End-User Representatives	Provide feedback on data completeness and usability from a day-to-day workflow perspective. This could be a principal, project manager, or other member of the team who will be using the new system on a regular basis. It might be best to have a representative from different departments, like design, management, finance, and marketing.
Vendor / Implementation Partner	Provides migration tooling, methodology, and technical expertise for the target system. Ideally, these should be employees of the new software you are migrating to.

Conduct a Data Inventory

Before any migration work begins, you need a clear picture of what data you have and where it lives. A data inventory should capture:

- All data categories (e.g., customers, vendors, transactions, products, employees, documents)
- Data volumes for each category (record counts, file sizes)
- Data formats and current storage locations
- Data relationships and dependencies between records
- Data that is actively used vs. archived or historical
- Data that is subject to regulatory or compliance requirements 2.3

Important:
Do not underestimate how long a thorough data inventory takes. For organizations with years of accumulated data across multiple systems, this step alone can take several weeks. Start early.

Define Scope and Priorities

Not all data needs to be migrated, and not all data should be migrated. Work with your team to define:

- In-scope data: What records and modules are required in the new system on Day 1?
- Historical data: How far back does your history need to go? Full history, last 3 years, last 7 years?
- Archived data: Will legacy data be archived separately or left in the old system for reference?
- Excluded data: What can be left behind (e.g., obsolete records, test data, decommissioned entities)?

Defining scope early prevents scope creep during migration and helps set realistic timelines. Decisions made here directly impact both cost and complexity.

Establish a Data Governance Policy

Before migration, agree on governance rules that will guide data decisions throughout the project:

- Who has authority to approve data changes or exclusions?
- How will data disputes or ambiguities be resolved?
- What is the process for documenting decisions made during migration?
- How will data quality issues be prioritized and addressed?

Key Insight

Planning is not about creating documentation. It's about making decisions before you're under pressure to make them. The firms that move through migration fastest are the ones that walked into the process knowing who owns each decision, what data is in scope, and what "good enough on day one" actually means for their firm.

Data Assessment and Cleanup

Migration doesn't create data quality problems — it reveals them. Every firm accumulates years of imperfect data entry: duplicate client records, projects assigned to the wrong billing category, time entries linked to closed phases, phone numbers stored in six different formats across as many users. In the old system, the team works around these issues through habit and institutional memory. Someone knows that "ABC Consulting" and "ABC Consulting LLC" are the same client. Someone knows that the project coded under the wrong phase was never fixed because it would have taken too long. In the new system, those workarounds don't exist yet, and the institutional knowledge that compensated for bad data doesn't transfer with it.

This is the phase where firms most consistently underestimate the work involved. The implementation team can provide tools, templates, and guidance, but only your team has the business knowledge to make accurate decisions about your own data. Whether a duplicate record should be merged or archived, whether an orphaned project entry should be resolved or excluded, whether an inactive client belongs in the new system at all — those aren't technical questions. They require someone who knows your firm's history to sit with the data and make judgment calls. That takes time, and the only way to account for it accurately is to build it explicitly into the project plan before the migration starts.

The payoff for doing this work well is significant. Firms that arrive at go-live with clean, well-mapped data spend their first weeks learning the new system, not correcting the old one. Firms that skip or rush the cleanup phase spend those same weeks rebuilding confidence in data that staff have already learned not to trust. The quality of your new system on day one is almost entirely determined by the quality of what you bring into it. .

Common Data Quality Issues

When assessing your existing data, look for the following common issues and clean it up whenever possible, prior to starting a data migration:

Issue	Description & Impact
Duplicate Records	Multiple entries for the same customer, vendor, entity, or expense. Causes confusion, inflated counts, and workflow errors in the new system.
Incomplete Records	Fields that are required in the new system but missing in the source data. Identify what must be populated before or after migration.
Inconsistent Formatting	Phone numbers, addresses, dates, and codes stored in multiple formats. Requires standardization to migrate cleanly so make these decisions prior to migrating.
Orphaned Records	Records that reference deleted or non-existent parent records. These must be resolved or excluded.
Outdated / Stale Data	Records for customers, products, or accounts no longer active. Evaluate whether to migrate or archive.
Non-Standard Characters	Special characters or encoding issues that can corrupt data during transfer.
Incorrect Categorization	Records assigned to wrong categories, types, or statuses based on how your new system defines them.

