

アジャイル・インテグレーションによる デジタル革新

アジャイル・インテグレーションとその必要性

はじめに

あらゆるものが変化しています。フォーチュン・グローバル 500 などのランキングにもよく表れているとおり、今は企業レベル・業界レベルどちらにおいても重要な転換期です。20 世紀においてこのような急激な時代の変化は、テクノロジーの変化と資本の拡大によってもたらされてきました。¹ 従来のマーケットセグメント間で直接的な競争が存在することは明らかですが、現代における破壊的デジタル革新は、新たな分野で競争力を発揮し、利益を得るチャンスも生み出しています。たとえば、Netflix のような動画ストリーミングサービスは、サービス運営のために作成したソフトウェアを巡るコミュニティをも生み出しました。また、Amazon のようなオンライン小売業者は、パブリッククラウド管理の点でもイノベーションをもたらしています。イノベーションとは、見栄えの良い顧客向け UI のことだけではなく、組織の柔軟性を高め、既存のノウハウを活用し、新しいアイデアを取り込むためのテクノロジー、プロセス、およびカルチャーの基盤がなければなりません。

戦略レベルで言えば、現在のソフトウェアには、ビッグデータ、アプリケーション・プログラミング・インタフェース (API) や、モノのインターネット (IoT) といった取り組みから、オムニチャネル・エクスペリエンスに至るまで、多岐にわたるさまざまな新しいビジネス目標に基づく提供が期待されています。さらに、ソフトウェアは複数のビジネス機能、ビジネスモデル、エンゲージメント・チャネル、そしてステークホルダー・エコシステムを通じて動作することが求められておりながら、かつてない速度で変化やイノベーションに対応しなければなりません。企業はソフトウェアやハードウェア・システムに何を期待するのでしょうか。市場の新たな現実に適応し、ビジネスチャンスを収益化するため、効率的かつダウンタイムを招くことなく対応する能力を求められています。価格設定を一晩で変更したり、世界中の顧客やスタッフに新しい製品を迅速に提供したりすることができる企業は、何段階にもわたる手動の検証手順を経て 3 か月後にやっと公開に至る企業に比べて、圧倒的に有利です。

さまざまなビジネス目標を実現し、競争力の高いサービスを提供するには、エンタープライズ・インテグレーション (アプリケーションやデータを統合する能力) が鍵となります。デジタル・イノベーションや破壊的デジタル革新が広がるにつれ、より困難な新しい要求が増え、旧来のアプローチでは対応が難しくなりつつあります。社内業務のワークフローや顧客とのやりとりは、今でも基幹システム (SoR) の関連 IT インフラストラクチャに依存しています。しかし、確実な社内ソリューションを提供するための時間短縮は、格段に困難になりました。クラウド・アプリケーションやハイブリッドクラウド IT 環境がますます普及していることに加え、パートナーや顧客にリーチするためにシステムを拡張する必要があるなど、新たな課題に対応するため、モダン・アプリケーションへの需要が生じています。このため、エンタープライズ・インテグレーションや、サービスの迅速提供と継続的デリバリーの重要性がますます高まっています。これらの急増する新たな課題に対処するために Red Hat が提案する方法は、さまざまなアプリケーションや情報システムを統合するアジャイル・インテグレーション戦略です。



facebook.com/redhatjapan

@redhatjapan

linkedin.com/company/red-hat

jp.redhat.com

1 <http://www.kauffman.org/what-we-do/research/2012/06/what-does-fortune-500-turnover-mean>

アジャイル・インテグレーション

アジャイル・インテグレーションとは、アーキテクチャに関わるアプローチです。アジャイルな手法やプラクティスをさまざまなテクノロジーと組み合わせ、柔軟かつ適応性に優れた統合ソリューションに適したプラットフォームを使用することにより、アプリケーションやデータの迅速な統合を目指します。

