

# How to Successfully Migrate Your MarTech in 4 Phases

A Guide for Successfully Switching Web and App-Based Marketing and Analytics Solutions



Whether you're moving from one tag management system to another, switching web analytics solutions, or adopting an alternative marketing technology for your website, it's important to know what steps will lead to a successful migration for any MarTech implementation.

This white paper presents a 4-phased approach to completing your migration successfully:

- 1 | Strategize
- 2 | Document
- 3 | Migrate
- 4 | Test



# Phase 1

## Strategize

---

Before you begin your MarTech migration, you will want to have a detailed strategy in place. This will ensure that each member of your team understands the process and is able to stay on track to complete the project on time. Here are a few areas to consider in your strategy:

### Get your boss's blessing

A technology migration can take a lot of time and resources, and if you have a manager who doesn't fully understand the value of optimizing your MarTech solutions, accurately communicating the importance of this migration is the first step to ensuring a successful transition. Consequently, you might need to put together a business case with the following details to convince managers that the time and resources spent on this project are worth it.

### Document the business objectives that this migration is helping to reach

You will likely have already made your decision and justified it, but it's best to have a record of the reasoning behind your migration so you can map your efforts back to your initial goals and keep your team focused.

### Identify who will be involved in the migration and what they will contribute

Be sure to specifically outline the roles and expectations of each team member, as well as the estimated amount of time that will be required of each employee. This is especially important if you have to get help from other teams.

### Decide how you are going to document, migrate, and test.

As you flesh out the details of how you will carry out your migration, here are some things to consider:

- Do you have any outdated, unused, or unauthorized tags you should update or sunset?
- Should you update naming conventions to be more consistent or easily understood?
- Do you need to update your data layer?

In some cases, you may decide to start from scratch with your implementation. If you do so, you will be able to resolve old errors and upgrade to new methodologies, but be prepared to spend a significant amount of time making the migration.

You will likely find some skeletons in the closet and unexpected challenges you will need to address (more on this in the Migrate phase). However, at the tail end of the process, your implementation might be much better off.

## QA Testing:

One of the most important parts of this phase is to plan how you're going to test your implementation to make sure everything is deployed correctly and functioning as expected.

Manual testing is possible using tools like **tag debuggers**, Charles, Fiddler, or the developer tools in your browser (more on manual testing in the Test section), but these options are time-intensive and prone to human error.

Automated testing is much quicker and more reliable. An automated testing solution like **ObservePoint** enables you to audit your tagging implementation before, during, and after your migration to ensure it was accurately deployed according to your expectations.

## Compile a projected timeline for the migration.

You can use the four phases in this white paper, supplemented by specifics of your own situation, to structure a project timeline that will keep your team on track and help others be aware of upcoming changes.

# Phase 2 Document

---

The goal of a successful migration from one solution to another is, at a minimum, to maintain the status quo while making the desired improvements. To do this, you need to establish a baseline (a snapshot or some form of documentation of your current tagging implementation) to check your updates against. You can create this documentation manually or via automation.

## Manually

If you're approaching the Document phase manually, you will need to visit your tagging plan or solution design reference (or create one if you have not already) and ensure that it is accurate and up to date.

Manually creating and maintaining a **tagging plan** is an involved process, so be sure to factor in a significant amount of time for crawling through your site code page by page to thoroughly document what you have deployed.

If time isn't on your side, or if you're aiming for a higher degree of accuracy, automation can speed up the process and eliminate inaccuracies in your documentation.



## Via Automation

As an alternative to manually documenting the technologies on your site, you can use an automated solution like ObservePoint. ObservePoint's software can conduct a comprehensive scan of your website and create a detailed record of all technologies deployed on it.

ObservePoint's scans (**Audits**) quickly create an up-to-date catalog of your current implementation to compare against your new one. As a result, you can migrate using a more streamlined approach, such as lift and shift (more on this under the Migrate stage).

