

WebSphere Portal から Liferay DXP への 移行

ガイドとベストプラクティス

免責

この文書は情報提供のみを目的として作成され、本書発行日現在における Liferay DXPの仕様及び慣行を示すものです。これらの情報は予告なしに変更される場合があります、本書の記載情報及びLiferay製品の使用にあたっては、読者各位で独自の判断を行い、保証されないことをあらかじめご留意ください。本書は、Liferay、その他関連企業、契約上の確約、条項及び保証をするものではありません。Liferayの顧客に対するLiferayの責任は、Liferay及びLiferay製品の契約によって管理されます。本書にはLiferayとその顧客間の契約と関係することはなく、契約内容に変更を加えるものでもありません。

目次

免責	2
目次	3
はじめに	5
Liferay DXP とは	5
WebSphere PortalからLiferay DXPへの移行	6
このガイドについて	6
移行作業	6
1. 開発環境のセットアップ／要件	8
使用ツール	8
開発ツール	8
Liferay Developer Studio	8
Liferay Workspace	8
Gradle	9
Maven	9
テーマジェネレータ	9
Maven Dependency Repository	10
2. データ量とデータの範囲	10
フィット&ギャップ分析	11
パーミッションとロール	12
ギャップを埋める	13
3. ID(ユーザー)管理の統合	14
4. 埋め込みSQLとアプリケーションコードの変換	14
JavaのNon-Javaどちらか？	14
現行システムのデータベースへのアクセス権の有無	15
ギャップを埋める	15
5. デジタルアセットの統合／移行	16
コンテンツの凍結	16

移行テスト	16
バックアップ	16
ロールバックプロセスと危機管理計画	16
データベースアクセス	17
データマッピングとLiferay APIの呼び出し	17
現行システム移行用プラグインの有無	17
現行システム向けのAPI、サービス及びエクスポート形式／メディアの有無	18
エクスポート形式	19
サードパーティAPI	20
現行システムからの移行範囲	20
6. ポートレットの移行と開発	20
データマッピングと開発	20
7. 移行後のテスト	22
開発テスト	22
本番システムの完全な移行	23
最終ユーザーによる承認	23
8. デプロイ及びGo-Live	24
ダウンタイム又はコンテンツ凍結期間	24
サインオフ	24
まとめ	24
成功への鍵	24
落とし穴	24
サブスクリプション及びグローバルサービスについて	25
Liferay エンタープライズサブスクリプション	25
Liferay グローバルサービス	25
参考文献	25

はじめに

利用し続けてきたシステムのプラットフォームが時代遅れ、または製品開発が終了してしまった、もしくは会社の急激な成長などにより、買収などにより増えた各部門、子会社などで利用しているシステムの統合がおいつかず、サイロ化してきたなど、会社の成長や、技術の変化などで新しいプラットフォームへの移行を検討されている会社が多くあります。

このような課題を抱える企業は、サイロ化したデータを1箇所に統合し、認証なども統合されたシステムとして、さらなる会社の成長にも対応できる、拡張性の高いシステムの必要性を検討しています。

このガイドは、それらの課題にLiferay DXPがどのようなソリューションを提供できるか、また現行システムからの移行にあたっての検討事項を説明します。

Liferay DXP とは

Liferay デジタルエクスペリエンスプラットフォームは、ヒューレット・パカードやアメリカ海軍などをはじめとする大企業、巨大組織で利用されています。

これらのお客様は数百万人のユーザーと、数十万のアクティブユーザーが同時に利用し、1分の停止で数千万円の損失になってしまうようなミッションクリティカルな環境でLiferay DXPを利用しています。

Liferay DXPは、LinuxやWindowsサーバーといった複数のOS上、またTomcat, JBoss, Wildflyといった複数の主要なアプリケーションサーバーで動作し、OracleやMySQL, PostgressやSQL Serverといった主要データベースとの互換性を持つため、お客様の環境に合わせて設定も可能です。

それだけではなく、昨今のシステムはクラウド上で動作することも多く、オンプレミスでのシステム構築はもちろん、AWSやAzure、GCPといった主要なクラウドベンダー上でも動作実績が多数あり、世界中の異なる業界のお客様に利用していただいております。

WebSphere PortalからLiferay DXPへの移行

Liferayの強みはその拡張性の高さにあります。WebSphere PortalからLiferayへの移行を検討されている場合、WebSphere Portalと同様、LiferayはJSR-286のポートレットが動作する環境を提供しています。それに加えてOSGiフレームワークの導入により、サーバーを動作させたままでの機能（バンドル）の追加、削除が容易に行えるのも特徴です。

機能的にもWebコンテンツ管理、ドキュメント管理など似た機能を搭載しているため、移行におけるLiferayの機能を学ぶラーニングカーブも、比較的低く抑えられる可能性があります。

このガイドについて

このホワイトペーパーの目的は、移行プロジェクトの計画を立てるのを支援することです。その際に必要な、既存システムからLiferay Digital Experience Platform(DXP)へ移行で考慮すべき項目を説明します。

作業のレベルは、既存のプラグインを実行するだけの単純なタスクから、多くの工数とリソースを必要とするかなり複雑な開発プロジェクトまで、非常に広い範囲にわたります。問題は、特定の移行プロジェクトに必要な作業レベルをどのようにして把握しているかです。

答えは多くの要因によって異なるため、このホワイトペーパーではそれら方法論を概要レベルで説明します。最初に、移行に必要な範囲と、作業を評価するためのベストプラクティスを説明し、次にLiferayのお客様に最適な方法と、実用的なプロジェクトのスケジュールを計画する方法を検討します。また移行に伴う作業に影響を与える要因も合わせて触れます。

移行作業

Liferay DXPにはオープンなフレームワークとAPIがあり、現在のシステムにあるあらゆるデータを活用することができます。あるシステムからLiferay DXPにコンテンツを移行するのに使用する最も一般的な方法は、レガシーシステムからデータを読み取り、そのデータをパラメータとして使用してLiferay APIを呼び出すLiferayプラグインを作成することです。

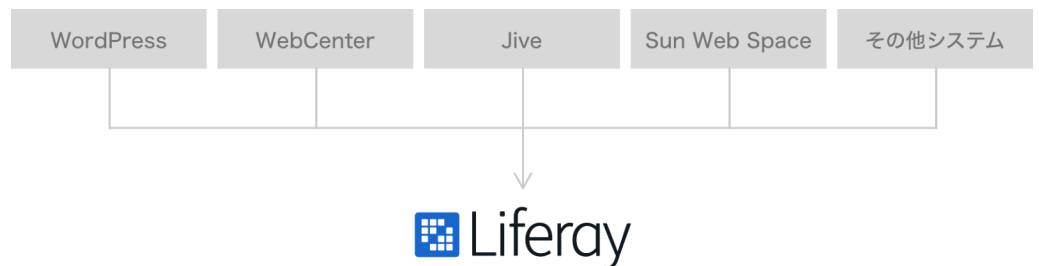
たとえば、レガシーCMSのコンテンツを移行する場合、Liferayプラグインは、データベースに直接アクセスするか、CMSのAPIまたはサービスを呼び出すことによってCMSコンテンツデータを読み取り、Liferay CMS APIを繰り返し呼び出し、