Your Role in Data Cleanup

Data cleanup is a shared responsibility. While your implementation partner may provide tools and guidance, only your team has the business knowledge to make accurate decisions about your data. You should expect to:

- Export and review data extracts provided by your migration team
- Identify records that should be merged, updated, or excluded
- Resolve ambiguous or conflicting records by consulting internal subject matter experts

Plan Your Cleanup Time Carefully

Data cleanup is consistently underestimated. A realistic estimate is that for every 10,000 records, your team should budget 2–4 hours of review and decision-making time—more if data quality is poor. Build this into your project timeline explicitly.

- Update records directly in the source system where possible
- Sign off on data quality before migration runs proceed

Data Mapping

Data mapping is the process of defining how each field in your source system corresponds to a field in the target system. This is a technical and business exercise that requires input from both your team and your implementation partner.

Key mapping decisions include:

- Direct mapping: Fields that translate 1:1 between systems with no transformation needed.
- Transformation mapping: Fields that require reformatting, calculation, or logic to map correctly.
- Default values: Fields required in the new system that have no source equivalent—a default value must be agreed upon.
- Unmappable data: Fields in the source system that have no equivalent in the target—decide whether to archive or discard.

A completed data mapping document should be reviewed and approved by both business and technical stakeholders before migration begins. Work with your new software's implementation team to work through all of these decisions. They are a great resource for guidance through these steps.

Key Insight

The quality of your new system on day one is almost entirely determined by the quality of what you bring into it. Data cleanup is not a vendor responsibility — it's a firm responsibility that requires people who know your data to make decisions about it. Budget the time honestly, and treat it as an investment in everything that follows.

Timeline and Phase Expectations

The first question almost every firm asks when starting a migration is "how long will this take?" It's the right question, and the honest answer is that the timeline is more within your control than you might expect. Technical complexity plays a role — data volume, system configuration, the cleanliness of your source data — but the single most consistent cause of migration delays isn't technical at all. It's the firm's own team being too stretched to show up for the review, validation, and decision-making work when it comes due.

A well-structured migration follows a defined sequence of phases, and each phase has a natural handoff point where the implementation team needs input, approvals, or decisions from your team. When those handoffs happen quickly, the project moves. When they stall — because the data champion got pulled into a deadline, because the principal who needs to sign off is traveling, because nobody blocked time for the validation review — the timeline slips. Not because something went wrong technically, but because the queue stopped moving. This is why protecting your team's migration time before the project starts is not a soft recommendation. It is the most concrete thing you can do to influence your go-live date.

The phase framework below gives you a clear picture of what to expect and when your team's participation is most critical. Read through it before you agree to a go-live target. The deliverables and handoffs at each stage are predictable — which means you can look at your firm's calendar, identify the windows where availability will be tight, and plan around them rather than discover them mid-project.

Typical Migration Phases

Phase	Description & Key Activities
1. Discovery & Planning	Define scope, assemble team, conduct data inventory, establish governance. Deliverable: Migration Plan document.
2. Data Assessment	Extract and analyze source data, identify quality issues, produce data quality report.
3. Data Cleanup	Remediate data quality issues in the source system. Customer-led with partner guidance.
4. Data Mapping	Define field-level mapping between source and target systems. Reviewed and approved by both parties.
5. Test Migration	First migration run into a test/sandbox environment. Identify gaps, errors, and missing data.
6. Validation & Remediation	Customer team validates migrated data. Issues are documented, triaged, and resolved.
7. Final Sign-Off	Customer formally approves migrated data for go-live. Any open issues are documented.
8. Production Migration (Cutover)	Final migration run into the live production environment. Occurs at or near go-live.
9. Post-Migration Validation	Spot-checks and user acceptance testing in the live environment. Support is heightened.

