# The Ultimate Guide to Implementing Membership Software

*Your Playbook to Getting Started with a New System*

Aptify®

# Table Of Contents

# Congratulations!

Your organization decided it needed new membership software. You built an RFP, met with vendors, made a selection, got board approval, negotiated contracts, and probably went through countless other steps along the way.

You’ve seen what your new software can do and you’re anxious to get started.

# Introduction

The implementation phase is that difficult time where a lot of work has to happen before you can reap the rewards of your new system. There are long hours, conflicting ideas, and various obstacles along the way. The excitement from the selection process starts to wane and you're faced with new processes, new procedures, and new things to learn.

But the news isn't all bad. This is a time of opportunity.

Implementation allows you to review your processes and change what hasn't worked. It gives you permission to question priorities and to build a team that will lead your staff into new territory.

Change is hard, but it also can be rewarding.

***The Ultimate Guide to Implementing Membership Software* will show you**

- how to keep your implementation on time and on budget,
- how to build the right team and communicate with them so everyone stays on pace,
- how to set the right expectations with your entire staff so that you'll have organizational buy-in throughout the experience, and
- how the implementation timeline is built, so you have full insight into the process of what happens when.

Let's get started…

# Staying on Budget

There's no disputing a new membership system is a major investment. Managing the costs of your new membership software extends well beyond the shopping and selection stages. As you enter the implementation phase, there are several strategies you can employ to make sure the project stays within budget and in scope.

## Be Prepared for Change

Successful implementation projects require time, patience, and flexibility. Rarely does the initial plan that organizations create turn into a reality, as changes to the plan often have to occur. Going into this process with realistic expectations is essential and will help you tackle each phase of the project in a more productive manner.

The key to keeping your implementation within budget is to look to best practices—an implementation that's done in the way that others have done it in the past means there's no recreating the wheel. The more you can commit to following best practices, the more predictable and the lower your implementation costs will be.

Following best practices, though, is directly tied to being ready for change. Chances are, what you've always done in the past won't be the best way forward. **After all, if you're not looking to make a change in process—if you're looking to do what you've always done—do you really expect different results?**

Be open to the idea of change when your implementation team suggests a best practices rule. Sometimes this means that you'll need to prompt those in senior management to accept something new, other times you'll have to finesse a conversation with the board about changing policies and bylaws to adopt a new process.

Remember, the end result is a system that best helps you achieve your organization's goals. There may be new paths to get there, but the result will be better—and less costly—in the end.

Here, we'll show you some cost-saving techniques that apply directly to a software implementation. For a full look at what you can expect to pay for an entire system, along with some tips about gaining board approval, check out the [Pricing Toolkit](Pricing Toolkit).

# Work in Phases

Most organizations have complex needs, so when mid- to large-sized organizations set out to gather all their requirements, the list can get very long, very quickly. The best way to tackle this—and to keep costs down—is to work in phases and focus first on a subset of what's needed. For instance, plan to implement half of the full requirements in the first phase, then in the next quarter the second, and remainder in the third.


While this could be seen as more of a cost-delay model than a cost-reduction model, it's not. And here's why. When organizations embark on implementing a leaner amount of requirements right out of the gate, they end up needing to implement less in the long run. Those who decide to implement in phases find that, once they go live after phase one and get comfortable with working in their new system, their total requirements are actually reduced.

Again, this relates directly to being prepared for change. Those organizations that are open to change will find that, with phase one fully implemented, there are new processes available and new ways of doing things that eliminate the need for additional configurations that were originally planned.

Of course, this doesn't necessarily mean that everything will be taken care of in phase one, but you may see your list of original requirements start to shrink after go live. Familiarity with the new system after phase one is also beneficial because you and your implementation team will find that it's easier to work in a system that's already up and running. Releasing incremental functionality after you're live will save more time and money than if you converted everything all at once. Plus, you'll be able to stop paying maintenance fees to your outgoing vendor once you turn your new system on. A phased implementation will have you doing that much sooner.

## Avoid Scope Creep

We've all been there—it's halfway through a construction project and, as long as everything is ripped up and exposed, why not add another bathroom or redo the floors. The inclination to add scope to a project is a natural one.

The same goes for membership software implementation. You may have been as detailed in your requirements gathering as possible, but once you get into the implementation, those seemingly "little" requests and add-ons start appearing.

Scope creep is, naturally, one of the biggest areas you can manage to stay within budget. The bulk of this management, however, takes place not when you're in the thick of it but right from the start. A good implementation methodology strongly assumes that spending time upfront will save an organization a considerable amount in both time and money than discovering things toward the middle or end of the project.

To that end, spend time talking to your vendor's implementation team about exactly what you want and need, no matter how exhaustive the RFP is. That's not because the RFP isn't necessarily complete, but because these discussions tend to reveal deeper requirements. Additionally, the implementation team can bring to light the needs and best practices of similar clients. They may be able to suggest a requirement you'll need for processes that will save you in the future.