Here is an example report for a single page:

Page Information		Tags	Cookies	Request Log	Console Log	Tag Initiators	Rules
Total Tag Requests	27	# of Unique Tags	12	# of Broken Tags	---		
				TAG STATUS	REQUESTS	CODES	
>	Facebook Events JS Bootstrap	Social Media	●		1	↓	Accounts: 1
∨	Google Ads Conversion Tracking	Advertising	●		1	↓	Accounts: 1
OBSERVEPOINT DATA		UNIQUE REQUEST ⓘ					
Account	947068447						
Status Code	200						
Tag size (bytes)	370						
Load Time	272						
Duplicate Request	0						
Multiple Request	0						
VARIABLE DATA							
async	1						
bg	ffffff						

## Benefits of Using Automated Audits

### Speed

Automated scanning software can audit and catalog much faster than any human—at least 5 times faster for a scan of 100 pages. In all likelihood, you will have to scan more than 100 pages, in which case automation is even faster.

### Currency

Documentation of your tagging plan quickly becomes outdated, so if you have to make any updates during your migration, you will also have to update your documentation. Automated solutions, however, can run scans frequently, giving you an up-to-date catalog of what you have on your site.

### Accuracy

If tasked with manually scanning through an implementation and cataloging all technologies on a site, a human is bound to make mistakes. Automated solutions minimize errors by carrying out a repeatable process that doesn't miss technologies.



# Phase 3

## Migrate

---

Depending on which software you are migrating to and from, there may be a functionality in place that allows you to quickly migrate your site at the touch of a button. However, these solutions are frequently ineffective, prone to error, and do not exist for every software. As a result, most teams choose to migrate their sites by implementing their new technology in a staging environment (using either a pre-production server or the staging capabilities of a tag management system) before they go live.

### Understanding the Development, Staging, Production Model

In the context of MarTech migrations, this model allows you to test implementations before going live and risking public malfunctions that interrupt data collection or customer experiences. These are different environments along the web or app development process, and each environment is used as a testing ground to prepare for the next one:

**Development Environment** is your initial experimentation environment where you can build the minimum viable product of your website or app.

**Staging Environment** is where you prepare properties to be seen by the public. You should conduct major testing and adjustments here before going live.

**Production Environment** is the live version of your site or app, and is completely accessible to the public. Any updates to your production environment should be tested in the staging environment first to ensure functionality.



As you get ready to migrate your technologies over to the new version of your site, you can choose to implement one of two strategies: fresh start or lift and shift. These strategies follow similar processes but with some important differences.

### Fresh Start

The fresh start approach involves removing the current tagging on your site and building a brand new implementation. To successfully migrate with this approach, you will need to:

- Rewrite your entire documentation in your TMS and create a new tagging strategy.
- Test your new site in a staging environment, either manually or by leveraging automation, to check that everything is working the way you want it to.
- Push your site through to production and run another round of tests (more on the specifics of testing in the next section).

Migrating with a fresh start approach gets the job done, but it is a hands-on, time-intensive approach that requires manually duplicating a lot of your previous work.

### Lift and Shift

This strategy is significantly faster but is only possible using automation. With the lift and shift approach, you use an automated solution to essentially “copy and paste” your current documentation, so you can transfer all your data from one

location to another without having to re-architect anything. To implement this strategy, you would:

- Use an automated tool to scan and document the current implementation.
- Duplicate this implementation to the staging environment and run a scan to test that everything is set up properly.
- Push your site live and run another scan to test that everything is deploying correctly.
- Set up automated audits to regularly test your implementation going forward.

Using an automated solution to migrate with the lift and shift approach allows you to leverage the work you have already done on your site and feel confident that all the technologies on your current site will be accurately replicated on your newly migrated site.



# Phase 4

## Test

### When To Test

As mentioned in the previous section, ideally, you will carry out tests in your development and staging environments before pushing live to production. Doing so will make finding errors easier and relieve the stress of pushing your new implementation into production, where errors would be public and could impact customer experiences and revenue opportunities.