データを移行します。同じ方法で、Liferay DXPに変換可能なユーザ、ロール、ドキュメント、およびその他のアセットからの他のすべてのコンテンツも移行してきます。ここでの主な質問は次のとおりです。

1. APIまたはWebサービスを使用して、レガシーシステムデータにアクセスできますか?または、データベースまたはエクスポートフォーマットに直接アクセスできますか?
2. Liferayプラグインは既に作られていますか?

Liferay Marketplaceには、あるシステムから別のシステムに自動的にコンテンツを移行する、公式のLiferayプラグインはありません。しかし、Liferay Communityのメンバーは、移行のためのプラグインとツールを作成しており、またLiferay Global Services(Liferayのコンサルタント)は、パートナー企業と連携して、移行ツールを作成したりすることで、レガシーシステムからLiferay DXPにデータを移行するお手伝いをします。

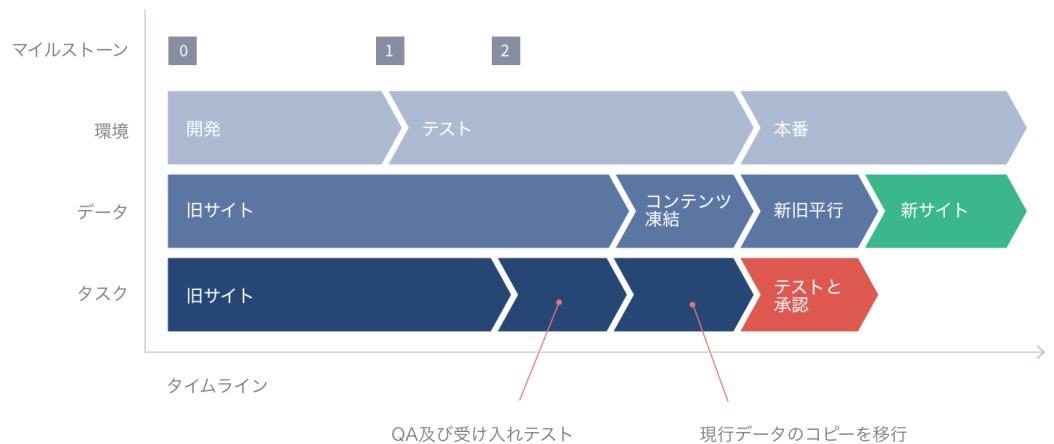
何年にもわたって、Liferayは、Wordpress、Jive、Sun Web Space、Oracle WebCenterなどのシステムからの移行を支援してきました。



一般的な移行プロセスは次のような流れになります：

1. 開発環境のセットアップ/要件
2. データ量とデータの範囲
3. 認証の統合
4. 埋め込みSQLとアプリケーションコードの変換
5. デジタルアセットの統合又は移行
6. ポートレットの移行と開発
7. 移行後のテスト
8. デプロイ及びGo-Live

ここにLiferay DXP移行プロジェクトのサンプルのタイムライン概要を示します。以下のように、プロジェクト全体の中で最も時間がかかるのが、データ移行プラグインの計画と開発であることがわかります。



1. 開発環境のセットアップ／要件

使用ツール

- JDK
- データベースクライアントツール
- Liferay DXP
- Liferay IDE, Liferay Developer Studio (LDS) もしくは他の開発環境 (Eclipse、NetbeansやIntelliJなど)

開発ツール

Liferay DXPの開発はツールに依存しません。コードは、開発者が使い慣れている任意のツールを使用してDXP用に開発できます。次のセクションでは、Liferay DXPの開発を成功させるためにLiferayが提供するいくつかの一般的なツールについて説明します。

Liferay Developer Studio

Liferayは開発者を支援するためにLiferay統合開発環境(IDE)を提供しています。Liferay Developer StudioはEclipseプラグインであり、Liferay設定ファイル用の様々なエディタと、Liferay Workspaceとの統合環境を提供します。最新リリースのLiferay Developer Studio 3.1は、Liferay DXP用に設計されています。

Liferay Workspace

Liferay WorkspaceはLiferayプロジェクトを保持し管理するために設計された環境です。WorkspaceはGradleベースのツールで、Liferay DXPモジュールを作成するための公式の方法です。開発者はWorkspaceを使用してLiferay DXPバンドルを管理し、開発中のモジュールをビルド/デプロイすることができます。

す。Gradleビルドスクリプトは、ビルドを自動化するための継続的インテグレーション（CI）サーバーの機能を提供します。

Liferay Workspaceを使用すると、開発者環境とクラウド環境の間で一貫性が保たれます。たとえば、Gradleバージョン間で大きな違いがある可能性があるため、Gradleラッパーを使用して、サポートするGradleバージョンを強制します。特にOSGi環境にデプロイする場合は、バージョンが重要であるため、すべてのモジュール（ポータル-カーネルなど）に共通の依存関係も提供します。

Liferayワークスペースを使用してLiferay DXPモジュールを構築するためのドキュメントは、次の場所にあります。

https://dev.liferay.com/en/develop/tutorials/-/knowledge_base/7-1/liferay-workspace

Gradle

WorkspaceはGradleベースのツールであるため、開発チームはGradleに精通することをお勧めします。GradleはApache AntとApache Mavenのビルドツールの最高のコンセプトを活用したビルドツールで、Mavenからの設定よりも規約をサポートし、Antよりも簡単にタスクを作成しカスタマイズできます。GradleはGroovyベースの言語を使ってタスクを定義します。Liferayには、Liferay DXPモジュールおよび、テーマジェネレータで使用するために設計された、カスタムビルドタスクが用意されています。

Maven

Mavenを利用してLiferay DXPモジュールを構築することもできます。LiferayはMavenでのLiferay DXP開発を容易にするために設計された、Maven統合プラグインも提供しています。

Mavenを使ってLiferay DXPモジュールをビルドするためのドキュメントは、次の場所にあります。

https://dev.liferay.com/en/develop/tutorials/-/knowledge_base/7-1/maven

テーマジェネレータ

Liferay DXPのテーマは、新しいThemes Generatorツールを使用して構築されています。Themes GeneratorはGulp上に構築されたコマンドラインツールで、Liferay DXPテーマの作成とビルドに使用できます。

テーマジェネレータの使用に関するドキュメントは次のとおりです。

https://dev.liferay.com/en/develop/tutorials/-/knowledge_base/7-1/introduction-to-themes