アジャイル・インテグレーションとは

Red Hat における「アジャイル・インテグレーション」は、他とは若干異なります。一般的な意味でのアジャイル・インテグレーションは、継続的インテグレーションや継続的デリバリー (CI/CD) と関連しており、さまざまな開発プロセスを統合、つまり組み合わせ、1つの連続的なプロセスにすることを表します。Red Hat の定義するアジャイル・インテグレーションは、「統合 (インテグレーション)」のテクノロジーとプロセスに焦点を当てたアーキテクチャのアプローチを指します。このアプローチでは、アジャイルな方式と柔軟なマイクロサービス・アーキテクチャを活用することで、複数のシステムやサービスにわたってアプリケーションとデータを統合します。これにより、急速に変化するデジタルビジネス上の需要に対し、迅速に適応できるようになります。アジャイル・インテグレーションの実現には、スピードと適応性に優れたソリューションのための最新型プラットフォーム、プロセス、およびテクノロジーが不可欠です。アジャイル・インテグレーションのアプローチは、お客様が自社のインテグレーション・サービスを CI/CD プロセスに組み込むうえで役立ちます。

アジャイル・インテグレーションを目指すべき理由

エンタープライズ・サービス・バス (ESB) など従来のインテグレーション・テクノロジーは、さまざまなアプリケーションを統合および接続するために必要な機能を提供します (変換、ルーティング、オーケストレーション、接続性など)。サービス指向アーキテクチャ (SOA) などのアーキテクチャ・パターンを ESB と組み合わせることで、インテグレーション・ロジックを再利用可能なサービスとしてカプセル化するためのプラットフォームが実現します。SOA では、モジュール式のビジネス機能や、これらのサービスの再利用を可能とする機能が活用されます。しかし、テクノロジーとガバナンスが複雑であること、実装サイクルに時間がかかること、アジリティよりも再利用可能性が重視されることなど、SOA には課題もいくつか存在します。

何年にもわたって多くの企業では、モノリシックなアプリケーション相互運用性の問題を緩和するため、指数関数的に増え続けるアプリケーション同士の接続数への対応を試みてきました。接続数は、企業内のアプリケーションの数を x とすれば $x^2(x-1)$ で表わされます。この問題に対する解決策は、すべてのアプリケーションを統合して単一のエンタープライズ・サービス・バスを実現することでした。しかし、多数での接続という問題が解消されることはなく、それらの接続の複雑さを ESB という箱の中に閉じ込めるだけに留まりました。ESB は垂直方向にのみスケーリング可能ですが、それ自体がモノリシックなアプリケーションとなりました。このアーキテクチャの場合、ESB 内の接続を制御するための一元管理が必要となりました。すべてのアプリケーションを1つの一元的な「接続ボックス」に押し込んでも、複雑さを軽減するという目的は果たされませんでした。このソリューションは、新規アプリケーションを開発する際のアジリティを低下させていたのです。一元的な ESB アーキテクチャや、それに関連する開発プロセスでは、新しいサービスの構築と修正や、新サービスを通じたイノベーションの実現が困難でした。マイクロサービス・アーキテクチャの出現により、この問題の複雑性は指数関数的に増大しました。

マイクロサービス形式のアーキテクチャでは、アプリケーションの機能を単独でデプロイ可能なサービスとして設計・開発することにより、よりアジャイルなアプローチが実現します。マイクロサービス・アーキテクチャは、アジャイルなビジネスシステムの構築を容易にし、ビジネスが素早く変化できるようにすることで、新機能の構築、実験、ディスラプションへの対応を強化します。しかし、アプリケーションを分割して多くの別個の小さなサービスにするため、変換、オーケストレーション、接続などのインテグレーション機能がやはり必要です。マイクロサービスの開発にインテグレーションは不可欠ですが、一元的な ESB テクノロジーとデプロイメント・アーキテクチャでは、アジャイル開発プロセスのためのマイクロサービス・アーキテクチャを実現できません。そのため、別のインテグレーション・アプローチが必要です。

アジャイル・インテグレーションには、より適応性に優れたソリューションを提供できるプラットフォーム、プロセス、およびテクノロジーが不可欠です。アジャイル・インテグレーションを採用すると、マイクロサービス・アーキテクチャを含むアプリケーション開発プロセスにインテグレーションを含めることが

でき、アジリティがさらに高まります。新たな革新的ソリューションのシステム提供を担当する分散型のチームにとって、インテグレーションは重要な機能となります。さまざまなテクノロジーが提供する機能を、企業やプロセスに関する手法と組み合わせることによって、真の革新が実現可能となります。