What Determines Timeline Length

Migration timelines vary widely—from a few weeks to many months—depending on the following factors:

- **Data volume:** More records require more processing, mapping, and validation time.
- **Data quality:** Poor quality data requires more cleanup effort before migration can proceed.
- **Scope:** Migrating all historical data takes longer than migrating active records only.
- **Customer availability:** Migrations stall most often when customer teams are unavailable to review, approve, or provide decisions.
- **System complexity:** Highly customized source systems or complex data relationships require more mapping effort.

The #1 Cause of Migration Delays

Customer resource availability. Data review, cleanup, and validation require your team's active participation. When internal teams are stretched thin or unavailable, timelines slip. Protect your migration resources and set clear expectations internally before the project begins.

Setting Realistic Expectations

Entering a migration with honest expectations makes the difference between a team that stays confident through the process and one that loses trust in the project when the first complication surfaces. Three things are worth naming upfront:

Test migrations will reveal issues. The first test migration is rarely perfect — nor is it supposed to be. Its purpose is to identify gaps before they reach the production environment, not to demonstrate that everything is ready. Treat issues found here as the process working correctly.

Some data will require decisions. You will encounter records where the right course of action is genuinely unclear. Have your decision-makers identified and responsive before this phase begins, not after the first issue surfaces.

Not everything migrates perfectly. Some edge cases, custom fields, or legacy data may not have a clean path to the new system. Plan for manual entry of exceptions — and keep a running log of these decisions so they don't have to be relitigated after go-live.

Key Insight

The go-live date you agree to at the start of a migration is a projection, not a guarantee. What converts it into a reality is your team's availability at the critical handoff points — data cleanup, validation, and final sign-off. Block that time on the calendar before the project starts, and protect it the way you'd protect a client deadline.

Roles, Responsibilities & Communication

When a migration stalls, the instinct is to look for a technical explanation. Most of the time, the actual cause is simpler: nobody was sure who was supposed to make a decision, so nobody made it. The implementation team filed an issue, the firm's project contact said they'd check with the principal, the principal was in a client meeting, and the queue stopped. Three days later the issue is still open. Two weeks later it has created two downstream problems that also need resolving. The migration didn't fail because of bad data or a software limitation. It failed because ownership was ambiguous and communication was informal.

This challenge is more pronounced in A/E firms than in larger organizations. Principals wear multiple hats. There is often no dedicated IT staff. The person with the deepest knowledge of the firm's data is also managing four active projects or may be an outside financial consultant. In that environment, it's tempting to assume the implementation partner will drive the process and surface issues as they need attention. That assumption is the source of more migration delays than any technical factor. The implementation team is responsible for the technical execution — the tooling, the migration runs, the documentation. They are not responsible for driving your firm's internal decision-making. That ownership has to come from your side, with a named person and a clear escalation path.

The communication structure doesn't need to be complicated. For most A/E firm migrations, a weekly check-in, a shared issue log, and a clear agreement on who approves what is sufficient. The value of establishing this at the start of the project — not after the first issue surfaces — is that it removes the ambiguity before it has a chance to create delays. Everyone on both sides knows where to go, who to ask, and what happens when something can't be resolved at the working level.

What Your Implementation Partner Is Responsible For

- Providing migration tooling, methodology, and technical execution
- Running test and production migration loads
- Documenting issues found during migration and escalating to the customer team
- Providing validation support and guidance on how to review migrated

data

- Documenting the final migration configuration for future reference

What Your Organization Is Responsible For

- Providing access to source system data and documentation
- Completing data cleanup and remediation in the source system
- Reviewing and approving data mapping decisions
- Allocating subject matter experts to validate migrated data
- Making timely decisions on data quality issues and exceptions
- Providing final sign-off before production migration
- Communicating internally to prepare your organization for the transition

Establishing a Communication Rhythm

- Migrations benefit from structured, regular communication. Establish the following at the start of your project:
- Weekly status meetings: Review progress, surface blockers, and confirm upcoming milestones.
- Issue log: A shared, living document tracking all data issues, their owner, and resolution status.
- Escalation path: A defined process for escalating decisions that cannot be resolved at the working level.
- Decision log: Documentation of all significant scope or data decisions made during the project

Key Insight

An implementation partner can execute a technically flawless migration and still deliver a delayed, frustrating project if ownership on the firm's side is unclear. Name the decision-makers before work begins, establish the communication structure in the first week, and treat the issue log as a living document — not a formality. Ambiguity is the most preventable cause of migration delays.

