Migrating to Liferay DXP:

Best Practices, High-Level Approach and Strategy

Liferay

Table of Contents

Executive Summary

A migration to Liferay Digital Experience Platform (DXP) is one of the most important steps an organization can make when transitioning from a legacy technology. The level of effort can range greatly, from a simple task that merely requires running an existing plugin to a fairly involved and complex development project requiring many man hours and resources. The question is: How do you know the level of effort required for your specific migration project? The answer depends on many factors. In this paper we will survey high-level methodologies. First we will discuss the best way to assess the scope and effort required for migration, then we will explore specific methodologies that have worked best for Liferay customers as well as ways to sketch a serviceable project timeline. We will end by highlighting some factors that affect the level of effort involved in a migration. The goal of this paper is to help organizations assess their own resources at-hand and come up with a plan for their migration project.

How do I go from my current system to Liferay?

Liferay DXP has an open framework and API, which can be used to leverage any data that you may have in your current system. The most common method used to migrate content from one system to Liferay DXP is to create a Liferay plugin that reads data from the legacy system and uses that data as parameters to call the Liferay API. For example, if there is data in a legacy CMS and the content needs to be migrated, the Liferay plugin would read that CMS content data either by directly accessing the database or calling that CMS's API or services, and then would iterate through that data and call the Liferay CMS API until all of the data is migrated over. The same method would be used for all other content from users, roles, documents and any other assets that can translate over to Liferay DXP. The key questions are: 1) Can I use an API or web services to get to the legacy system data, or do I need direct access to the database or an export format? and 2) Is there already a Liferay Plugin made for the specific system that I want to migrate from?

There are no official Liferay plugins available in Liferay Marketplace that automatically migrate content from one system to another. However, Liferay Community members have created plugins and tools for migration, and Liferay Global Services (Liferay consultants) has access to Liferay customer-specific tools to migrate data from a legacy system to Liferay DXP. Over the years, Liferay has helped customers migrate from systems such as Wordpress, Jive, Sun Web Space and Oracle WebCenter.

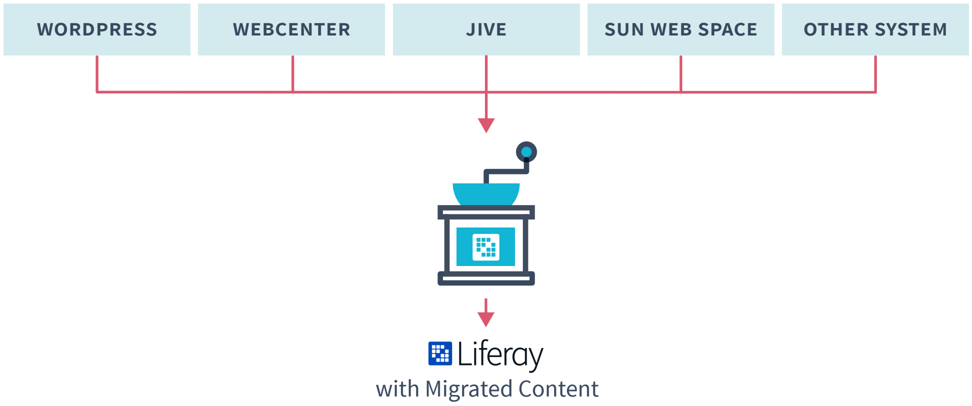


Figure 1

Questionnaire

Evaluating the level of effort that will go into your migration is a key step in preparing for a successful migration. The following are recommended questions for companies to ask in order to better understand the scope of the migration, pinpoint the areas that will require a team's attention moving forward, and bring to light important issues to consider.

Is there a Liferay migration plugin for my current system?

The best-case scenario for a migration is that a Liferay plugin already exists for your current system. But if you find an official plugin from Liferay is not available, a third-party developer may have built a plugin that fits your criteria or the Liferay Global Services team may be able to tailor a plugin to your exact specifications. Customers vary in their unique needs — for example, in choosing to include or exclude different metadata associated with their content. Customer A may want to migrate all their content from Jive, whereas Customer B may want all their content migrated excluding their Blog content. In another case, Company A and Company B both want to migrate Blog content, but Company A wants all the metadata for every Blog post, whereas Company B only requires a subset of the entire metadata.