テーマジェネレータの代わりに、BLADEテーマテンプレートを使用することもできます。

https://dev.liferay.com/en/develop/reference/-/knowledge_base/7-1/theme-template

Maven Dependency Repository

外部のMavenリポジトリアクセスに制限がある場合、開発者は内部リポジトリを利用できます。これを行うには、settings.gradleファイルを修正します。

- 内部リポジトリのMavenリポジトリURLを変更します
- 明示的にワークスペースに存在しない依存プロジェクトを含むすべてのプロジェクトのリポジトリURLを更新するには、settings.gradleファイルの末尾に次の行を追加します：

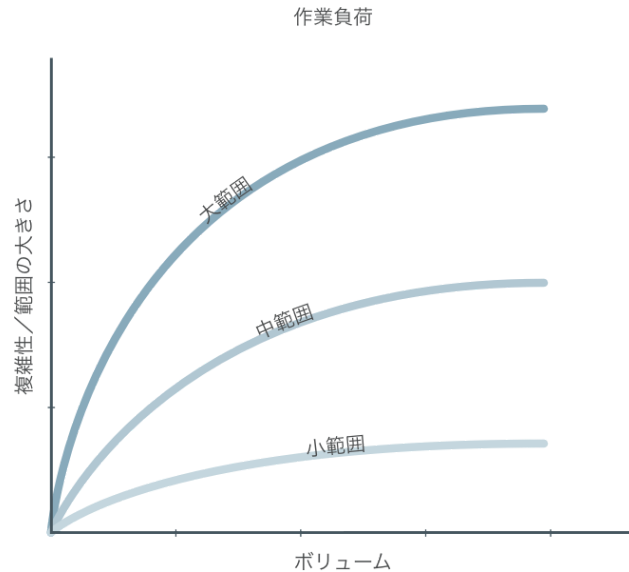
```
gradle.allprojects {  
  
    repositories {  
  
        maven {  
  
            url "<repository url>"  
  
        }  
  
    }  
  
}
```

2. データ量とデータの範囲

ほとんどの場合、移行プロジェクトではデータの量よりもデータの範囲が作業量に影響します。これは、スコープが広いほどデータエンティティが多くなるためです。つまりより多くのマッピング、API呼び出しが必要となり、そのための開発が必要になります。

膨大なデータ量も、開発したデータ移行プラグインの実行時間に影響しますが、実際の実行時間は移行プロジェクト全体でみると、その一部でしかありません。移行プロジェクトの時間の大部分（最大半分まで）は、データマッピン

グ、分析ギャップの埋め合わせ、マイグレーションプラグインの開発とテストに費やされます。



移行プロジェクトがいつ完了できるかを合理的に見積もることができるように、必要な手順をすべてタイムラインに含めることが重要です。適切な移行には、データ自体の実際の移行以外のいくつかのタスクが含まれます。それらも考慮すべき項目です。

フィット&ギャップ分析

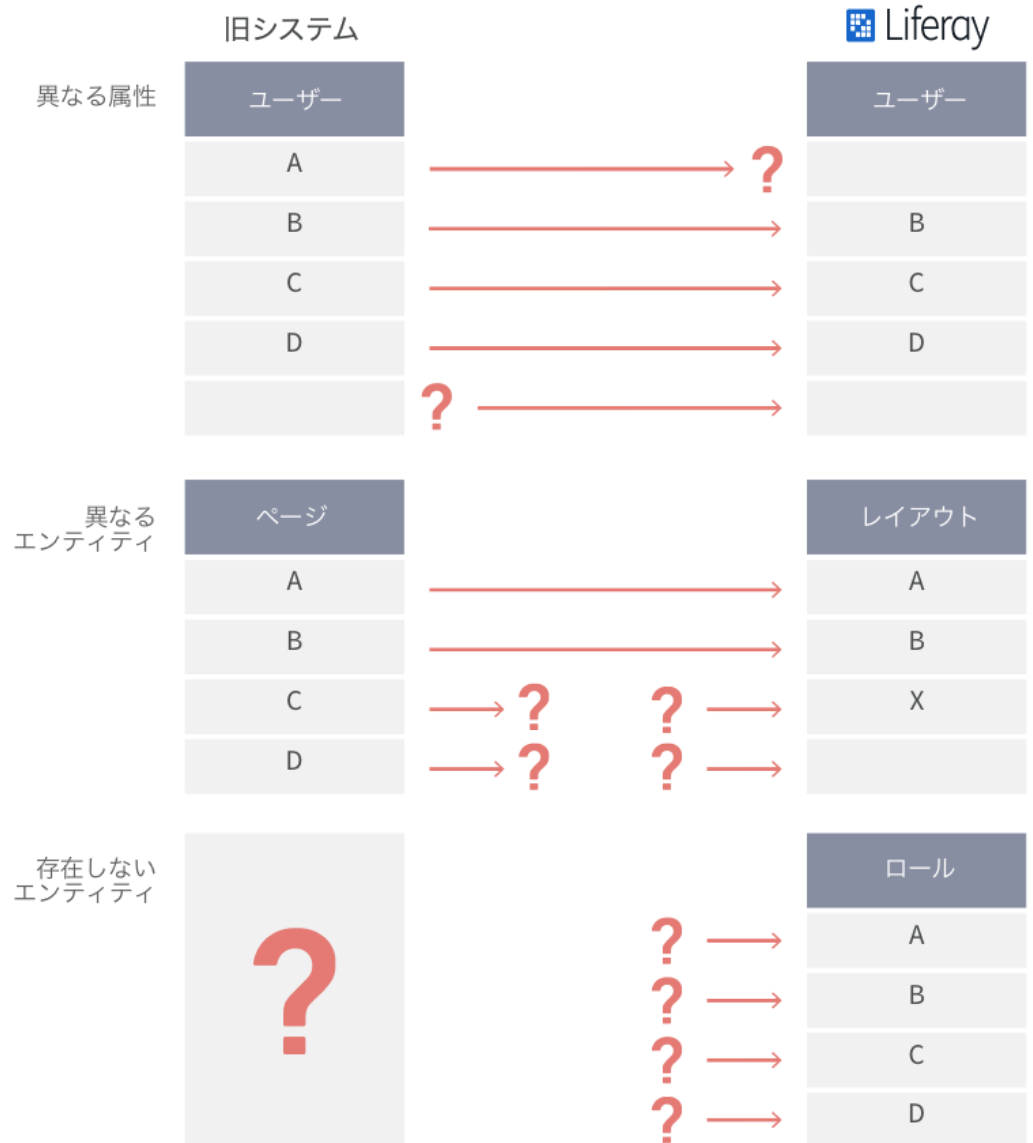
すべてのWebフレームワークが同じというわけではありません。他の製品に存在するものがLiferayにまったく存在しない可能性があり、その逆も同様です。データフィールド、またはメタデータが移行元の製品にしか存在しない場合は、そのデータをLiferayに保存する方法、または保持するかどうかを決定する必要があります。