## It All Comes Down to Change

Staying open to change is your best cost-savings tactic. The less time it takes to go live with streamlined requirements, the less your overall scope will be. And when you define everything you'll need from the start, you'll keep surprise costs from sneaking in. As you and your team open yourselves to new processes, you may find that you can do more with less, leaving you with a system that stays within budget while helping to achieve your goals.

# Establishing the Right Team

A software implementation is one of the most time-consuming projects your team will face and assembling a strong team to play a part in this process will put your organization on a path for success and growth.

The first step is to find your people—those who understand (or will at least listen to) your concerns and who share your goal of getting things right. The implementation team (consisting of both your staff and vendor reps) needs to work together and communicate openly and consistently. Establish a rhythm of communication so that you can work with one another throughout the implementation process by posting questions, assigning tasks, and sharing deliverables. Everyone involved in the implementation should be kept in the loop at all times. With everyone communicating, you're less likely to run into big problems later on and more likely to build a membership system that best meets your needs.

So, what does that team look like?

# The Internal Team

Depending on the size of your organization, it's likely that many of your staff members will be focusing a significant portion of their time toward the implementation process. And, they'll be juggling their day-to-day tasks. That's a lot to accomplish each day!

When assembling your internal team, you want a broad representation from your organization—preferably at least one to two people from each department. **Input from across the organization is essential, because each team and department has different needs, processes, and problems to be addressed. Get input from everyone, and then allow an informed committee to make an educated decision, keeping everyone's issues in mind.**

## Your internal team will include the following:

### Executive Sponsor

The executive sponsor offers support, direction, and guidance to team members at all levels of the organization. This includes assisting with the change management strategy and rolling communication out to the greater organization. This person is the ultimate cheerleader for the project, advocating for continued support and commitment from within, as well as resolving any escalated issues.

### Project Manager

Your internal project manager will work with the project manager on the vendor side to monitor and control budget, schedule, scope, and quality. This role is responsible for addressing resourcing issues and removing obstacles from the overall project, as well as ensuring key milestones are completed.

### Subject Matter Experts

Your subject matter experts are tasked with providing business process input in the early stages. They also author key documentation, provide data and functionality validation, and are key to making sure staff are properly onboarded in preparation for go-live.

### Technical Team

The technical team provides the necessary infrastructure and administrative support, including data conversion assistance and testing.

## Choosing the Team Lead

There is a lot of work involved in examining internal processes, speaking with managers and team leaders, and fostering long-term use of the software. However, everybody needs a head, and the team needs a leader who will be the "final say" when the team can't come to 100 percent agreement on issues.

Should this leader be someone from tech, familiar with software implementation? Or maybe she should be somebody from operations, who knows the ins and outs of the organization. He could be an administrator with all of the authority that the title implies. All of these candidates have the potential to be the right pick, provided that they meet certain other criteria. Here's how to decide.

### Someone Who Is Familiar With Processes Across the Organization

The right membership software will be used across the organization, not just by operations, but also by accounting, administration, and member services. In order to choose a solution that meets all of these varying needs, it's important that they have at least some understanding of the processes, procedures, challenges, and best practices of all of the departments who will be using the software.

### Someone Who Has the Authority to Make Decisions and Achieve Buy-In

Even if they have the know-how, people won't think they have the final say unless they have a position of authority. This might mean assigning a more suitable title to a great mid-level manager, or even turning to one of the executive-level managers. Just choose someone who others will listen to and follow.

### Someone with Knowledge of the Technology, the Business, and the People

Though the considerations of each department's needs are paramount, it's important that your team lead understand three vital elements: they must understand the technology behind the software (to be able to identify and understand best practices and solutions), they must understand the organization's business processes (to track the metrics the business side needs and power their goals into the future), and they need to understand their people (to facilitate buy-in, communicate go-live details, and set up onboarding).

### Someone Who Is Dedicated to the Cause

Finally, the process of implementing membership software is lengthy. You need someone to head it up who has a passion and dedication for the issue, so that they won't give in to fatigue or boredom, or simply toss up their hands when the going gets tough.

### The Vendor Team

When evaluating member management system providers, it's best to take into consideration their experience level with implementations, especially ones that are similar to your organization and the industry that you work in. And, generally, if the providers have that experience, they're able to deliver faster and also provide you more realistic estimates on implementation. They can even re-leverage prior work.

**Exercise some degree of caution when dealing with software vendors that don't implement the software themselves or have control over the underlying platform. If they don't do the implementation, then sometimes it's difficult to determine which party should be held accountable at the end of the day when there are support issues or system issues. It's good to set those guidelines and expectations from the beginning.**

Implementing a new software requires you to "spill your guts" to your vendor and their team. You will have to reveal what you consider to be organizational secrets. And you can expect the vendor to probe you with questions about your processes, procedures, customers, data, and much more. You need to feel comfortable enough to trust them. And of course, the vendor needs to first make make sure they've earned your trust. If you don't, they will not be able to help you get the most out of your software. You need to trust that the partnership you're entering with your vendor is the right fit and accept them as part of your team—they are not only rooting for you, but are working to make your organization successful in the long-run.