Data Validation

Validation is the phase where a migration becomes real. Up to this point, the work has been largely invisible to most of the firm — planning documents, data exports, cleanup sprints, mapping exercises conducted between your data champion and the implementation team. Validation is when your project manager, your financial controller, and your principal actually sit down with the new system and confirm that what they're looking at matches what they expected. That moment either builds confidence in the new platform or surfaces problems that need to be resolved before anything goes live.

The temptation when a go-live date is on the calendar is to treat validation as a checkpoint to get through rather than a process to take seriously. That instinct is understandable and worth resisting. A rushed validation that misses a financial reconciliation error or a broken record relationship doesn't save time — it defers the problem to a moment when it will be far more expensive to fix, because your team will now be trying to untangle it while running the business on an unfamiliar system. The right time to find issues is during validation, not during the first billing cycle after go-live.

Two things are worth establishing clearly before validation begins. First, the people doing the validation need to be the people who use the data, not just the project manager or IT lead. Financial balances need to be reviewed by whoever runs your books. Project records need to be checked by someone who knows your project structures. Validation done by people who don't work with the data day-to-day will miss things that a practitioner would catch immediately. Second, not every discrepancy found during validation is a migration error. Some will reflect data quality issues that existed in the source system before migration began. Distinguishing between the two matters — migration errors get fixed by the implementation team, pre-existing data quality issues get fixed by yours.

What to Validate

A thorough validation should cover the following areas:

Validation Area	What to Check
Record Counts	Does the number of migrated records match what was expected? Reconcile totals by category.
Data Completeness	Are all required fields populated? Are there unexpected blanks?
Data Accuracy	Do values in the new system match the source? Spot-check a representative sample of records.
Relationships & Links	Are related records correctly linked (e.g., contacts linked to accounts, invoices linked to customers)?
Financial Balances	Do financial totals, balances, and history reconcile to source system reports?
Calculated Fields	Are computed values (e.g., aging, totals, summaries) calculating correctly in the new system?
Attachments & Documents	Are associated files, attachments, and documents present and accessible?
Custom Fields	Are custom or organization-specific fields migrated with correct values?

Validation Best Practices

- Use source system reports as your validation baseline. Export key reports from the old system before migration so you have reference totals.
- Do not validate from memory. Always compare against documented source data.
- Prioritize by risk. Validate financial data and core transactional records first.
- Sample broadly. Do not only check records you know well—include records from different time periods, categories, and users.
- Document issues formally. Log every discrepancy in the shared issue log, even minor ones.
- Distinguish between migration errors and data quality issues. Not every discrepancy is a migration failure—some may reflect pre-existing data problems.

Sign-Off Process

- Before production migration, a formal sign-off should be obtained from appropriate stakeholders. This sign-off confirms that:
- Validation has been completed to an agreed standard
- All critical issues have been resolved or have an accepted workaround
- Open issues have been documented and accepted by the business
- The organization is prepared to proceed to go-live
- Sign-off is an important milestone—it represents your team's confirmation that the data is ready. Do not treat it as a formality.

Key Insight

Validation done by people who don't work with the data day-to-day will miss what a practitioner would catch immediately. The right people to validate financial records are the people who run your books. The right people to validate project records are the people who manage your projects. Put the right eyes on the data before go-live — not after.

Cutover and Go-Live

Cutover is the process of transitioning from your old system to your new one. It is the most high-stakes phase of the migration and requires careful coordination. Up to this point, the migration has existed alongside normal business — planning meetings, cleanup sessions, validation reviews, test migrations running in a sandbox environment. On cutover day, the new system becomes the system. That transition is meaningful in a way that earlier phases aren't, and it deserves a level of coordination that goes well beyond the technical execution.