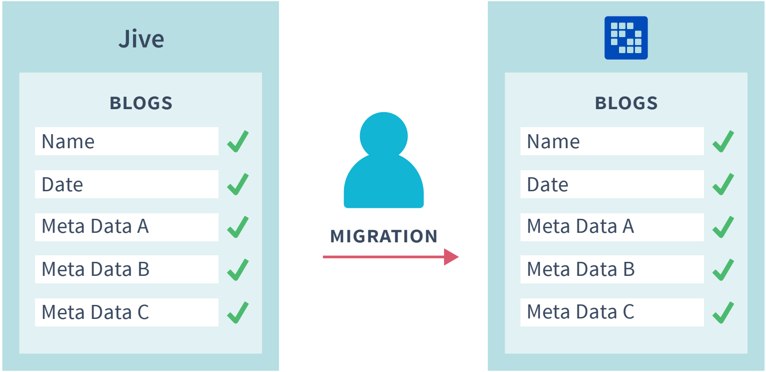


Figure 2

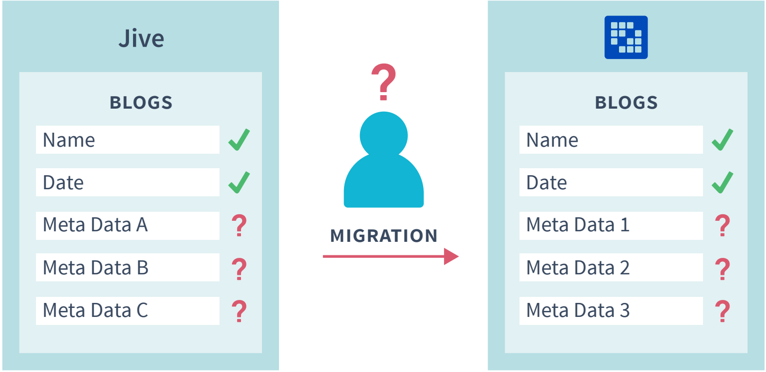


Figure 3

Is there an API, services or export format/media for my current system?

If there is no plugin available to run the migration right away, then a plugin will need to be developed to call any existing APIs, services or export formats.