## You should expect your vendor team to include the following:

### Executive Sponsor

The vendor's executive sponsor is there to provide support and remove hurdles from the overall progress. This role is vendor counterpart to your internal executive sponsor.

### Project Manager

The vendor project manager operated much the same as the internal PM does—monitoring and controlling the budget, schedule, scope, and quality. Additionally, it is this person's responsibility to listen to your changing needs and make adjustments as necessary.

### Solution Architect

The solution architect leads the early stages of implementation, identifying the design and approach for the project and promoting industry best practices.

### Business Analyst

This position is an asset to the team, as the business analyst examines and documents your business processes and requirements while acting as a liaison between the internal and vendor teams.

### Lead Technical Consultant

The lead technical consultant builds the technical design document based on specifications from the solutions architect and the business analyst. This person creates the effort estimates for work products and has overall responsibility for design, development, and deployment activities.

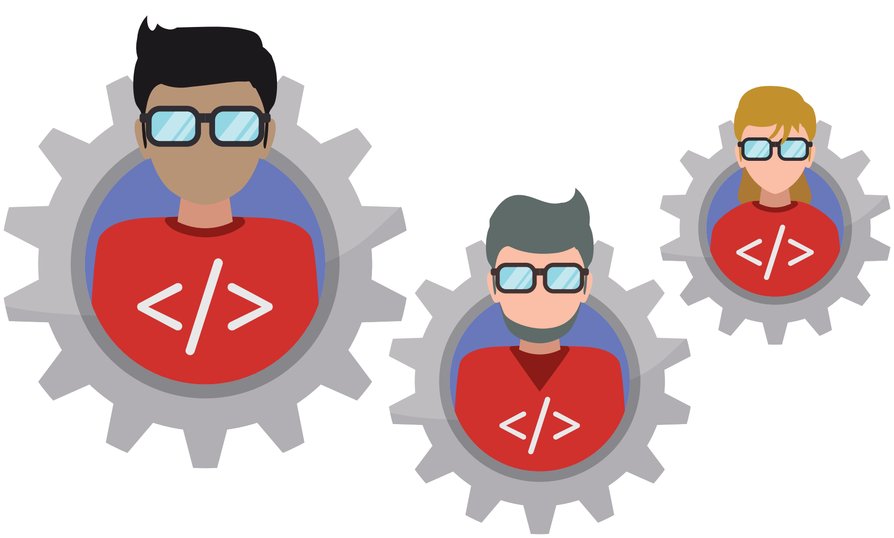
### Other Vendor Personnel

There are many more staff working behind the scenes to support the main players listed here, including those in development, quality assurance, and training.

## Third-Party Implementers

With you and your team having to juggle your day-to-day jobs and the impending implementation process, it can be tempting to consider using a third-party implementation party to assist you. If you opt to do this, be sure to do your research. Make sure they are following best practices in the membership software industry regarding implementation practices and that they aren't endangering your upgrade path. The downside to having a third-party implementer is that they may not follow the software developer's toolkit and might do what they feel is best practices and what they are most familiar with.

However, depending on your organization's needs and your team's bandwidth, it might make the most sense to work with a third-party. They can provide you with a structure to the implementation process and can help you move things along. Just because one organization chooses not to go that route doesn't mean you shouldn't. Research your options thoroughly before you proceed.



# The Importance of Communication

Whether you're dealing with people from inside your organization, those representing your software vendor, or a team from a third-party, open communication and trust are essential to keeping the project running successfully. No matter how efficient your team is, the implementation stage will take time, and you'll be spending a considerable amount of your working life with the team you chose. Being honest and keeping the lines of communication open will make the entire experience much more enjoyable and stress-free. And, who knows, if all goes well, you might even want to spend time with the group after go-live!

# Getting Organizational Buy-In

Change. The word carries a lot of baggage. Some people think of change in quite a positive light: changes come along with the promise of a new beginning and a fresh start. But when it comes down to it, change is hard. Even when people know that the change is for their own good, it's difficult to obtain. Why is that? It's all a matter of basic psychology.

Change in business really isn't all that different from change in one's personal life. After all, no business change occurs without a change in the people. Change management, whether it is supposed to foster acceptance for your new membership software or to get employees used to new work processes, is not unlike any other change: switching to a healthier diet, getting accustomed to a regular exercise routine, or stopping a bad habit that you know is damaging your health.

Most change fails because it is viewed as occurring from the outside in, instead of from the inside out. Organizational change management has to happen the same way: from the inside outward. Successful change management only happens when the fundamental need for change is realized, and that realization is then used as motivational drive to continue with the change. It also involves continual encouragement and support.