The two things firms most commonly underestimate going into cutover are the data freeze and the communication plan. The data freeze — the point at which staff stop entering new data into the old system — sounds straightforward until someone realizes it means no new time entries, no new invoices, and no new project updates for a defined window while the final migration runs. Staff who aren't briefed on this in advance will keep working in the old system out of habit, creating a gap between what was migrated and what was entered after the freeze. That gap has to be manually reconciled, which is exactly the kind of post-go-live firefighting that erodes confidence in the new platform before people have had a chance to learn it. Clear communication to every user about when the freeze begins — and why — is as important as any technical step on the day-of checklist.

The question of whether to run both systems in parallel during the transition period is a genuine strategic decision, not a default. Parallel running reduces risk: if an issue surfaces in the new system, the old one is still available as a reference or fallback. The tradeoff is real workload — staff effectively maintaining two systems simultaneously, which adds overhead at exactly the moment they're already adjusting to new workflows. For most A/E firms, a shorter parallel period with heightened support coverage is preferable to an extended parallel period that creates fatigue. Some firms choose to stagger the cutover entirely, going live on one division or team first before rolling out firm-wide. That approach trades speed for risk reduction and works well when divisions operate with enough independence that a phased rollout is practical.

Cutover Planning

A cutover plan should be prepared well in advance of go-live and should address:

- Cutover window: The date and time of the final production migration. Choose a low-activity period.

- Data freeze: The point at which data entry in the old system must stop to ensure the final migration captures all activity.
- Parallel period: Will you run both systems simultaneously for a period? This reduces risk but increases workload.
- Rollback plan: What happens if a critical issue is discovered after cutover? Have a documented contingency.
- Communication plan: How will you notify staff, customers, and vendors of the transition and any expected downtime?

Day-of-Cutover Checklist

- Confirm data freeze has been enforced in the source system
- Confirm all validation sign-offs are complete
- Run final production migration load
- Perform post-migration smoke test (key records, balances, links)
- Confirm system access for all users in the new environment
- Communicate go-live status to all stakeholders
- Stand up heightened support coverage

Managing the Transition Period

The first days and weeks after go-live require deliberate attention. A few things to keep in mind:

- **Expect a learning curve.** Users adjusting to a new system will have questions — not all of them are data issues. Build a clear triage process so staff know where to go with questions and so real data problems don't get lost in the noise.
- **Keep the source system accessible.** Maintain read-only access to the old system for a defined period — 30 to 90 days is typical — so staff can reference historical records while getting comfortable in the new environment.
- **Monitor data closely in the first two weeks.** Inconsistencies that surface early are far easier to resolve than ones discovered after weeks of new activity have built up on top of them.
- Protect your support coverage. The people who can answer questions in week one are the most important resource you have. Don't let them get pulled to other work the moment go-live is declared.

Key Insight

The technical execution of cutover is largely the implementation team's responsibility. The coordination — communicating the data freeze, briefing staff, timing the rollout — is yours. Firms that arrive at go-live with every user briefed, every access credential confirmed, and a clear triage process in place for the first week are the ones that build early confidence in the new system. Firms that treat those things as post-go-live problems spend the first month recovering from them.

Risks and How to Mitigate Them

Risk in a data migration feels like a technical problem until you look at what actually causes projects to fail. Data loss — the thing most firms worry about most going in — is the easiest risk to manage. Back up the source system before every migration run, and you have a complete recovery option regardless of what happens during the process. The risks that actually derail migrations are harder to insure against because they're organizational, not technical. Scope creep, calendar-driven go-live pressure, insufficient internal resources — these don't show up in a system log. They show up in project timelines that quietly slip and in staff who arrive at go-live without enough confidence in the new system to use it effectively.

The table below names the risks that surface most consistently across A/E firm migrations and the mitigation strategy for each. None of them should be surprises at this point in the guide — they're variations of the same themes that run through every earlier chapter. Scope decisions made too loosely in planning become scope creep in execution. Resource commitments that aren't protected become availability problems at the validation phase. A go-live date set before quality is confirmed becomes a pressure that overrides good judgment. The pattern is consistent enough that it's worth naming directly: the mitigation for almost every risk on this list is a decision made early in the project, not a reaction to a problem that surfaces later.