同様に、Liferayエンティティが特定のデータを必要としていて、古い製品からの一対一のマッピングがない場合は、異なるデータをまとめる必要があるかもしれません。フィット&ギャップ分析の段階では、チームの目標はLiferay展開でデータセットの基本を定義することです。

多くの場合これは、どのエンティティAPIを呼び出す必要があるか、各呼び出しに必要なパラメータ、および従来のシステムが提供できるものを決定することで、簡単に実行できます。

ここで指摘しておくのは、古いシステムのデータ品質が深刻な問題になる可能性があることです。新しいデータモデルの構築について説明する際には、古いシステムから排除できるものを文書化する必要があります。関連性がなく

なったため、何らかの方法で削除する必要がある、ユーザー、ページ、またはカスタムエンティティが存在する可能性があります。



パーミッションとロール

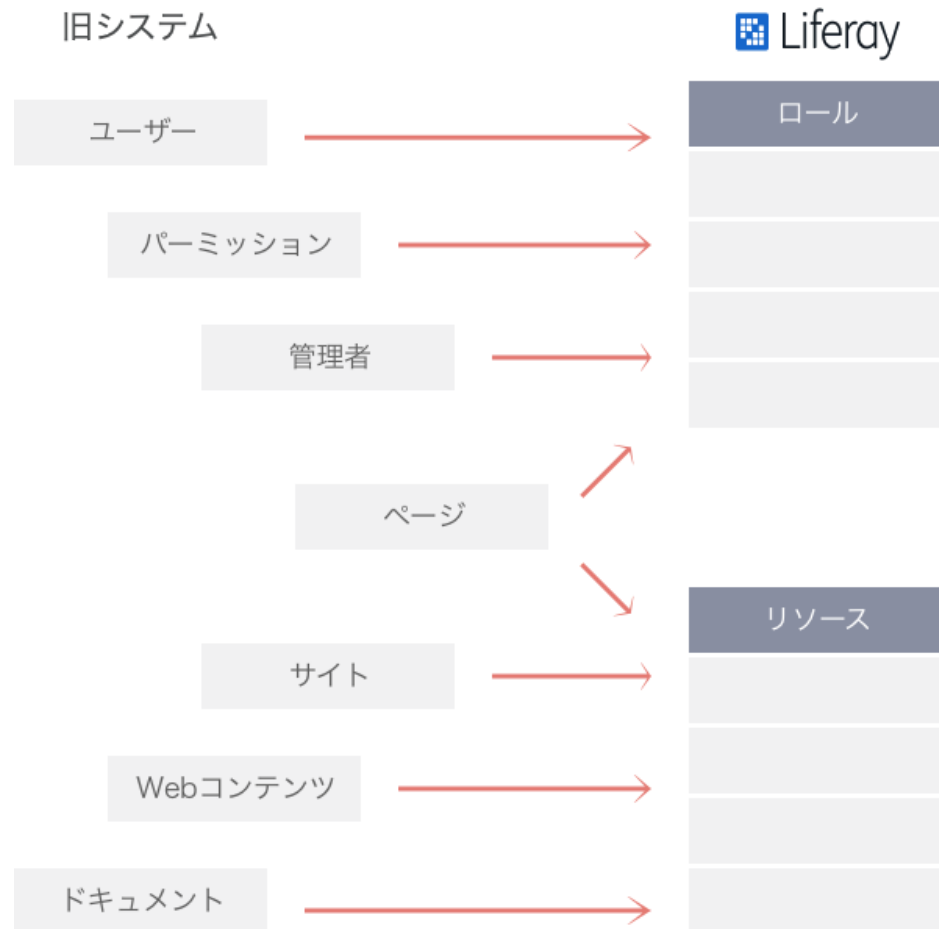
Liferay DXPはロールベースの権限管理システムを搭載しています。Liferayのロールとパーミッション管理の粒度は細かいので、新しいLiferayインスタンスにロールを割り当てるときは特に注意が必要です。

ユーザー、グループ、およびサイトをLiferay DXPに移行することはできますが、それでもロールにはLiferay内の同様のエンティティに割り当てる必要があります。Liferayのパーミッションは、特定のエンティティ（ユーザー、ユーザーグループ、組織など）に与えられるアクセスと能力を定義します。

ロールはこれらのパーミッションの集合であり、さまざまな粒度で割り当てることができます。レガシーシステムやWebサイトにはいくつかのロールと権

限があり、これらをLiferayのロールにマッピングするのは簡単な作業かもしれません。

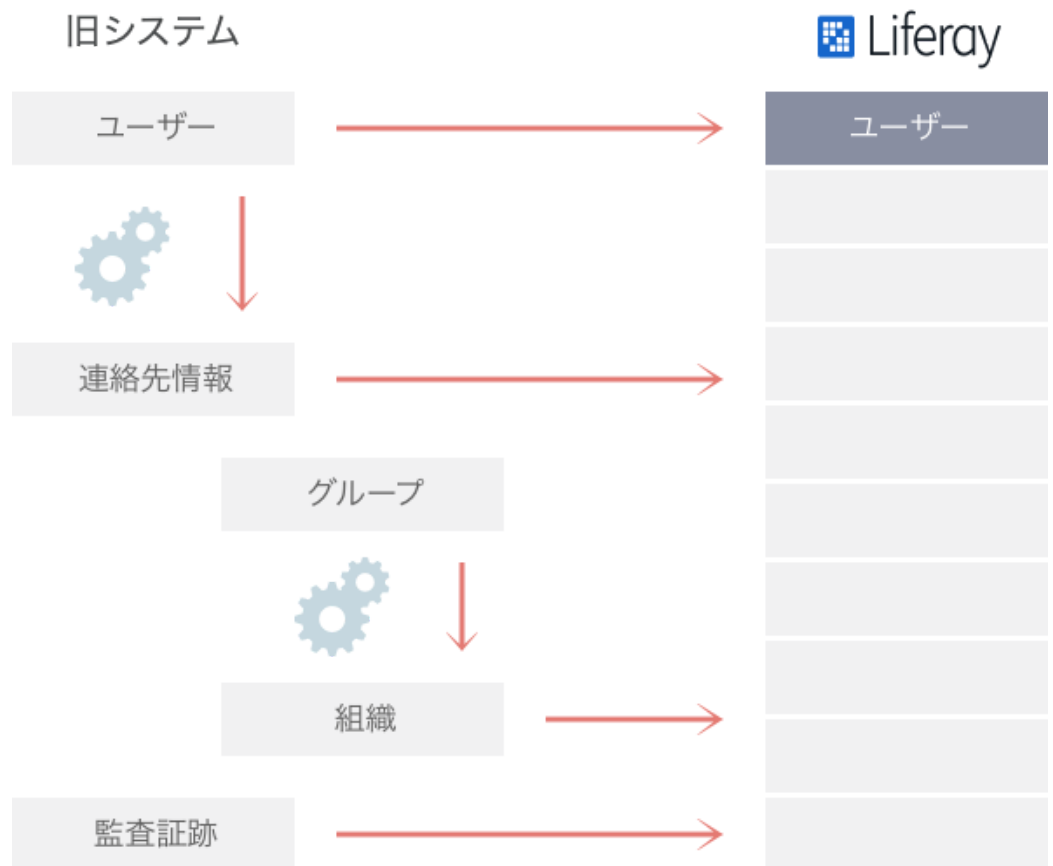
一方、レガシーシステムはLiferayにあるほど広範なパーミッションシステムを持っていない可能性があるため、以前にはなかったLiferayエンティティに対するデフォルトのパーミッションが必要になります。