When it's time to implement new membership software, you'll need to master the psychology of change management in the same way. Begin by creating the realization of the need to change. Why are the changes necessary? What was wrong with the current system and what will you achieve with the new one? Once you've identified the "whys" behind the change, you can use those reasons to drive the motivation for teams to embrace the new software and push forward with the procedural changes that your new system inevitably will bring.

Finally, you can't forget the crucial elements of support and encouragement. A good change leader will learn to recognize the difference between when to push and when to praise and give the needed prods and props to the right people at the right times.

## Setting Expectations

No matter the number of demos, the length of your RFP, or the completeness of the functional requirement documents your subject matter experts compile with your consulting team, something will always come up. From a major change in scope to a last-minute addition, even with the best communication, expectations are not 100 percent congruent.

That doesn't mean everyone failed at project management, that's just the reality of having multiple people involved.

**While it's actually impossible to expect the unexpected, you and your staff should be prepared for a degree of flexibility in the process. Things come up, changes are made, and people get nervous. It's best to face concerns head on and set the expectation that not everything will happen exactly as planned.**

## Be willing to identify & address doubts

When implementing any new software, you can always expect to meet some resistance from team members. Don't be put off by that. It's completely natural and normal. Over time, people get set in their ways, and when the time comes to change, there is that hesitation of disrupting their job.

Instead of disregarding their concerns, actively listen to them—show them the ROI and how it will streamline processes, specifically ones that impact their roles. Show, don't tell. Sometimes all it takes is an actual demonstration of how much it will improve their job rather than simply stating it in order to open their minds to the idea.

Obstacles can and will occur along the way. And the best way to address it is head-on, but in a positive and supportive way.

### Present the software proposal to the users (the good, the bad, and the ugly)

Vendor presentations are usually reserved for the software selection committee and perhaps a few high-level execs who need to sign off on the expenses. While it's not advisable (in the interest of time and other considerations) to have all of the vendors make their presentations to everyone, it is a good idea for the selected vendor to present their solution to everyone. This does several things: it shows your staff that you are transparent, it explains to them how the new membership management system will affect their daily work processes, and it answers a lot of their questions. Answering questions candidly is the key to alleviating fears and gaining support for the new software.

With any new change, whether it be personal or professional, there is always some fear that comes with it. Once again, it's completely normal, and it's important not to alienate staff members who have those fears. Instead, focus on the positives and enforce the value of training once the implementation takes place. Also, during the implementation, keep the team updated on what's happening. The more they are involved, the more comfortable they'll feel to ask questions, and hopefully in time, that will alleviate fears and give them a sense of optimism.

## Maintain Open Communication

Frequent and open collaboration means that everyone should take advantage of expressing themselves. During the software implementation process, it's important for all parties to clearly articulate their needs and ask as many questions as necessary. As you go, if something comes up that doesn't feel quite right or if something needs to be added, bring it to light immediately and make sure staff feels comfortable doing this as well. It's better to voice your concerns as soon as they arise than wait until the eleventh hour or, worse yet, not voice them at all.

## Celebrate Your Wins

Implementing a robust software is a huge time commitment and organizational priority that should be at the top of everyone's minds. And it's important to take the time to celebrate milestones along the way of the implementation, whether they're big or small.

Incremental milestones are important in a long-term project, when that end goal can seem so far, far away. Maybe you and your staff don't have the budget for blow-out parties at each step along the way, but some pizza, candy, or a late afternoon toast at the end of each implementation phase can go a long way toward easing everyone's stress levels and keeping the focus on small victories.

And along these lines, determine internally which employees are the best to lead the change and the positivity. Having them be "change champions" can help generate buy-in from those who are struggling with the work and the changes. Consider including the implementation in employees' job descriptions for the year the implementation takes place. That way, it's overwhelmingly clear that a successful go-live is everyone's responsibility and should be at the forefront of everyone's minds.

## Keep the End Goal in Mind

There's a reason you changed in the first place and, most likely, that reason is because you're moving toward something better. Yes, it's easy to get caught up in the "but we've always done it this way" mindset, and very few of us relish the opportunity for large-scale change. **When it all starts to feel overwhelming, just imagine where you'll be months after this is over—when what was new and different is now routine and empowering. Someday you really will look back and wonder how you ever got anything done before.** In the end, focus on the good out there that's coming your way. Your new system and what it will do to help you and your members will make it all worth it.

## Maintain the Skillset

In order to do things the "new and improved" way, you first have to come up to speed on how to do them. Learn what you can to arm yourself. Your organization is investing not only in a new system, but in the staff that will be using it. Good training will allow you to do your job better, work more confidently, and onboard other teammates who can follow your processes and procedures.

Everyone in the system is going to be doing something that helps run your organization. If staff members know how to use the system, then they can execute their job better. And you will realize a greater return on the investment in that system by having the people using the system properly. If your users don't know how to use the software, processes can get bogged down. In turn, employees will have a change in attitude and will start to dislike the system. Mistakes will be made. Ultimately, you could end back up where you started.