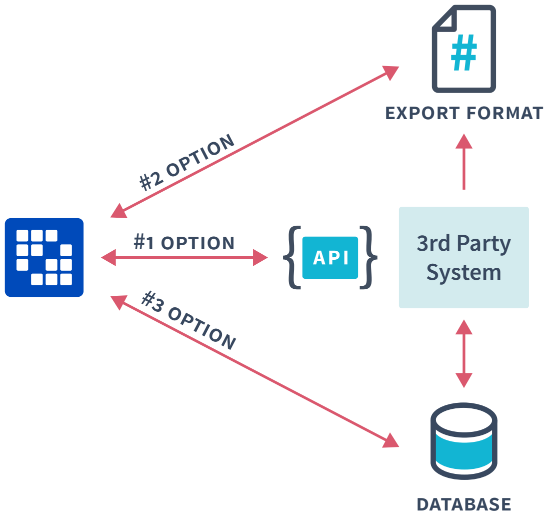


Figure 4

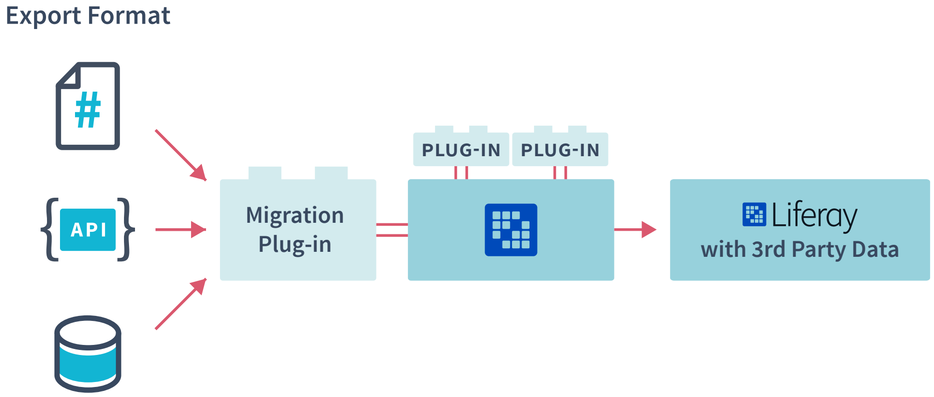


Figure 5

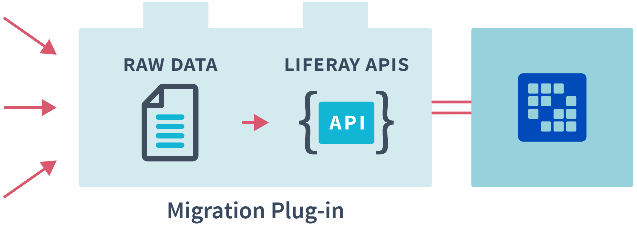


Figure 6

Java or non-java?

Because Liferay runs on Java, migrating from another Java technology may provide more avenues for interoperability. Though this may not greatly affect the migration of data from one system to Liferay due to the presence of APIs or other interfaces, your organization may explore the possibility of implementing Java technologies, such as WSRP, or migrating an existing Java web application from that product to Liferay DXP. These options would not necessarily be available for non-Java technologies.

Can I scrape web pages for HTML content?

If the content that needs to be migrated resides on a legacy system and is not too complicated (e.g., plain web content versus a web application with various forms and screens), you may be able to develop a screen scraper app. The app will be able to dynamically populate Liferay at runtime or as part of the migration process in a one-time task. It is worth noting that transforming/cleaning up HTML that you harvest this way is recommended so you can take full advantage of Liferay themes. You could be dealing with countless issues where you've used cut-and-paste from an old site (or Word documents) and wound up unintentionally embedding tons of inline CSS that prevents the theme from presenting a consistent look and feel.

Do I have access to my current system's database?

If there is no API, no (web) services and no export format available from the legacy system, then your only option may be to access the legacy data through the database. This method will require time-consuming but necessary research and discovery. Since the database will be raw data, making sure the correct data is being accessed and imported is the most important step you must take.

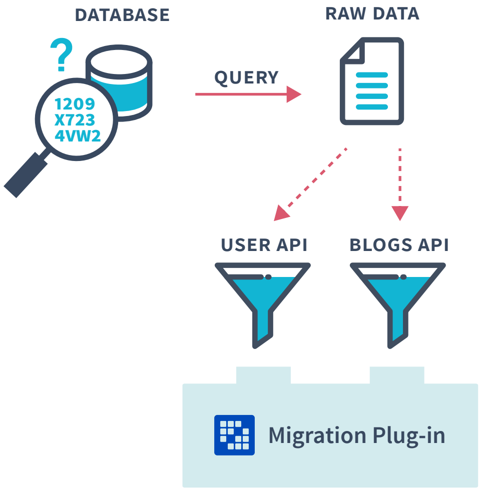


Figure 7

How much and what parts of my current data do I want to keep?

Take an inventory analysis:

- CMS
- Blogs
- Wikis
- Message Board/Forums
- Application data
- Other web content

Will I have any downtime or content freeze period?

Allowing for a scheduled downtime or a content freeze window will make planning easier and help minimize any data corruption or inconsistencies between the legacy system and new Liferay system. If downtime is not a possibility, your team will need to test and optimize your migration to be as fast as possible, and possibly account for any data gaps that occur during the actual migration task.

Do I need to integrate with an identity management system?

If you have a separate identity management system or server, your team will need to test against a copy of the production identities before deployment. This is especially true if the identity stores differ between your testing environment and production. If this is not possible, UAT testing will need to include tasks for testing against the production identities. At any rate, accounting for this will require appropriate tasks during testing.

Methodology

The methods by which you extract data from the legacy or third-party system vary, and are dependent upon what resources, if any, the software vendor has explicitly made available. Below, we will explore the most common methods of data extraction and the method of importing that data into Liferay DXP.