アジャイル・インテグレーションの実現

アジャイル・インテグレーションのアーキテクチャ手法には、以下の3つの主要な機能が必要です。



1. 分散インテグレーション：ビジネス要求の変化に対応するための柔軟性

デジタルチャネル（モバイル、ソーシャル、メッセージング、Web）を利用するユーザーが増えるにつれ、ソフトウェアはますますユーザーファーストのモデルに移行していき、Push 型よりも Pull 型の機能やサービスが重視されるようになりました。加えて、ユーザーフレンドリーなソフトウェアツールやクラウドサービスが容易に入手できるようになったことで、企業の IT 部門の役割が変化しました。一元的な管理や監視よりも、コラボレーションや、ビジネスを強化するための貢献が求められるようになったためです。また、ビジネス上の目標実現において最適なテクノロジーの選定を促すため、テクノロジー関連の意思決定において、ビジネスラインがさらに重視されるようになりました。

このような市場レベルと組織レベルの取り組みの結果、IT 部門は、よりモジュール性と分散性の高いモデルを採用すると同時に、セキュリティやガバナンスといった中核的ビジネス要件も引き続き遵守する必要に迫られています。エンタープライズ・インテグレーションのベストプラクティスにおいて事実上中心的存在だったインテグレーション・コンピテンシー・センターは、今ではより分散型のインテグレーションやビジネスを重視したモデルへと進化しつつあります。IT 部門内では大規模なインテグレーション・チームよりも、より柔軟かつアジャイルな対応が可能な、小規模なチームが重視されるようになりました。

より多くのものが相互接続された世界へと向かう中、新しいサービスやアプリケーションをより迅速かつシンプルに統合することが求められています。そのためには、よりモジュールで軽量な、パターンベースのアプローチにより、アジャイル・インテグレーションを促進する必要があります。フロントエンドやクライアントサイドの要件が変化する中で、バックエンド・インテグレーションにおいても同等の柔軟性が求められます。しかし、このような軽量なアプローチによるインテグレーションは、必ずしも簡単ではありません。従来の ESB は多くの場合、プロプライエタリー・テクノロジーの上に構築されており負荷も高く、他のモノリシックなアプリケーションと同様の欠点がありました。

オンプレミスでもクラウドでも、複数のエンタープライズ・システムにわたり迅速なインテグレーションを可能とする、軽量で柔軟性に優れたインテグレーション・プラットフォームが必要です。このようなインテグレーション・プラットフォームにより、開発者は短時間で API ベースの軽量なインテグレーション・サービスを構築し、必要な場所にデプロイし、適宜スケールアップできるようになります。

コンテナ

コンテナは、アプリケーションをランタイム環境全体と共にパッケージ化して分離するテクノロジーです。すべての依存関係、ライブラリ、実行に必要な設定ファイルのすべてを、便利な1つのパッケージとしてまとめることができます。これにより、ソフトウェアを信頼性の高い方法で、複数のコンピューティング環境にわたって実行することが可能になります。

たとえば、大手小売ブランドが、自社が大きなシェアを占める従来の販売チャネルで商品を販売しているとした場合、この小売業者は、競合のオンライン業者が起こした破壊的革新の影響を受け、従来のビジネスモデルを温存しながら状況に適応した各種デジタルチャネルを開拓する必要に迫られます。そのために、小規模なチームをいくつか結成します。これらのチームの規模は、4～7人程度に抑える必要があります。Amazon の CEO である Jeff Bezos は、この規模を「ピザ 2 枚で足りる程度 (Two Pizza Rule)」と表現しています。このように小規模なチームは、より軽量のインテグレーション・プラットフォームと API を活用することで完全なエンドツーエンドのアプリケーションを作成し、業務へ素早く対応することができます。

2. コンテナ：需要に合わせたスケーリング

最新のアプリケーションでは、何十万あるいは何百万というトランザクションに対応可能な規模へのスケーリングが求められることが少なくありません。このような需要はしばしば予定外で発生し、伸縮自在なスケーリングが求められます。これらのアプリケーションは多くの場合、適切なデータをオンデマンドで提供するため、それぞれが互いから独立してスケーリングを行う必要があります。また、デジタル分野における需要に迅速に対応するため、しばしば継続的な更新と開発が必要となります。さまざまなものが相互接続され、データ中心となっていく世界で成功を収めるには、基盤となるインフラストラクチャのスケーラビリティと管理、データアクセスの速度、そして継続的な開発とデリバリーのサイクルが重要となります。これを実現するには、コンテナ・テクノロジーを活用するアジャイル・インテグレーションが重要な役割を果たします。