One risk deserves particular attention because it's the hardest to recover from after the fact: user adoption. Every other issue on this list can be resolved with time, effort, and a clear escalation path. A team that has lost confidence in the new system — because their first experience with it was confusing, or because data they expected to find wasn't there, or because nobody was available to answer their questions in the first week — is a problem that doesn't resolve on its own. The most effective mitigation is involving end users in the validation process before go-live. When the people who will use the system daily have seen the data, confirmed it looks right, and had their questions answered before the switch is flipped, the transition lands differently than it does when the system simply appears on their desktop one morning.

Risk	Mitigation Strategy
Underestimating data quality issues	Conduct a thorough data assessment before planning the timeline. Budget cleanup time conservatively.
Insufficient customer resource allocation	Assign dedicated migration responsibilities. Protect team members from being pulled to other projects during critical phases.
Scope creep	Document and freeze scope early. Use a formal change request process for any additions. .
Data loss during migration	Maintain a full backup of the source system before every migration run. Never migrate without a backup.
Missed data dependencies	Map all relationships between data entities at the start. Test related records together, not in isolation.
Go-live date pressure overriding quality	Establish a clear readiness checklist. Go-live should be gated on data quality, not calendar dates.
User adoption issues post-go-live	Involve end users in validation. Their familiarity with the data improves acceptance and early issue detection. ?
Regulatory compliance gaps	Identify compliance requirements during planning. Involve legal and compliance teams early.

Post-Migration Considerations

Go-live is a milestone, not a finish line. The implementation team wraps up, access to the project is handed off to support, and the firm's attention returns to client work. That's the natural arc of every migration, and the relief that comes with it is earned. But a few decisions made in the weeks after go-live carry consequences that play out over months and years, and the ones that get skipped in the rush to return to normal tend to be the ones that matter most.

The most immediate is the old system. The instinct to shut it down quickly is understandable. But it is worth resisting for a defined period. Staff will have questions in the first weeks or months that are best answered by looking at the source data directly. Auditors and clients may request historical records. A discrepancy discovered 60 days after go-live may only be resolvable by going back to the original data. Keeping a single seat, or read-only access available for 90 days or so, costs almost nothing in the grand scheme of things and eliminates a category of post-migration problems that are otherwise very difficult to untangle.

The larger long-term consideration is data quality. A migration is a one-time event that gives a firm the opportunity to start with clean, well-structured data. What happens to that data over the following months and years depends entirely on whether the firm establishes standards, procedures, assigns someone responsibility for maintaining quality, and builds checkpoints into normal business operations. The firms that don't take this step will find themselves, three or four years from now, looking at the same duplicate records, inconsistent formatting, and orphaned entries that made the migration cleanup so time-consuming in the first place.

The migration didn't cause those problems — years of informal data entry did. Without a deliberate change to how data is managed going forward, the new system will accumulate the same issues at the same rate as the old one.

Decommissioning the Source System

Do not rush to decommission your old system. A responsible decommissioning plan includes:

- Maintaining read-only access for a defined period (typically 30–90 days post-go-live)
- Exporting and archiving source data in a non-proprietary format for long-term retention
- Confirming all regulatory retention obligations are met before deletion
- Documenting the decommission date and archival location for audit purposes

Ongoing Data Quality

Migration is a one-time event, but data quality is an ongoing discipline. After go-live:

- Establish data entry standards and train users on them
- Assign data stewards responsible for monitoring and maintaining quality by domain
- Schedule periodic data audits to identify and correct quality drift
- Build data quality checks into your business processes, not just your migration tools

Lessons Learned

After completing the migration, conduct a structured retrospective with your team and implementation partner. Capture:

- What went well and should be repeated in future projects
- What caused delays or issues and how they were resolved
- What you would do differently with the benefit of hindsight
- Recommendations for future migrations or system transitions

Key Insight

The new system starts with the cleanest data your firm has had in years. What it looks like three years from now depends on the entry standards and quality discipline you establish in the weeks after go-live — not on the migration itself. The migration cleaned the slate. Keeping it clean is a business process decision, not a software one.