## The Benefits of Buy-In

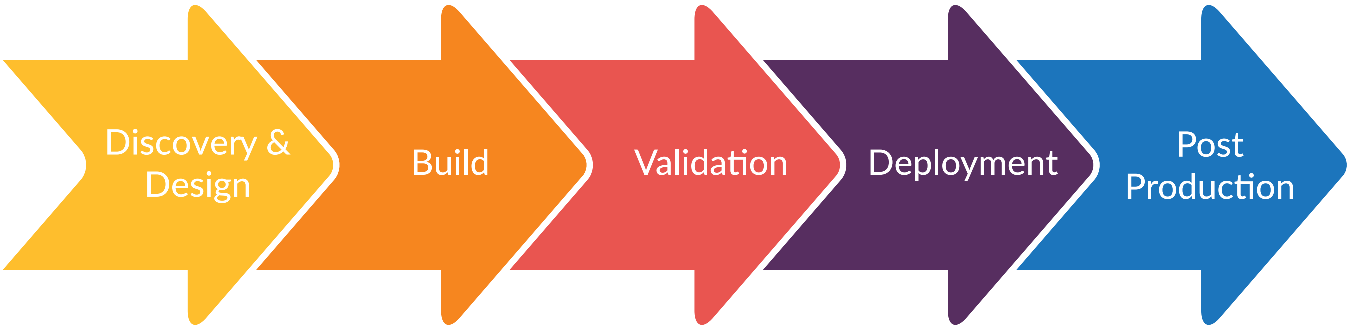
Organizational buy-in isn't crucial to an implementation. It's certainly possible to get a system up and running with just a few people running the process. But the success of the implementation and the longevity of the product being able to do what's necessary are critical to everyone having a stake in the project. Gathering input from staff on their wants and needs, celebrating with them during milestones throughout the process, training them so that they can feel good about the system and their abilities to work on it, and setting realistic expectations at each stage will not only make the end result better for your organization, but can motivate and inform your staff for months and years to come.

# Understanding the Process

So you've budgeted correctly, your team is in place, and you have an entire staff energized about the new system and ready to celebrate. What's next?

## The actual implementation.

This is where the rubber meets the road and the work begins. Here's what you can expect to happen over the next several months.



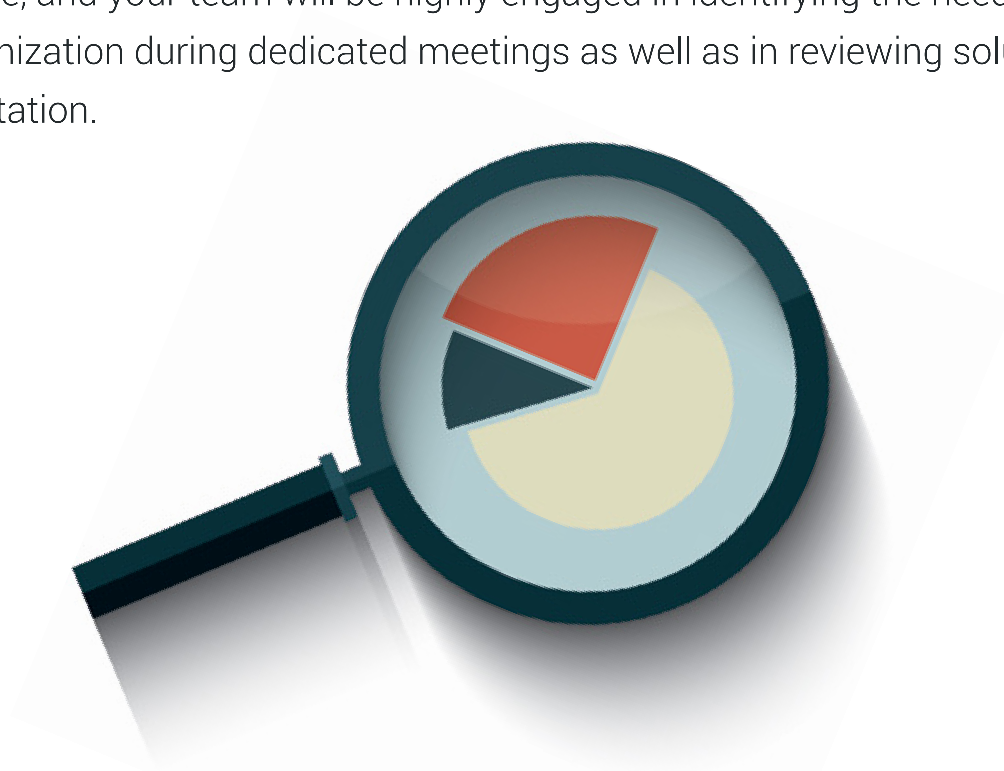
# The 5 Phases of Implementation

## Implementation typically has five phases:

- Discovery and Design (this also is known as Solution Design)
- Build
- Validation
- Deployment
- Post-Production

## 1. Discovery and Design

Before you can begin building your membership system, it is essential that you are able to identify the functional requirements needed to run your team. If these requirements are not known prior to implementation, then your organization will begin at the Discovery and Design phase. This phase lasts about three months on average, and your team will be highly engaged in identifying the needs of your organization during dedicated meetings as well as in reviewing solution documentation.