コンテナは、1 つ目のアジャイル・インテグレーション機能 (分散インテグレーション) の項で説明された分散インテグレーション・モデルを支えるものです。なぜならコンテナは、一元的な ESB の柔軟性に欠くモノリシック・アーキテクチャに伴うボトルネックを回避できます。コンテナを利用することで、必要に応じて独立してスケーリングでき、軽量かつ十分にテストおよび実証済みのデプロイ単位で構築することができます。また、複雑なシステムを、より小さな動作単位 (マイクロサービス) へと分割するための優れたパラダイムともなります。サービスをコンテナにデプロイすることで、互いに独立した複数のチームがデプロイ作業を行えるようになり、デリバリーが加速されます。さらに、コンテナを使用することにより、システムを構成する個々の要素において、一貫したバージョン管理やスケーラビリティの実現が可能となります。このすべては、インテグレーションにおけるアジリティを高める強力な基盤となります。

3. API：接続と再利用を効率化

Web やモバイルに対応したソフトウェアが普及するにつれて、企業、パートナー、顧客がビジネスプロセスとどのようなかたちで関わっているかということは、市場における重要な差別化要因となります。このようなエンドポイントとユーザー・インタフェースが増えていくにつれて、ポイントツーポイントのインテグレーションを維持することは困難となります。代わりに、ビジネス資産 (つまり IT システム、社内および社外の人材、クライアント・アプリケーション、および顧客) 同士をつないで潜在的価値を最大化するための手段として、API の採用が広がりつつあります。

API には、従来型の大企業がアジリティを高め、小回りのきく小規模な会社のように機能することを助けるというメリットがあります。反対に、新興企業にとって API は、新たな分野で迅速に自社の存在感を高めるための手段となります。API はインテグレーションの複雑さを軽減し、アプリケーション制作を加速することにより、社内のイノベーションを促進し、新しい顧客を取り込み、製品やサービスを拡張し、活発なパートナー・エコシステムの形成を促します。

通常、企業は API を公開すると、社内外の開発者、パートナー、顧客に対して統一された形のデータやトランザクション・インタフェースを提供できるようになり、データへのアクセスやトランザクションが向上します。また、API にアクセスするためのソフトウェア・アプリケーションを開発することにより、自

API

API は、組織があらゆる人とデジタルかつプログラマチックにやり取りする機能を表します。

API を使用することで、組織外の人がコードを作成してデータにアクセスしたり、リモートリソースを制御したり、トランザクションを処理したりできるようになります。

社や社会のためとなる新しい機能や価値の創出につなげることができます。これにより、ビジネスの在り方や運用方法を変革する可能性を秘めた、さまざまな新種のアプリケーションを作り出すことが可能となります。

API はバックエンドシステムから安全かつ継続可能な方法でデータを開放することにより、アプリケーション開発において以下のような重要な役割を果たすことができます。

- 新たな収益源の創出 (アクセス課金や e コマース化など)
- 既存サービスを新しいデバイスやプラットフォーム経由で提供することにより、リーチや顧客価値を増大
- バックエンドシステムを変更することなくアイデアの実現が容易となり、企業の技術革新を促進
- API ベースの統合されたアプリケーションを、アジャイルなチームにセルフサービスで提供

これらの 3 つの機能 (分散インテグレーション、コンテナ、および API) は、IT インフラストラクチャの敏捷性向上を目指す取り組みを支えるものです。というのも、これらの機能がそれぞれ抽象度を上げることによって、さまざまなチームの協業が可能となるためです。API とコンテナはどちらも、特定の重要なアセットを、広く認識されるレベルでパッケージ化します。インテグレーションを、必要に応じた個別の「インテグレーション」が分散して配置されているように扱う場合、個々の「統合」箇所は、インフラストラクチャ全体における重要な部分であると言えます。