ギャップを埋める

レガシーシステムからLiferay DXPへの1対1の関係のマッピングが簡単でない場合は、既存データを処理して生成するか、または異なるソースから正確に導き出す必要があります。これらデータはLiferayエンティティのAPIを呼び出すときに利用するものです。

たとえば、LiferayユーザAPIを呼び出すときに、「AddUser」機能で必要とされるパラメータのデータが、レガシーシステムのユーザエンティティに直接対応するものがないかもしれません。その場合はさまざまなエンティティやテーブルからデータを集めてきて入力したり、既存のデータから計算したりします。



3. ID(ユーザー)管理の統合

別のID管理システムまたはサーバーがある場合、チームは展開前に本番用IDのコピーと比較してテストする必要があります。テスト環境と本番環境でIDストアが異なる場合は特にそうです。これが不可能な場合、UATテストには本番用IDに対するテストのタスクを含める必要があります。いずれにせよ、テスト中に適切な作業が必要になります。

4. 埋め込みSQLとアプリケーションコードの変換

JavaのNon-Javaどちらか？

LiferayはJava上で実行されるため、他のJavaテクノロジーから移行すると、相互運用性が向上する可能性があります。

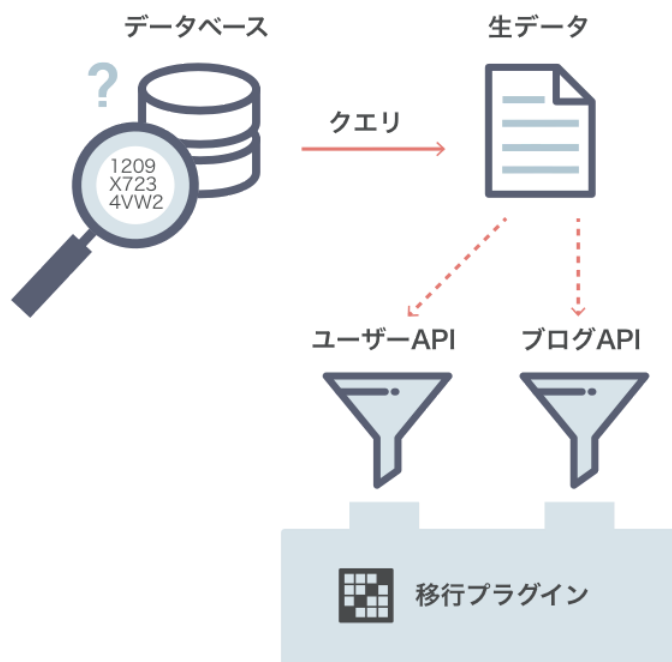
APIや他のインタフェースもあるので、この要素だけがLiferayへのデータ移行に大きな影響を与えることはないかもしれませんが、WSRPなどのJavaテクノロジーの実装や、関連する製品を経由しての、Java Webアプリケーションの移行

を検討する価値はあるかもしれません。これらのオプションは、Java以外のテクノロジーでは使用できるとは限りません。

現行システムのデータベースへのアクセス権の有無

レガシーシステムから利用できるAPI、(Web) サービス、およびエクスポートフォーマットがない場合は、データベースを介してレガシーデータにアクセスすることしかできません。

この方法は時間がかかる調査と解析が必要になります。データベースのデータは生データになるため、正しいデータにアクセスしてインポートできるかの確認が最も重要です。



ギャップを埋める

データ移行プロセス中に、以下のようなレガシーシステムまたはLiferayのギャップに気付くことがあります。

Liferayに、レガシーシステムに対応するデータ構造がない場合。

このような状況では、データの価値を決定する必要があります。価値がある場合は、既存のLiferayメタデータフィールドにマッピングするか、Liferayカスタムフィールドを使用して、レガシーデータのメタデータを作成できます。

Liferay DXP APIが、レガシーシステムには存在しないパラメータとメタデータを必要とする場合。

これは多くの場合、最初のケースよりも簡単な状況です。開発者は、ビジネス上の利害関係者と協力して、データの生成方法やデータのデフォルト設定方

法を決定できます。これは、欠けているデータのタイプ（すなわち、タイムスタンプと画面名）によって異なります。

5. デジタルアセットの統合／移行

コンテンツの凍結

移行前に、何かしらのコンテンツ凍結をする必要があるでしょう。一部のデータが新しいシステムへの移行に失敗した場合に備えて、コンテンツの凍結で環境復元を担保します。移行プラグインは、昼夜のアクセス頻度が低い時間で実行することも、事前に計画したシステム停止中に実行することもできます。

移行テスト

データセットの最新バージョンで、マイグレーションプラグインのドライランを可能な限り実行する必要があります。マイグレーションプラグインの開発が完了してからは、新しいデータやエラーのあるデータが本番データベースに取り込まれたり、バグや予期しない状況が発生したりして、それぞれに追加の例外処理が必要になります。データ破損の可能性もあります。

こうした状況をテストするには、本番データのコピーをできるだけ正確に作成して、ドライランを実行することが重要です。ドライランによって潜在的な問題が明らかになるだけでなく、マイグレーションの実行に必要な時間をより正確に見積もることができます。

バックアップ

移行を実行する前に、フルバックアップを計画し、レガシーシステムデータベースだけでなく、サーバー自体とすべての展開設定も含めて、できるだけ正確な本番環境のバックアップであることを確認します。コンテンツの凍結直前にバックアップを実行してください。計画どおりにいかない場合は、移行前の設定に復元できる必要があります。

ロールバックプロセスと危機管理計画

チームで作成したバックアップを使用して、できるだけ迅速、かつ安全な方法で移行前の状態にロールバックする計画を立てる必要があります。このバックアップとロールバックプロセスをテストして、移行前の状態を正確に再作成できることを確認する必要があります。また、これをドライランで実行する必要もあります。