Quick Reference: Key Questions to Ask

Everything covered in this guide can be distilled into a set of decisions that need to be made at specific points in the project. The checklist below isn't a substitute for the thinking in the preceding chapters — it's a prompt to make sure nothing falls through the cracks when the project is moving quickly and attention is divided. Run through it at each stage, and treat any "no" as a conversation to have before moving forward, not after.

Before Migration Begins

- Have we completed a full inventory of all data to be migrated?
- Is our migration scope documented and agreed upon with all stakeholders?
- Do we have dedicated team members assigned to migration activities?
- Have we assessed our data quality and developed a cleanup plan?
- Do we have a backup of our source data?
- Have we completed a data mapping exercise?

During Validation

- Have we reconciled record counts between source and target systems?
- Have we validated financial balances and totals?
- Have we spot-checked data accuracy across all major record types?
- Have we verified all record relationships and links?
- Have we logged and triaged all issues found?
- Have we obtained formal sign-off from appropriate stakeholders?

At Go-Live

- Is there a data freeze in place in the source system?
- Is a rollback plan documented and communicated?
- Is heightened support available for users in the first weeks?
- Is read-only access to the source system available for reference?
- Is there a clear issue triage process for post-go-live problems?

Final Thought

A successful data migration is not just a technical achievement—it is an organizational one. The organizations that succeed are those that treat migration as a business project, not just an IT project. Invest in planning, protect your team's time, and engage actively throughout the process. Your data is the foundation of your new system—make sure it is solid.

Conclusion: What Success Actually Looks Like

The principal at the civil engineering firm from the introduction — the one who pulled the plug on seven months of failed implementation — eventually completed a migration. He chose BQE CORE the second time around, and he went in differently: an implementation team made up of BQE employees who knew the software inside and out, and a deliberate decision to start with current project data rather than trying to bring a decade of history across all at once. The go-live took ten weeks. His team was fully operational within the first month. When asked what was different, he didn't talk about the software. He talked about knowing what he was signing up for before the work started — and having a team on the other side that was as invested in the outcome as he was.

That's what successful migration actually looks like. Not seamless — there were data discrepancies to resolve and questions from staff in the first weeks that needed answers. But manageable, because the scope was right, the team was prepared, and the process had a clear owner on the firm's side. The problems that surfaced were expected problems, handled through a process that was already in place. None of them became crises.

A successful migration is an organizational achievement, not a technical one. The implementation team can execute flawlessly — clean data imports, accurate mapping, a smooth cutover — and the project can still land badly if internal ownership is unclear, the timeline wasn't protected, or staff arrived at go-live unprepared. The inverse is equally true: firms that go in with realistic expectations, a named decision-maker, a scoped plan, and team members who were involved throughout consistently have better outcomes regardless of how complicated their data situation was at the start. The firms that struggle are the ones that treat migration as something that happens to them. The ones that succeed treat it as something they are actively managing — with the same attention and ownership they bring to a complex client project.

Your data is the operational foundation of your new system. The care you put into moving it determines what you're building on. The process outlined in this guide is designed to make that foundation solid — not by eliminating every complication, but by making sure none of them catch you off guard.

If you're evaluating BQE CORE or planning your implementation, the next step isn't a sales conversation. It's a migration consultation — a direct conversation with our team about your firm's specific data situation, what path makes sense, and what a realistic timeline looks like. Bring your questions and bring the hard ones. That conversation is exactly what our implementation team is there for.

Explore More Resources

Running a better architecture or engineering firm takes more than experience. It requires clear insight, practical tools, and guidance you can trust. This is where we can help. Our resource library is built for firm leaders who want to improve financial performance, strengthen project delivery, and make more confident decisions. Inside, you will find webinars that turn complex topics into clear actions, articles that challenge how firms operate and grow, and reports and ebooks grounded in real data and experience.