RED HAT のアプローチ: お客様事例

KEYBANK

米国の銀行系金融サービス会社 KeyBank は、複雑な規制やセキュリティ要件を遵守しながらデジタルチャネルにおけるビジネスを拡大するため、デジタルチャネルのモダナイゼーション・プロジェクトに着手しました。Red Hat は、コンテナベースのアプローチの導入支援を通じて、KeyBank の目標達成を次のようにサポートしました。

- アプリケーションをインフラストラクチャから分離することにより、複雑性を軽減
- DevOps プロセスを推進し、開発チームが設定と管理に費やす時間を最小化することで、開発業務を促進
- 顧客情報のセキュリティ強化およびコンプライアンス支援

SCHIPHOL グループ

空港を運営する Schiphol グループは、所有する膨大なデータをもとに、旅行者へ重要な情報を提供しています。その背景にある主なビジネス目標は、同グループの所有するデータから、空港に関する正確で高品質なリアルタイムの情報を抽出して、社内外の幅広い関連パートナーへ提供し、卓越したカスタマーエクスペリエンスを実現することでした。

Schiphol グループは、インテグレーション、API 管理、およびコンテナに関する主要テクノロジーとして Red Hat® JBoss® Fuse、Red Hat 3scale API Management Platform、および Red Hat OpenShift Container Platform を選択し、既存のエンタープライズ・サービス・バスから置き換えたほか、RESTful API を通じてサービスを公開し、マルチクラウド・プラットフォームを作成しました。その結果は次のとおりです。

- パートナー統合のための開発コスト削減
- パートナーを通じた収益源の増加
- パートナーと共有する機能やデータの可視性と制御の強化

まとめ

デジタルサービスの時代にあって、エンタープライズ・ソフトウェアは多くの新たなビジネス目標を満たす必要があります。そのためには、ビジネス上の成果を向上して競争力を保つ手段として、モバイル、ビッグデータ、IoT（モノのインターネット）、クラウドなど、さまざまな革新的テクノロジーを取り入れなければいけません。加えて、これらのテクノロジーの影響で変化のスピードが加速していく中、現代のビジネスで成功を収めるには、複数の情報源からデータを迅速に統合する能力が不可欠であることが明らかです。

ESB など従来型のインテグレーション・テクノロジーには技術的な複雑さがあり、実装サイクルも長いため、現在のビジネスに必要とされるアジリティの実現が困難でした。柔軟でビジネス変化へ適応できるインテグレーション・ソリューションに特に適したプラットフォームを基盤とする、アジャイル・インテグレーションならば、既存のテクノロジーと最新テクノロジーを最大限に活用できます。適切なツールとアジャイルなプロセスによってサポートされるアーキテクチャの手法には、次の3つの主要な機能があります。

- 分散インテグレーション：より迅速な適応を可能とする柔軟性を提供
- API：より効率的な接続性を実現
- コンテナ：需要に応じたスケーリングに対応

これらの機能により企業は、従来型のインテグレーション・テクノロジーと手法から最新のアジャイル・インテグレーション・モデルへと進化し、デジタルなビジネス変革を目指せるようになります。

RED HAT について

オープンソースソリューションのプロバイダーとして世界をリードする Red Hat は、コミュニティとの協業により高い信頼性と性能を備えるクラウド、Linux、ミドルウェア、ストレージおよび仮想化テクノロジーを提供、さらにサポート、トレーニング、コンサルティングサービスも提供しています。Red Hat は、お客様、パートナーおよびオープンソースコミュニティのグローバルネットワークの中核として、成長のためにリソースを解放し、ITの将来に向けた革新的なテクノロジーの創出を支援しています。

アジア太平洋
+65 6490 4200

インドネシア
001 803 440224

ニュージーランド
0800 450 503

ベトナム
800 862 6691

オーストラリア
1 800 733 428

日本
03 5798 8510

フィリピン
800 1441 0229

中国
800 810 2100

ブルネイ / カンボジア
800 862 6691

韓国
080 708 0880

シンガポール
800 448 1430

香港
852 3002 1362

インド
+91 22 3987 8888

マレーシア
1 800 812 678

タイ
001 800 441 6039

台湾
0800 666 052



facebook.com/redhatjapan
@redhatjapan

linkedin.com/company/red-hat

jp.redhat.com
#f7931_0817