A. Export Formats

Software Vendors may provide their own proprietary export format or a standardized format, such as an XML file. Sometimes, it is a combination of both. Liferay DXP exports site content in LAR (Liferay Archive) files, which contain XML files and a structure of files and directories that have objects such as documents from the document repository. Liferay recommends going with a standard format as opposed to a non-standard one; if the software vendor's format is consistent from version to version, it may not matter. It is important to find out if there is any documentation on the export format. If there is no documentation, then the data discovery and data mapping exercise for this method will require a considerable time investment. The accuracy of data mapping will be crucial for successful migration.

B. Third-Party APIs

The product data may be accessible via APIs. Web services are a common example of how data is exposed to third parties. It is quite common to find official documentation on what sort of data the APIs return and thus save time on the data discovery time investment. The data mapping will still need to be done, however.

C. Database Access

Examining the raw database is probably the most laborious and undesired method of data extraction. If the data to extract is in database and tables, with no available documentation, then the time and effort required to match the appropriate columns and tables to the objects and content will be formidable and the task will be quite tedious. With older systems in particular, there is no alternative choice.

D. Data Mapping and Calling Liferay APIs

The data mapping exercise is the most crucial and time consuming stage of data migration. Whether it be from a raw database, or from APIs that return documented datasets, developers still need to get it right. For example: although a legacy system and Liferay DXP both have blog features, each piece of metadata must be carefully organized and mapped into its corresponding Liferay metadata. For a successful transfer, every item in your Inventory Analysis will need to be curated carefully to ensure no data is lost.

E. Filling in the Gaps

You may notice gaps in the legacy system or in Liferay during the data migration process. Two common problems may arise:

1. The legacy system may have data that does not exist in Liferay.

For this situation, you must decide how valuable the data is. If it is valuable, then it can be mapped to an existing Liferay metadata field that was not immediately obvious, or Liferay custom fields can be used to create metadata for this legacy data.

2. Liferay DXP APIs require parameters and metadata that do not exist in the legacy system.

This is often an easier situation than (1) above. Developers can work with business stakeholder to decide how to generate the data, or with what to default the data. This will depend on the type of data that is missing (i.e., time stamp vs. screen name).

F. Choosing the Best Method

While Liferay does not designate one approach as better than the other, the rule of thumb is that the Liferay plugin that processes the legacy data and subsequently calls the Liferay APIs should be portable and maintainable from version to version. Investing more time and money into this early stage will positively impact the amount of time and effort required in later stages.

G. Tools Required

- JDK
- Database client tool
- Liferay DXP
- Liferay IDE, Liferay Developer Studio (LDS) or other development environment

Timeline

It is important to include all of the necessary steps in a timeline so you can make a reasonable estimation of when to complete a migration project. A proper migration will include several tasks other than the actual migration of the data itself. These are the items to consider.

A. Discovery and Gap Analysis

Not all web frameworks are the same: what exists in other products may not exist at all in Liferay, and vice versa. If a data field or metadata only exists in the product you are migrating from, you may have to decide how to save that data in Liferay or whether you even want to keep it. Similarly, if a Liferay entity needs particular data and there is no one-to-one mapping from the older product, you may have to put the data together from disparate pieces. In the Discovery and Gap Analysis phase, your team's goal is to define the nuts and bolts of the dataset in a Liferay deployment. Often this can simply be done by deciding which entity

APIs need to be called, which parameters each call requires and what the legacy systems can provide.

It’s important to point out here that data quality in the old system can be a serious issue. In talking about building the new data model, we should document what can be eliminated from the old system. There may be whole classes of users, pages or custom entities that are no longer relevant and should be stripped out in some way.