You can also put ideas into practice with tools, templates, and checklists designed for immediate use, connect with peers through user groups, and learn from real customer success stories. Every resource is created with one goal in mind: to help you run a better firm. Explore what is most relevant to you and take the next step forward

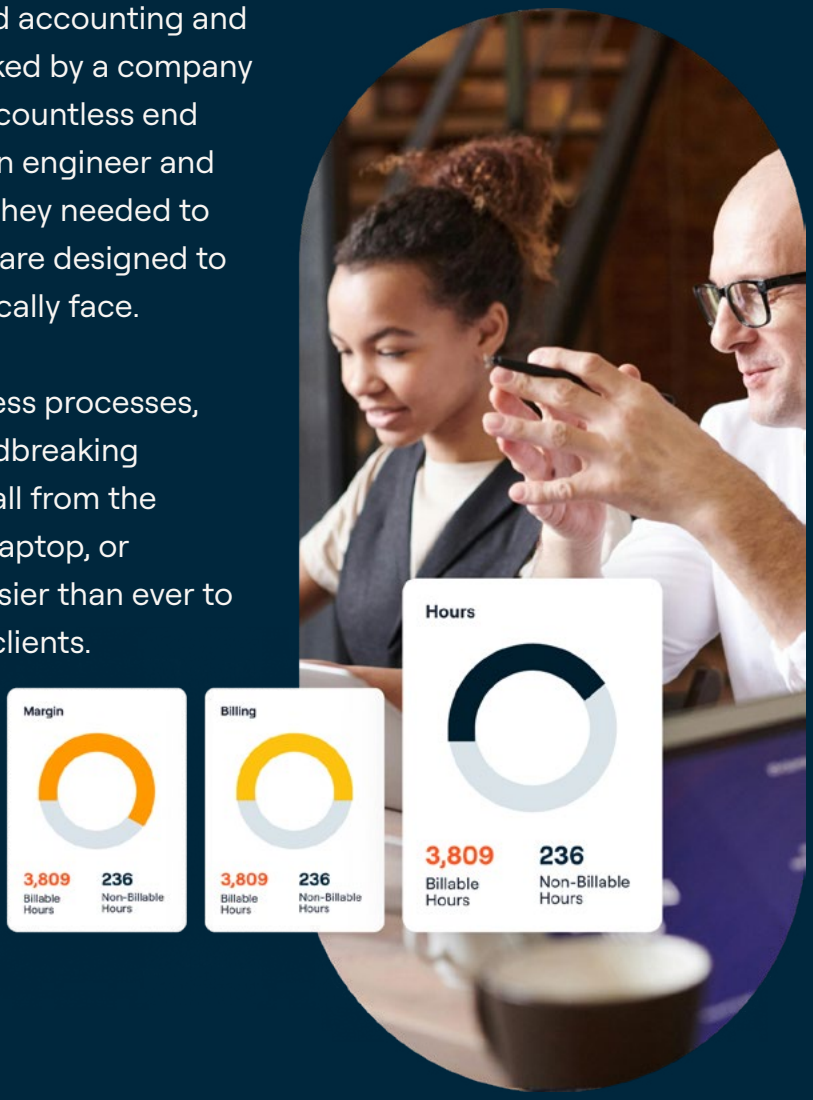
Webinars **Reports & Ebooks** **Tools & Templates**
Customer Success **Blog** **User Groups**

BQE CORE

If you're serious about improving efficiency and profitability, you need access to technology that makes the process of tracking Key Performance Indicators (KPIs) straightforward. BQE CORE is an all-in-one firm management platform with integrated accounting and project management tools that's backed by a company with over 30 years of experience and countless end users worldwide. It was designed by an engineer and architect to give their firms the tools they needed to thrive. Thus all of the built-in features are designed to address the pain points A/E firms typically face.

BQE CORE can streamline your business processes, while simultaneously providing groundbreaking insights that will help grow your firm, all from the convenience of a desktop computer, laptop, or mobile device. BQE CORE makes it easier than ever to collaborate with both your team and clients.

Running your business should be exciting, not chaotic. That's why we're here, and most importantly, why we developed CORE.



bqe

TO LEARN MORE, BOOK A DEMO →