データベースアクセス

生のデータベースを調べることは、おそらく最も面倒で、望ましくないデータ抽出方法です。抽出するデータがデータベースとテーブルにあり、利用可能なドキュメントがない場合、適切な列とテーブルをオブジェクトとコンテンツに一致させるために必要な時間と労力は膨大なものになり、タスクは非常に面倒なものになります。ですが特に古いシステムでは、他に選択肢が無いことも多くあります。

データマッピングとLiferay APIの呼び出し

データマッピングの確認は、データ移行の最も重要かつ時間のかかる段階です。生のデータベースからのものであろうと、文書化されたデータセットを返すAPIからのものであろうと、開発者はそれを正しく扱う必要があります。

たとえば、レガシーシステムとLiferay DXPの両方にブログ機能がある場合、各メタデータは慎重に整理され、対応するLiferayメタデータにマッピングされる必要があります。正常に転送するには、インベントリ分析のすべての項目を慎重に整理してデータが失われないようにする必要があります。

現行システム移行用プラグインの有無

移行の最善のシナリオは、現在のシステムにLiferayプラグインがすでに存在していることです。ただし、Liferayの公式プラグインが利用できない場合は、サードパーティのプラグインを利用したり、Liferay グローバルサービスチームにあなたの正確な仕様に合わせてプラグインを調整してもらおう、という選択肢もあります。

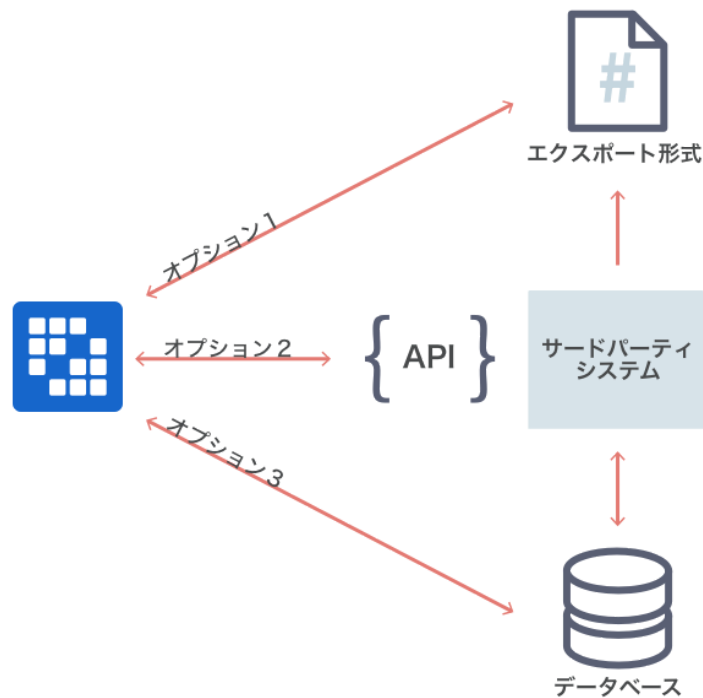
たとえば、コンテンツに関連付けられている、さまざまなメタデータを含めるか、除外するかなど、顧客のニーズはそれぞれ異なります。顧客AはJiveからすべてのコンテンツを移行したいと思うかもしれませんが、顧客Bは自分のブログコンテンツを除いてすべてのコンテンツを移行したいと思うかもしれません。

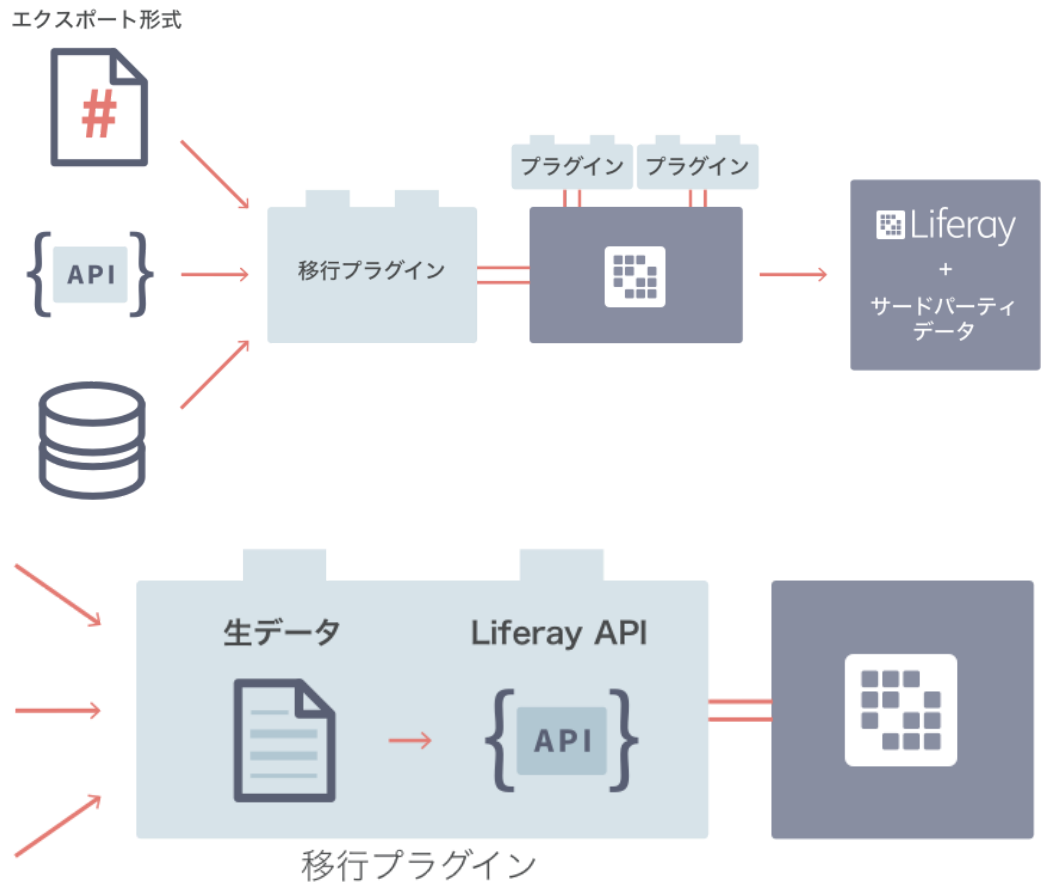
別のケースでは、A社とB社の両方がブログコンテンツを移行したいが、A社はすべてのブログ投稿のすべてのメタデータを望んでいるのに対し、B社はメタデータ全体のサブセットしか必要としません。