**Here's a quick breakdown of the high-level tasks to be completed in Discovery and Design:**

- Conduct discovery meetings with the team to create a Functional Requirements Document (FRD) outlining your business requirements for this project.
- Creation of a Solution Design Document by the vendor based on the approved FRD that describes how to leverage the software to meet your requirements.
- Plan for the data-conversion process.
- Identify possible risks that could slow down the project.
- Work with your vendor to create an implementation plan that will help determine the key deliverables and time commitment from both teams.

### Solution Design

If it is unnecessary for your organization to begin at the Discovery phase—because you've already determined the functional requirements needed for this project—then you can instead begin at the Solution Design phase. The Solution Design phase lasts about two months on average and requires a medium level of commitment from your team. It involves gathering information about your current business processes, developing solution-design documentation, participating in data-conversion planning, and determining technical needs and build specifications.

Solution Design is an important phase of the project, as this is where you will define how you want your system to function to meet your process needs. In this phase, you will deliberate with your subject matter experts, attend meetings, clearly define the scope, and provide timely feedback regarding all the documents produced. Your feedback on the solution-design document will help the developing team better meet your requirements so they can build a system that meets your needs.

During the Solution Design phase, you will also be working on an implementation plan. Your vendor will work with your team to determine how to best proceed with the design project, preparing you to begin the Build phase of implementation.

## 2. Build

Build is typically the largest phase of the project, averaging about four months, and it is the phase during which the developing and quality assurance team will build, test, and demonstrate system configurations for your team. The overall goal for the Build phase is to produce high-quality deliverables that match the exact specifications laid out in the Design phase. While you will not be building the system, it's essential for your subject matter experts and technical team to be engaged in this phase as well, reviewing developed application functionality and soliciting feedback.

## 3. Validation

You will be deeply involved in the Validation phase, as it lasts three months on average. The Validation phase is the testing portion of the project and takes place after either a portion of or your entire system has been delivered. This phase is also commonly referred to as User Acceptance Testing or UAT. The objective of this phase is to ensure that the system is accurate and meets the requirements documented in the beginning of implementation. Any problems that occur during testing will be fixed before the next phase begins. Once you've approved the system, the project will move on to the Deployment phase of implementation in preparation for go-live.

**Three types of testing will take place:**

1. Performance testing: The portions of the system that are determined to have heavy user load will be tested to ensure it is performing as it should.

2. System Integration Testing: Once the entire solution is built, the system is tested to ensure that components are working together properly.

3. Regression: Verify that the existing system and any remaining functionality that will be integrated with the membership software is working as expected.

During the Validation phase, your team will also receive end-user and developer training prior to the go-live, ensuring a smooth handoff.

## 4. Deployment

Once the system has been approved in the Validation phase, you will move on from the development and testing stages and enter the Deployment phase. This phase typically lasts two to three months and involves launching the system into production. To ensure a smooth transition, you'll receive help from training staff, create a successful go-live plan, perform tests, and set up a mock go-live before the system actually goes into production.

**After the mock go-live has been completed, you will begin the go-live process, which includes:**

- Transferring the system from the testing/staging environments to the production environment.

- Performing the last data conversion from the legacy system.
- Performing data validation to ensure the data conversion was successful without contaminating the data that was transferred.

### 5. Post-Production

The final stage involves hands-on support so you can quickly address any issues that may arise after the system is live. Your technical team will work with your vendor's support team for system trouble-shooting and end-user support to ensure the system is running smoothly for everyone.