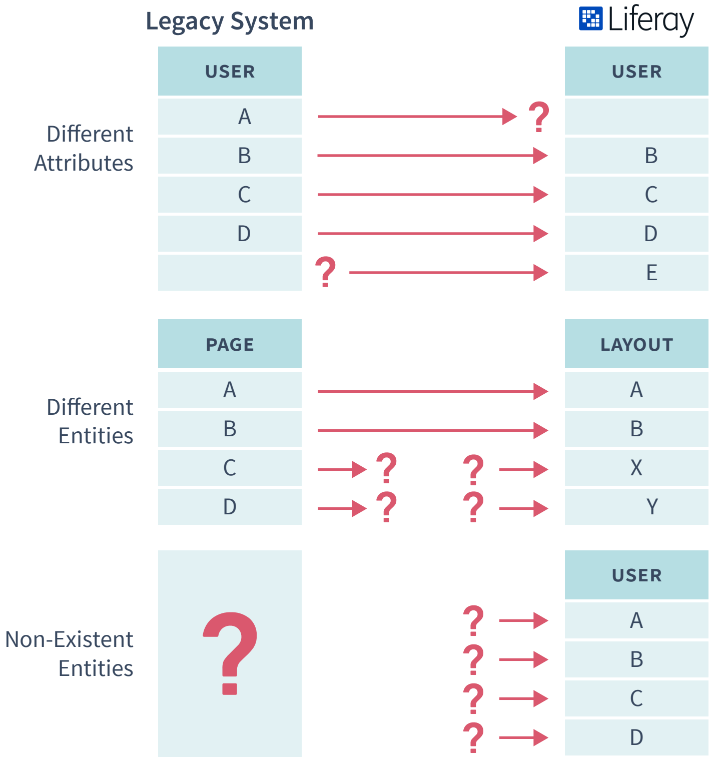


Figure 8

1. Permissions and Roles

Liferay DXP has a Roles-Based-Access-Control system. The Roles and Permissions in Liferay is fine-grained and particular care must be given when assigning roles in your new Liferay instance. Although you may migrate users, groups and sites to Liferay DXP, roles still need to be assigned to similar entities in Liferay. Permissions in Liferay define the access and ability given to a certain entity (users, user groups, organizations, etc.). Roles are collections of these permissions, and they can be assigned at various granularities. The legacy system or website may have some roles and permissions, and it may be that mapping these to Liferay roles is a simple task. On the other hand, the legacy system may not possess as extensive a permissions system as there is in Liferay, and therefore will require some default permissions to Liferay entities where there were none before.

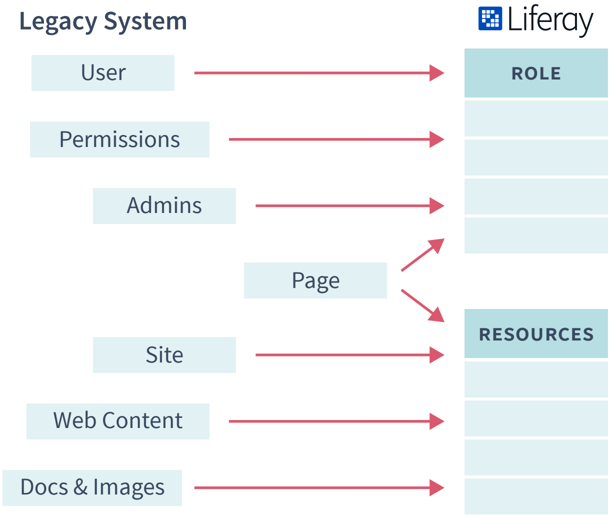


Figure 9

2. How to Fill in the Gaps

If mapping a 1-to-1 relationship from your legacy system to Liferay DXP is not straightforward, data may need to be calculated or accurately derived from disparate sources. Usually this comes into play when calling APIs for Liferay entities. For example, when calling the Liferay User API, there may be some parameters required by the "AddUser" function in which there is no data directly available from the legacy system's user entity. This data may be filled in from many different entities and tables, or calculated from existing data.

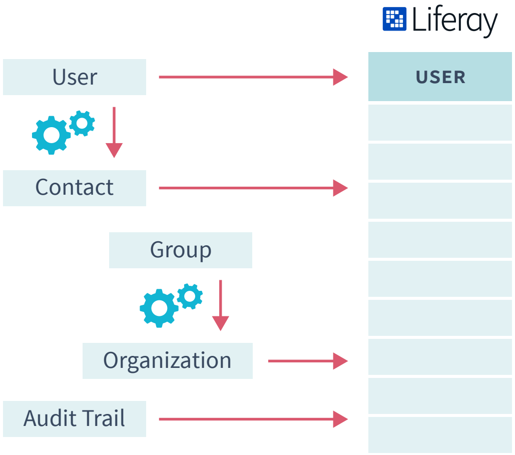
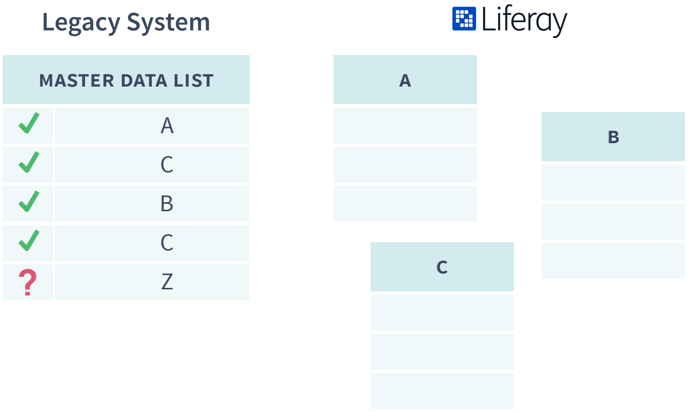