現行システム向けのAPI、サービス及びエクスポート形式／メディアの有無

移行をすぐに実行できるプラグインがない場合は、既存のAPI、サービス、またはエクスポート形式を呼び出すためのプラグインを開発する必要があります。





Webページのスクレイピング

移行が必要なコンテンツがレガシーシステム上にあり、それほど複雑ではない場合（たとえば、プレーンWebコンテンツとさまざまなフォームや画面を持つWebアプリケーション）、スクリーンスクレイパーアプリケーションを開発することができます。

このアプリは、実行時に、または移行プロセスの一環として、Liferayに動的に追加できます。Liferayテーマを最大限に活用するためには、この方法で収集したHTMLを、変換/クリーンアップすることをお勧めします。

それにより、古いサイト（またはWord文書）のカットアンドペーストを使用し、意図せずに大量のインラインCSSを埋め込んでしまい、サイトの見栄えが壊れてしまう、というような問題に対処することができます。

エクスポート形式

ソフトウェアベンダーは、独自のエクスポート形式またはXMLファイルなどの標準化された形式を提供していることがあります。

Liferay DXPは、サイトコンテンツをLAR（Liferay Archive）ファイルにエクスポートします。LAR（Liferay Archive）ファイルには、XMLファイルと、ド

キュメントリポジトリのドキュメントなどのオブジェクトを含む、ファイルおよびディレクトリの構造が含まれています。

レガシーシステムからのデータエクスポートは、エクスポート形式に関するドキュメントがあるか確認してください。そこに記載されている標準フォーマットを使用することをお勧めします。

もしドキュメントがなければ、この方法でのデータマッピングはかなりの時間を必要とします。ですがデータマッピングの正確性は、移行を成功させるためには不可欠です。

サードパーティAPI

データがAPIを介してアクセス可能な場合もあります。Webサービスは、データを第三者に公開する一般的な例です。APIがどのような種類のデータを返すかについての公式ドキュメントをあたってみてください。それにより移行が必要なデータを見つける時間を節約できるかもしれません。ただしいずれにせよ、データマッピングは行う必要があります。

現行システムからの移行範囲

移行項目例:

- CMS
- ブログ
- ウィキ
- 掲示板/フォーム
- アプリケーションデータ
- その他Webコンテンツ

6. ポートレットの移行と開発

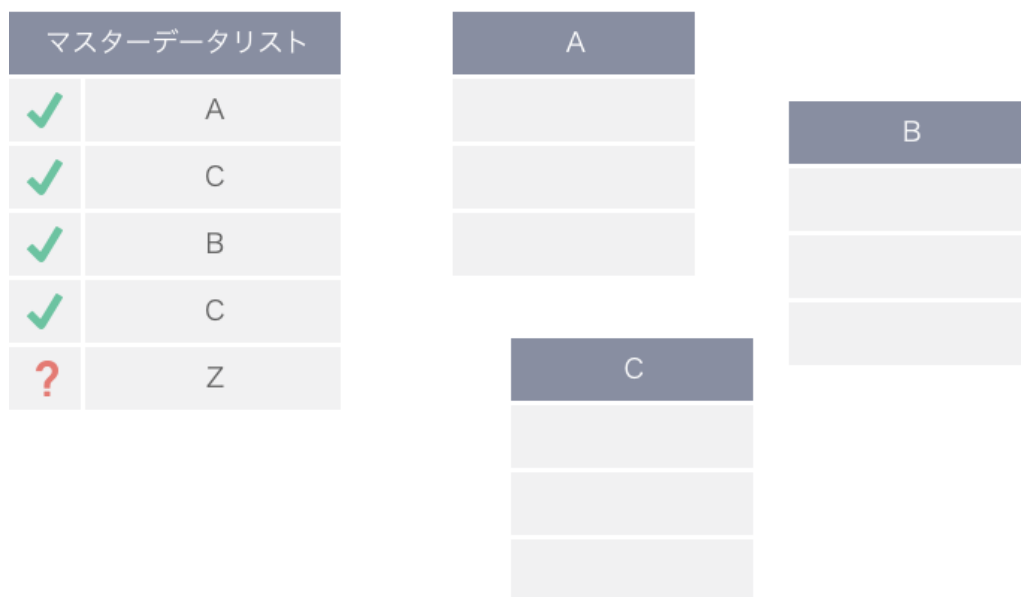
データマッピングと開発

要件定義が開始され、ギャップ分析が完了すると、Liferayサービス層に対するすべての呼び出しをマップできているはずですが、移行に必要なすべてのデータの、完全なチェックリストを作成する必要があります。

このチェックリストの各項目は、Liferay DXPへの移行方法に対応している必要があります。多くの場合、このメソッドは単純なAPI呼び出しになります。手動でタスクを実行するのは魅力的かもしれませんが、このようなアプローチは人的ミスを引きやすいため、お勧めできません。

旧システム

 Liferay



移行項目の完全なリストがまとめられたら、すべてのレガシーシステムデータのより大きなアトミックな移行（すなわち、すべての個々のAPIコールの集約）を自動化するための開発プロジェクトを開始します。

Liferayはレガシーデータを読み込むための、Liferay用“マイグレーションプラグイン”の作成を勧めています。

上で少し触れましたが、LiferayマイグレーションプラグインはレガシーAPI、データベース、エクスポートフォーマット、あるいはシステムが提供するデータは何でも読むことができます。その後、読み込んだデータを解析して、Liferay API呼び出しごとに必要なパラメータを選択し、すべてのデータが移行されるまで、APIを繰り返し起動しLiferayへのデータインポートを実行します。

1つで全てを処理するような巨大なプラグインを作成するのではなく、移行ステップや機能毎などのパーツに分割することをお勧めします。ここで考慮する必要があるのはETLの視点、抽出(Extract)、変換(Transform)とデータ投入(Load)です。

1つのステップで3つすべてを実行できる場合もありますが、統合ポイントによっては、ライブバックアップを取って（Extract）、初期処理スクリプトを実行してクリーンアップし（Transform）、Liferayプラグインを実行するか、設定する必要があります。そして実際のデータ投入（Load）を実行するためのGroovyスクリプトの使用、という形になります。