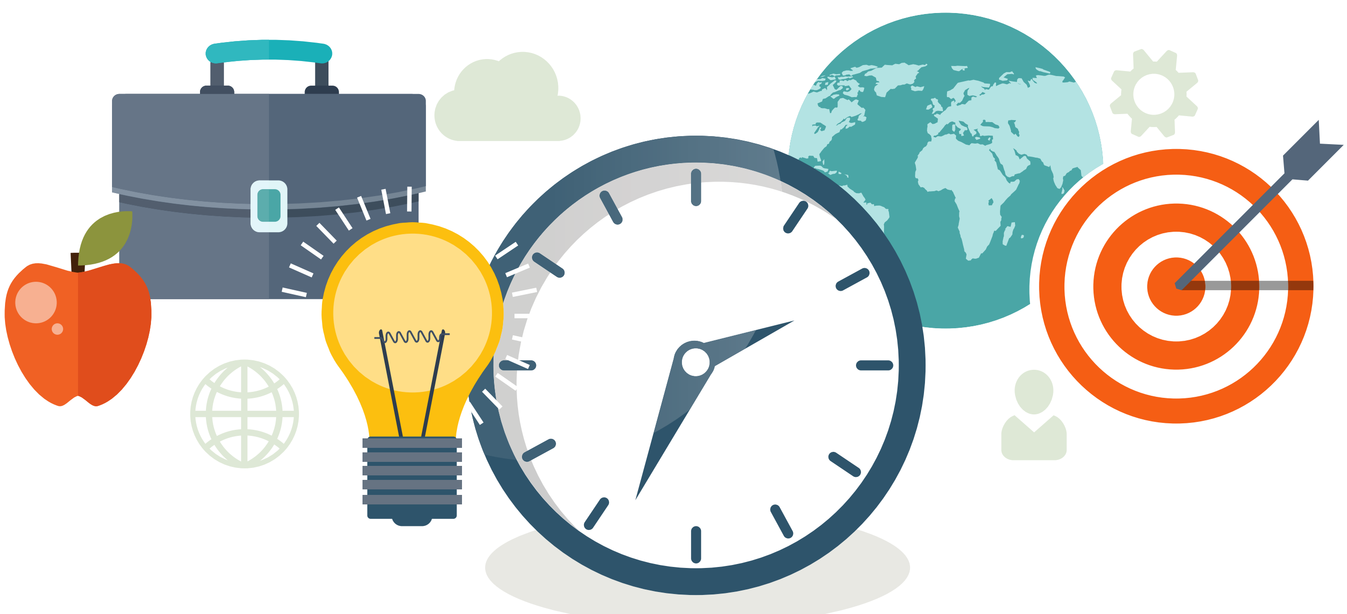
The time commitment required in the post-production phase depends on the amount of testing done prior to the go-live. More comprehensive testing most likely means that fewer issues will arise and therefore a smaller time commitment is required. On average, you will need about two to six months to settle into the new system and complete this last phase.

## Time Commitment

Every organization is different, which means the duration of an implementation project largely depends on each individual client. Your specific schedule will be largely determined by the amount of data you'll be converting and the functional requirements produced in the initial phase. You'll work with your software vendor in the Discovery and Design phase to create an estimated timeline. Implementation projects average about 11 to 12 months, but one can last over two years, depending on requirements. Keep in mind that your estimated timeline is simply that: **an estimation.**

Each phases requires different level of commitment from your team. Here's a quick breakdown depicting the amount of time you can expect to dedicate during each phase of the project:

- **Design:** 2-3 months
- **Build:** 4 months
- **Validation:** 3 months
- **Deploy:** 2 months
- **Post-production:** 2 months



Greater commitment is often needed during the initial and final phases. Throughout the project, expect your team to have the following commitment levels:

- **Discovery and Design:** Medium to high commitment
- **Build:** Low commitment
- **Validation:** Medium commitment
- **Deployment:** Medium to high commitment
- **Post-Production:** Low to medium commitment

# Establish a Rhythm

As we've said several times already, good communication is essential to a successful implementation project. Everyone involved in this project should be communicating with one another and understand what the goals are, what decisions are being made (or need to be made), and which tasks need to be completed. Throughout each phase, two recurring meetings will take place:

- **Project meetings:** These meetings usually occur weekly and involve the project managers and subject matter experts from both the internal and vendor teams. During these meetings, you'll discuss the status of the project, what has been accomplished, and what is yet to come. This is a good way for both teams to check in with one another and ensure everything is running smoothly.

- **Executive oversight meetings:** The executive sponsors in both your organization and the vendor team will meet each quarter to discuss the project's progress and what issues need to be dealt with. Once again, these meetings are meant to ensure any problems that are occurring are solved, and that the overall progress of this process is running as close to plan as possible.

# Data conversion & cleansing

Typically, the most time-consuming task in an implementation project is the process of data conversion—it is also crucial to your implementation success. To cut down on expenses, many organizations are tempted to tackle data conversion in-house, but this is a risky move. Even the most expert in-house IT department won't have the experience with the new system like your software vendor has. After all, the vendor team is made up of experts in the new system and can (and will) save time, frustration, and money in the long run when you turn the data migration process over to them. That said, your IT team will be heavily involved. They're the experts in the current system and know the data intimately, so they absolutely need to be active in the process. But the ultimate responsibility for making sure data conversion happens and goes through successfully should fall to the vendor team working the implementation.

Data conversion starts as early as the sales process, where you'll start talking about the amount of data moving over and how to get prepared for the process. You'll identify the data sources that you will need to convert to the new system, including data from an existing membership management systems, spreadsheets, and any other systems of record.

Then data is mapped into the new system. This is done through a process of a lot of small iterations, where little bits of data are moved into the new system. This is also a good time to get the users involved so they can start seeing the data in the new system and start testing it. This helps with user buy-in to the new system and gives staff the opportunity to keep the teams updated if something is missing or if everything's migrating over correctly. Once it's working, the next data set is pulled over. This occurs throughout the implementation, and the process can vary from running data conversion a dozen times to running it almost weekly over the course of a year. The data conversion process runs from start to finish. An organization will run data conversion one final time during go-live to ensure all data is up to date during the last weekend before the system fully launches.