Figure 10

B. Data Mapping and Development

Once the discovery is initiated and gap analysis is completed, your team can start to outline and map out every single call that must be made to the Liferay service layer. A complete checklist of all of the data that must be included in the migration must be created. Every item on this checklist must correspond to some method of migration into Liferay DXP. Often, this method will be a simple API call. While it may be tempting to perform the tasks manually, it is important to remember that such an approach is prone to human error and as such is not recommended.



MASTER DATA LIST	
✓	A
✓	C
✓	B
✓	C
?	Z

Figure 11

Once a complete list of migration items is accounted for, initiate a development project to automate the greater atomic migration (i.e., the aggregation of all the individual API calls) of all legacy system data. The recommendation for Liferay is that your team create a Liferay "Migration Plugin" to read the legacy data. As outlined above, the Liferay plugin can read the legacy APIs, the database, an export format, or whatever else the system provides. The Migration Plugin can then read the data, parse through it and pick out the parameters required for each and every Liferay API call. These calls would then be fired off in succession until all the data has been fed through the Migration Plugin.

Before creating a big monolithic plugin, you may want to break it up into steps and pieces. You can think of migration in terms of ETL - Extract, Transform and Load. Sometimes you can do all three in a single step, but depending on your integration points you might be forced to take a live backup (E), run an initial processing script to clean it up (T), then run a Liferay plugin or set of groovy scripts to do the actual load (L). Sometimes teams try to do all three at once — build a plugin that calls the live database and then hit Liferay APIs that load it into Liferay's database. That may make sense in some situations but there are tradeoffs. There may be a scenario where that is just not possible, or where you want quality checkpoints at each step in the process.

- Did we get all of the correct data from the live database in the initial extract?
- Were we able to transform it into a standard form, such as a LAR or set of files for the Resource Importer?
- Did the load run to completion without errors?

One advantage of breaking the process into steps (there are pros and cons) is that you can do those quality checks at each phase and rewind or test just one phase as needed. Being able to rewind/replay any individual phase (i.e., "The transform looks bad, let's redo just that part.") can hugely speed up a migration implementation.

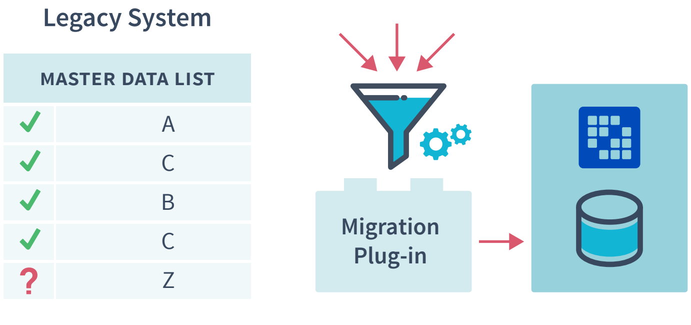


Figure 12

C. Development Testing

Once a Migration Plugin is developed, testing involves running the plugin against a sample dataset from the legacy system. In the testing phase, your team will need to anticipate exception handling, logging, whether or not to pause for different sections of data and any bugs within the API calls. As your Migration Plugin grows more refined, you can test it against larger datasets and iron out more bugs until you can run the plugin against a copy of your entire legacy dataset. Once you are confident that your Migration Plugin is ready, it is time to plan a content freeze for your migration execution.