ライブデータベースを呼び出すプラグインを作成し、Liferay APIを使用して、データをLiferayのデータベースに投入するステップ3つ、すべて一度に実行し

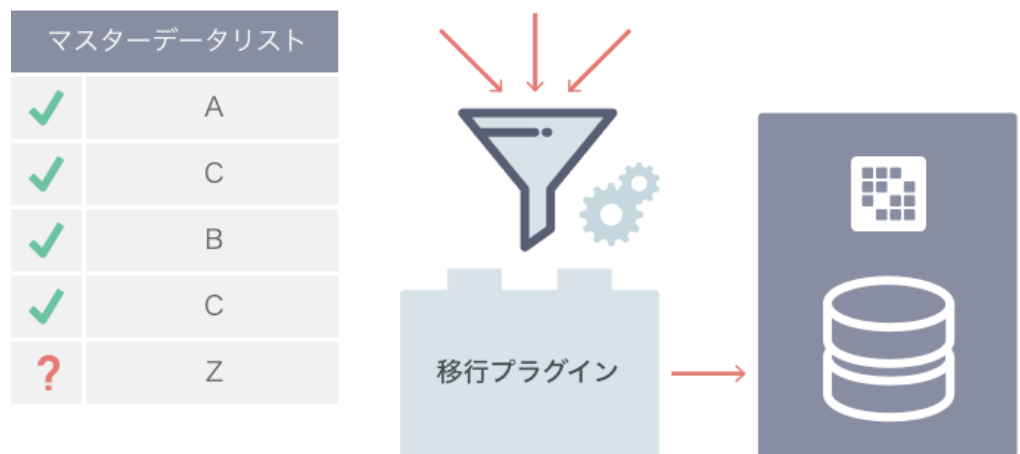
たい場合もあるかもしれません。状況によっては有効なこともあるかもしれませんが、それによるトレードオフも事前に理解しておく必要があります。ケースによってETLを一度に実行するのが不可能な場合や、プロセスの各ステップで質の高いチェックポイントが必要な場合があります。

- 最初の抽出で、現行システムのデータベースから正しいデータをすべて取得できましたか？
- それをLARやResource Importer用の一連のファイルなどの標準形式に変換できましたか？
- データ投入はエラーなしで完了するまで実行されましたか？

プロセスを複数のステップに分割することの利点（長所と短所があります）は、各段階で品質チェックを行い、必要に応じて1段階だけ巻き戻し、またはテストできることです。個々のフェーズを巻き戻し/再生できるようになると（つまり、「変換がうまくいかないの、その部分だけをやり直しましょう」）、移行の実装が大幅にスピードアップします。

旧システム

 Liferay



7. 移行後のテスト

開発テスト

移行プラグインの開発後、レガシーシステムのサンプルデータセットに対してプラグインを実行するテストを実施します。テスト段階では、チームは例外処理、ログ記録、データのさまざまなセクションで一時停止するかどうか、およびAPI呼び出し内のバグを予測する必要があります。

移行プラグインがより洗練されたものになったら、より大きなデータセットに対してそれをテストし、レガシーデータセット全体のコピーに対してプラグインを実行できるまでより多くのバグを解決することができます。移行プラグ

インの準備が整ったと確信したら、今度は移行実行のためにコンテンツの凍結を計画します。

本番システムの完全な移行

これで、計画どおりに進まない場合に備えてすべてのベースがカバーされたので、実際の移行のスケジュールを設定します。マイグレーションプラグインが移行プロセスを実行する際には、必ずすべてのLiferayログとシステム・ステータスを監視してください。ログにいくつかのメッセージを含め、発生する可能性のある異常を予期しておいてください。

最終ユーザーによる承認

実際の移行が実行されたからといって、その作業が完了したわけではありません。主要なチームメンバーとコンテンツ所有者が新しいLiferay DXPの一部をテストし、彼らが関係するデータが確実に転送されているか確認するために、ユーザー受入テスト期間が必要です。

主要な関係者を特定し、移行チェックリストのすべての項目に重要な関係者が割り当てられていることを確認します。タイムラインを与えて、このユーザーが受け入れられるまでの期間を定義します。また、不足している可能性のあるデータまたは項目を修正するために、ある程度のバッファ時間を含めません。

旧システム

 Liferay



8. デプロイ及びGo-Live

ダウンタイム又はコンテンツ凍結期間

スケジュールされたダウンタイム、またはコンテンツのフリーズウィンドウを考慮すると、計画が容易になり、データの破損や従来のシステムと新しいLiferayシステムとの間の矛盾を最小限に抑えることができます。ダウンタイムが発生しない場合、チームはできるだけ早く移行をテストおよび最適化し、移行作業中に発生したデータの齟齬や問題への対応が必要です。

サインオフ

レガシーシステムの主要な利害関係者を特定し、新しいLiferayに展開されたデータをユーザーが承認したことを確認します。移行データの要件と成功の鍵となる要素が満たされていることを相互に合意することは、組織にとって重要なステップです。

まとめ

このホワイトペーパーでは、既存のシステムからLiferay DXPへの移行方法とアプローチの計画について、概要を説明しました。万能の解決策はありませんが、さまざまな検出に関する質問、移行手法、移行作業の要素、スケジュール、その他の一般的な作業レベルを決定する要素が、ほとんどすべての移行を計画する上で重要です。成功と落とし穴の最も重要な鍵をまとめると：

成功への鍵

1. 利害関係者とコンテンツ所有者の特定
2. サインオフ要件の開発
3. しっかりとテストされたバックアップとロールバック戦略
4. 生産データを正確に反映したテスト環境
5. データ発見およびマッピング段階への多大な投資

落とし穴

1. 移行前にバックアップに失敗し、バックアップまたはロールバック計画がない。
2. 主要な利害関係者を特定しない。
3. ユーザー受け入れテストを許可せず、具体的なガイドラインと標準だけを参照しただけでシステムの引き渡しを行う。

4. 移行実行中のコンテンツ凍結の戦略の概要を説明しない。

サブスクリプション及びグローバルサービスについて

Liferay エンタープライズサブスクリプション

Liferay エンタープライズサブスクリプションは、導入や運用におけるリスクを減らし、長期にわたった安定したLiferayの活用に最適な方法です。サブスクリプションには機能向上やバグフィックスが含まれるサービスパックの提供や、エンタープライズレベルのサポートなど、多くのサービスが含まれます。

Liferay グローバルサービス

Liferay 専門家の手助けが必要でしょうか？その場合は弊社 Liferay グローバルサービス（コンサルタント）がお手伝いします。あなたの技術スタッフが、Liferay 製品の専門知識を持った、Liferay コンサルタントと共に働くことによって、幅広い製品と開発の知識を習得できます。また Liferay DXP の使用方法を最適化して、既存および将来の IT 投資を最大化します。

参考文献

- [開発時チェックリスト\(7.1版\)](#)
- [パフォーマンスチェックリスト](#)



Liferay は、さまざまなデバイスを通して Web のデジタル体験を創造するソフトウェアを提供しています。Liferay のプラットフォームはオープンソースがもたらす革新性と併わせ、高い信頼性とセキュリティを兼ね備えています。我々はビジネスとテクノロジーによって、世界に優れた足跡を残すことを目指し日々活動しています。

Liferay の製品は、金融、ヘルスケア、政府・官業、保険、小売、製造など、さまざまな業界の企業・団体のお客様に採用されています。詳細は、liferay.co.jp へアクセスしてください。

© 2019 Liferay, Inc. 無断複写・転載を禁ず。