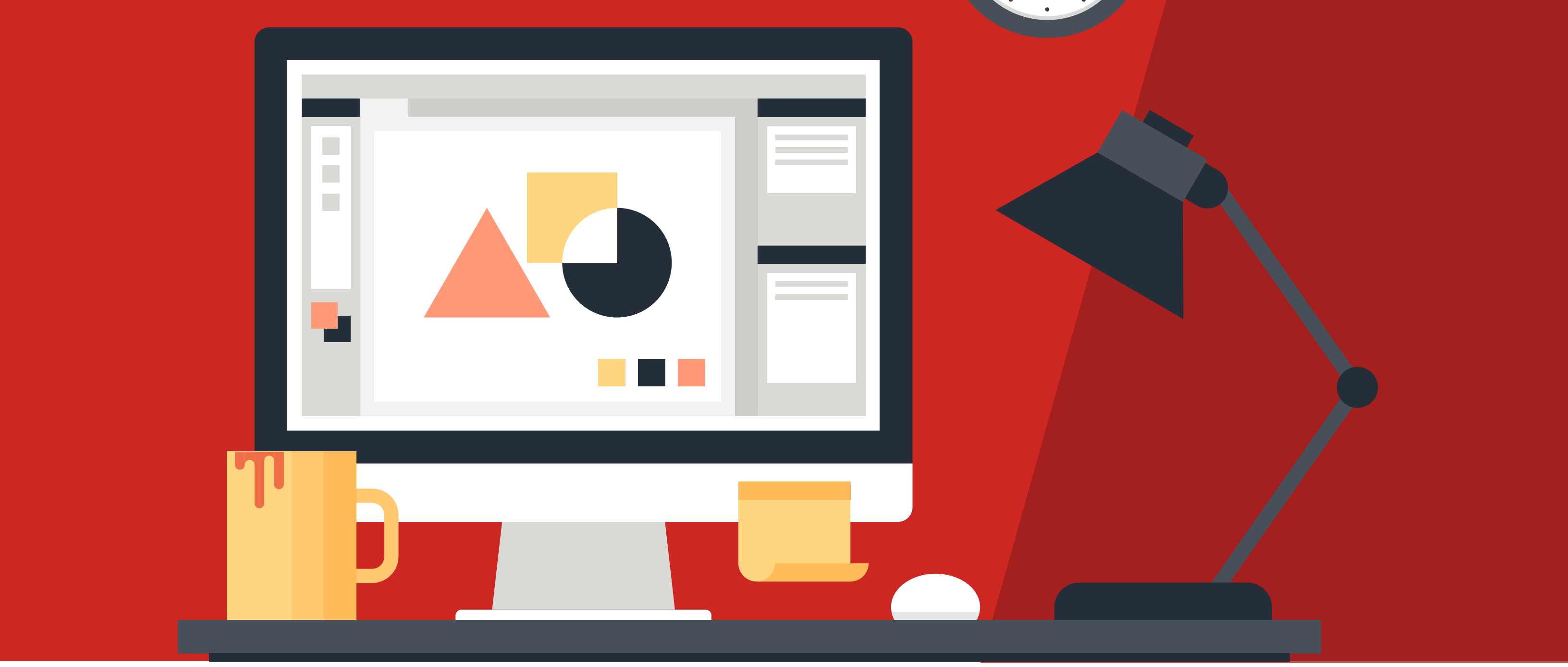
**The data you'll be converting often falls into three categories:**

- Types and Codes: Includes values in drop-downs and various codes in the system.
- Master Data: Data such as People, Companies, and Employees that exists only once in the system; transactions are used repeatedly.
- Transactional Data: Data describing business activity, such as Orders and Subscriptions.

Duplicate records or inconsistent data can cause issues during the conversion process, which is why data cleansing is a necessary step during data conversion.

For much more on the data conversion process, check out our data conversion FAQs eBook, **Demystifying Membership Software Data Conversion.**

A membership management system can streamline business processes, consolidate your data, and give you better insight into your organization and its membership base. But here's something to keep in mind, although it does help you manage your organization, you and your team still are still responsible for maintaining what's inside it. Data plays a big part in understanding your membership better, but any system is only as good as the data entered into it. Be sure to establish a data cleansing plan so that the data you enter is correct, complete, and not duplicated. Then, put steps into place to assure that the data remains accurate after the system is up and running.



# The End of the Beginning

Even after the implementation phase is over, your team should continue the close relationship it's built with your vendor. Ask questions and seek solutions from support staff when needed.

Once you've reached the final stage, take some time to celebrate the completion of a long project with everyone involved. Implementation isn't easy, and your team has worked hard for months to ensure the process goes smoothly. Plus, now that the implementation process is over, you can finally start enjoying the rewards of the new system, managing your organization from an intuitive, efficient platform, and focusing less on the system itself and more on serving your members.