D. Pre-Production QA & User Acceptance Testing

Prior to migrating and to going live in production, you really need full QA & User Acceptance testing done on a full dataset in a test environment. There are too many projects that fail to get complete user acceptance and business sign off prior to the launch date. The weekend of the install/migration is the wrong time to be asking questions about site layout, styling, etc.

E. Content Freeze

You will most likely need to do some sort of content freeze. A content freeze will provide security in case some data fails to migrate to the new system. The Migration Plugin can be run at an odd hour of the day/night or you can plan for a system outage that is announced well ahead of time in your organization.

F. Migration Dry Run

You will need to stage a dry run of your Migration Plugin on the most recent version of your dataset as possible. Since development of your Migration Plugin was completed, new or erroneous data could easily have been introduced into your production database, spawning bugs or unexpected situations to navigate with additional exception handling for each. There is also the possibility of data corruption. A dry run is important to test for these situations, and with as accurate a copy of production data as possible. Not only will a dry run uncover potential problems, but it will also give you a better estimate of the time needed to execute the migration.

G. Backup

Plan a full backup before executing the migration and ensure it is as accurate a backup of the production as possible, including not simply the legacy system database, but the server itself and all of your deployment settings. You can perform the backup just before the content freeze. In case things do not go according to plan, you must be able to restore to pre-migration settings.

H. Roll-Back Process and Contingency Plan

Using the backup that your team has created, you must make a plan to roll-back to the pre-migration state in the fastest, most secure way possible. This backup and roll-back process must be tested to ensure that you can recreate your pre-migration state accurately, and a dry run of this must be done as well.

I. Full Migration of Production

Now that you have all your bases covered in case things do not go as planned, it is time to schedule the actual migration. Be sure to monitor all Liferay logs and system status as your Migration Plugin executes its migration process. Hopefully, you have included some messages in the log and anticipated any abnormalities that may arise.

J. Final User Acceptance Window

Just because the actual migration was executed does not mean the job is done. There must be a user acceptance period to allow for key team members and content owners to test parts of the new Liferay DXP and ensure that the data that they are concerned with has been transferred over. Identify your key stakeholders and make sure that all of the items on your migration checklist have a key stakeholder assigned to it. Give a timeline and define a period of time for this user acceptance to happen. Also, include some buffer time to fix possible missing data or items.

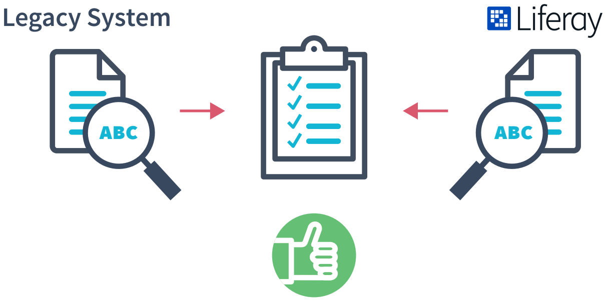


Figure 13

K. Sign Off

Key stakeholders in the legacy system must be identified and be required to sign-off on the user acceptance of the data in the new Liferay deployment. It is an important step in your organization to come to mutual agreement that the migration data requirements as well as the key identifiers for success have been met.

L. Example Timeline

Given all of the above, here is a sample high-level timeline for a Liferay DXP migration project.

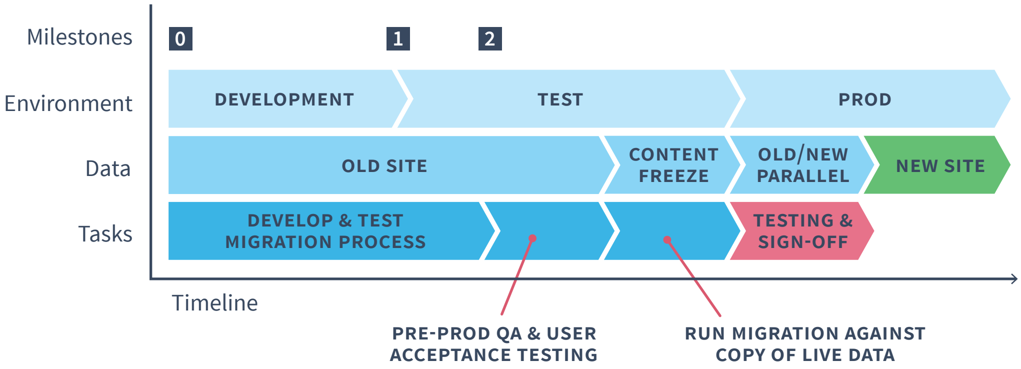


Figure 14

Please note that even here, you can see that the planning and development of the Migration Plugin piece takes up the bulk of time in the overall project.