From there, you should perform incremental tests every time you manipulate your tags, whether during the migration or for ongoing validation.

### What To Test

The essence of testing is comparison, whether you're comparing the new implementation against your documentation plan or your production environment. When testing, you will want to prioritize the most important parts of your site relative to the marketing/analytics technology you are deploying.

Some of the most important sections of your site to test during a migration are:

**Critical Conversion Paths** - If your site has a booking, shopping, login, or other important conversion path influenced by the MarTech of concern, focus your testing on those paths. Conversion paths are action-oriented, as are the tag triggers on these paths, so you will want to replicate the actions of each conversion path under as many circumstances as possible to ensure they work.

**Top Pages** - Test your most important pages where the MarTech is deployed to make sure it's working. For high-traffic pages, having a tag go down for even a short window can have a significant effect on data collection and potential revenue opportunities.

**Data Layer** - Many analytics and marketing solutions rely on the data layer object. As you test, verify the data in your data layer matches your standard formatting conventions to ensure the right information is being collected after your migration.

### How to Test

#### Manually

If you decide to test manually, be sure to carve out a significant amount of time for testing your implementation as it moves through the development, staging, and production environments. To successfully execute manual testing, you will need to design testing plans that specify which tags should exist on

which pages, what variables should be collected, and how different click events and conversions should be tracked. Then, you will need to write out your test designs and begin crawling through your site code, page by page, to check for errors.

### Via the MarTech's solutions

Some MarTech solutions include testing capacities that will test your migrated site for you and let you know if everything is implemented correctly. This functionality is not often available, so you will have to check with the specific technology to see if this service is provided. Even if it is, you may want to consider using a more intensive approach to your testing, as the self-testing capabilities of these solutions are typically limited and faulty. Accurate testing is a difficult, complicated beast, so an automated, enterprise-level solution designed specifically for testing is usually the best way to go.

### Via Automation

**ObservePoint's automated data governance solution** allows you to audit implementations with rule-based testing that reports when the data being collected doesn't match expectations or when tags aren't present. Using ObservePoint, you can apply existing portfolios of tests or generate new tests in order to verify new implementation is functioning as expected.

## Key migration testing features include:

### ✓ Journeys

Replicate and test critical customer journey actions, like clicking, editing a text field, checking a box, and more, while also testing for expected tags to fire with each action. Journeys ensure customer experiences are operating smoothly on your migrated site.

### ✓ Journey Comparisons

View run over run insights into what has changed in your tagging between user journey runs to ensure critical customer journey data is being tracked both after migration and after each future release.

### ✓ Remote File Map (RFM)

Locally test new scripts (like tag management scripts) in place of old scripts in a production environment to verify tags are firing properly before deploying new tags. Overall, RFM is extremely beneficial any time you're replacing JavaScript libraries.

### ✓ Audits

Audit your migrated implementation in development, staging, and once you push to production to verify that all tags are deployed and firing as expected. Then run scheduled, automated scans of your top pages going forward to make sure everything stays in place.





## ✔ Tag Hierarchy

In the case of tag management migrations, this enables users to determine which tags are deployed outside their TMS and migrate those into their new TMS.

## ✔ Data Layer Validation

ObservePoint's audits will identify which variables are being passed to your data layer so that you can validate all necessary data collected upon migration.

ObservePoint's automated solution makes it possible for you to easily verify that tags are deployed correctly throughout the migration and firing as expected thereafter. And you can also use ObservePoint's manual solution, **TagDebugger**, to spot-check individual pages as needed.

ObservePoint is here to help you make it to the post-migration phase as quickly as possible.

To learn more about how ObservePoint's automated testing and governance technology can simplify your migration project and beyond, schedule a demo.

[SCHEDULE DEMO](#)

# Post Migration The Finish Line

---

After your final testing phase, your Mar-Tech migration will be complete, and you can congratulate yourself on a job well-done.