Level of Effort: Amount of Data vs. Scope of Data

In most cases, the scope of data will require more effort in a migration project than sheer amount of data. This is because a greater scope involves more data entities. This will require more mapping, more API calls and more development work. While the sheer amount of data will affect the actual execution time of the Migration Plugin that you develop, the actual execution is not where the bulk of time is spent in your overall migration project. The bulk (up to half) of the time is spent on data mapping, filling in analysis gaps, migration plugin development and testing.

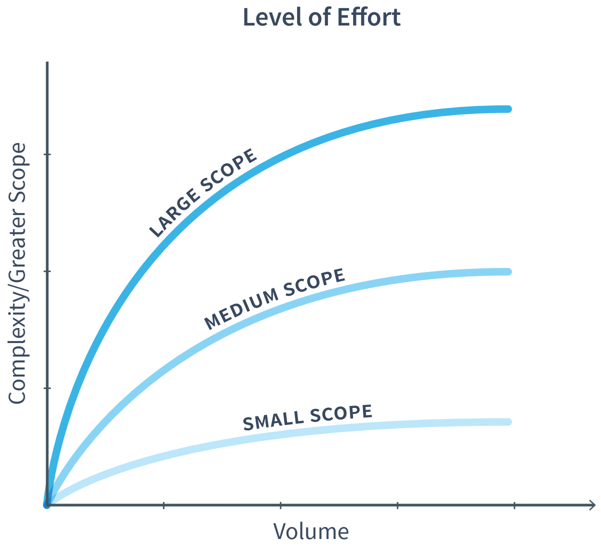


Figure 15

Summary

In this paper, we discussed and outlined a high-level strategy for how to approach and plan for a migration from your existing system to Liferay DXP. There is no one-size-fits-all solution, but the various discovery questions, migration techniques, components of a migration effort, timeline and other factors in determining the general level of effort are important in planning almost any migration. To sum up the most important keys to success and pitfalls:

A. Keys to Success

1. Identifying stakeholders and content owners
2. Developing sign-off requirements
3. A solid and tested backup and roll-back strategy
4. A testing environment that is an accurate reflection of production data
5. Significant investment in the data discovery and mapping phase

B. Pitfalls

1. Failing to backup before migration and not having a backup or roll-back plan
2. Not identifying key stakeholders
3. Not allowing for user acceptance testing and sign off with concrete guidelines and standards
4. Not outlining a strategy for content freeze during the migration execution

About Subscription and Global Services

Liferay Enterprise Subscription

A Liferay Enterprise Subscription is the best way for demanding enterprises to take advantage of Liferay's market-leading innovation with reduced risk exposure and long-term stability of professionally supported software. Your subscription entitles you to regular service packs, a commercial service level agreement and much more.

Liferay Global Services

Liferay Global Services is here to help. Gain access to Liferay core product expertise for your migration project. Your technology staff will acquire extensive product and development knowledge by working alongside Liferay developers. Feel secure in knowing that you will be up and running faster and safer. Optimize your usage of Liferay DXP to maximize your existing and future IT investments.

Moving Forward

Get a Free Demo

A Liferay team member is available to give you an in-depth look into the features and solutions possible with the latest version of Liferay DXP. Our customers include top companies across industries such as Carrefour, Coach, Danone, Fujitsu, Lufthansa Aviation Training, Siemens, Société Générale, VMware and the United Nations.

Request a free demo by visiting liferay.com/request-a-demo

Liferay makes software that helps companies create digital experiences on web, mobile and connected devices. Our platform is open source, which makes it more reliable, innovative and secure. We try to leave a positive mark on the world through business and technology. Hundreds of organizations in financial services, healthcare, government, insurance, retail, manufacturing and multiple other industries use Liferay. Visit us at liferay.com.