

# グローバル企業が取り組む「攻めのIT経営」 —クラウドネイティブなシステム開発のポイント—



「攻めのIT経営」にはデータ活用・データ分析の高度化が不可欠である一方、ビジネス環境が不確実性を増しているため、システム化に際しては俊敏さや柔軟さが大切になる。本稿では、日本のグローバル製造業による、クラウドサービスを全面的に採用したクラウドネイティブなシステムの事例を紹介し、そのメリットや課題について考察する。

野村総合研究所 生産革新本部  
福岡ソリューション開発部 上級アプリケーションエンジニア

いとう しんじ  
伊藤 真二

専門はクラウドコンピューティング、サーバーレスアーキテクチャー

## グローバル製造業の課題

言い古されたことではあるが、近年の消費者の嗜好やニーズの変化は非常に速い。ソーシャルメディアの普及も、これに拍車をかけている大きな要因だ。こうした変化に対応するために、企業は在庫・販売計画のリアルタイム化を必要としており、米国Amazon社のように巨大な流通システム改革を推進する企業も多い。サービスを定額制で利用するサブスクリプションモデルやシェアリングサービスも、変化に対応する手段と考えることができる。

一方で、米国の電気自動車メーカーTesla社のように、大量の需要に生産が追い付かない「生産地獄」（同社イーロン・マスク最高経営責任者の言葉）に陥る企業も出てきている。特にグローバル企業にとって、需要の把握とタイムリーな設備投資、技術革新は重要な課題である。顧客ニーズの把握や新技術の取り込みに失敗し、需要に的確に応えられない企業は、ビジネスチャンスを逃したり顧客の信頼に応えられなかったりすることで勢い

を失っていく。今、日本企業がデータ活用の高度化を中心としたデジタルトランスフォーメーション（DX）を求められている大きな理由の1つがそこにある。

## 需給管理高度化の取り組み

ところで、日本企業の勢いが失われた例としてモバイルインターネット市場の話がよく引き合いに出される。日本企業は1990年代から、NTTドコモの「iモード」のような世界に先駆けたモバイルインターネットの優れたサービスをつくり、顧客の期待に応えることで業績を伸ばしていった。しかし、その後のグローバル化、スマートフォン化という世界的な流れに乗り遅れてしまったことで、米国のApple社やGoogle社はもちろん、中国や韓国の企業にも水を空けられるようになった。製品自体の差別化が難しくなった家電製品やPCでも、日本企業は勢いを取り戻せないままである。

そのような状況のなかで、自動車やオートバイでは日本ブランドが依然として高い信頼

を勝ち得ており、海外でも大きなマーケットシェアを獲得している。日本企業は海外生産を拡大しているので必ずしも日本製とはいえないが、それでも世界中の顧客が日本ブランドを買い求めている。

以下でシステム開発の事例を取り上げるヤマハ発動機は、売上げの約9割が海外、オートバイの売上げは世界規模であり、特にアジア諸国ではトップレベルのグローバル企業である。このような実績を支えているのが、顧客満足度を高めるために同社が積極的に取り組んできた需給管理の高度化である。

この取り組みの背景には次のような事情がある。1980年代を販売のピークとした日本のオートバイは、長く少品種大量生産であったが、消費者の嗜好の多様化により、今は多品種少量生産が求められるようになった。これは世界的な傾向であり、商品を生産して売るだけといった販売手法から、顧客が本当に買いたいものを先読みして生産するプロセスへの転換が求められているのである。

ヤマハ発動機では、生産・在庫・販売を一貫して管理するための「デマンドチェーン革新部」が2016年に設置された。デマンドチェーン（需要連鎖）とは、最終顧客の需要を起点とした一連の活動のことで、情報、モノ、金という3つの流れの計画と実践を意味する。需要の変動に対して一連の活動の速度や柔軟性を高めて収益を改善する経営手法がデマンドチェーンマネジメントである。「デマンドチェーン革新部」は、工場、物流、営業から小売りまでをつなげて、顧客の細かいニーズに応える品ぞろえを実現することを任務としている。

2017年には、野村総合研究所（NRI）が支援して、需給管理のシステム化計画が策定された。この計画には、海外販売の中で特に重要なアジア拠点での新しいPOSシステム（以下、新システム）の導入が含まれる。新システムは需給調整の最適化を目指したもので、これは多くのグローバル企業が直面している課題でもある。その他、新システムでは、これまでではできなかった、海外の販売ディーラーの接客に関するKPI（重要業績指標）の分析が可能である。すなわち、顧客の来店から、接客、契約に至るまでの情報をデータとして取得し、データアナリティクス機能で分析することである。

## 新システムの特徴

ここでは、新システムの特徴を、開発方法とアーキテクチャーを中心に紹介する。

### (1) アジャイル開発とクラウドネイティブ

グローバルな情勢の変化に素早く対応するために、新システムの開発期間は3カ月とされ、その短期間で「販売店向けモバイル・タブレットアプリケーション」「POSシステムが利用するマスターデータやユーザー情報などを管理するWebアプリケーション」「接客・販売・在庫などの情報を分析するデータアナリティクスアプリケーション」の3つを提供することが求められた。また、開発開始時にはシステム要件が十分に固まっておらず、開発中の機能を確認しながらプロジェクトを進めることも必要だった。

このため、新システムの開発に当たっては、リリースのスピードを優先しつつその後

の改善を積み重ねていくアジャイル開発の手法を採用した。

DXのためのシステムでは、要件の変化などに応じて柔軟に変更することや、時には計画そのものを見直すことも必要になる。今回の新システムの開発でも、高度な分析（データアナリティクス）機能の追加、関係サーバーの能力強化、ネットワーク環境の改善など、顧客要件の変更に柔軟に対応しなければならなかった。

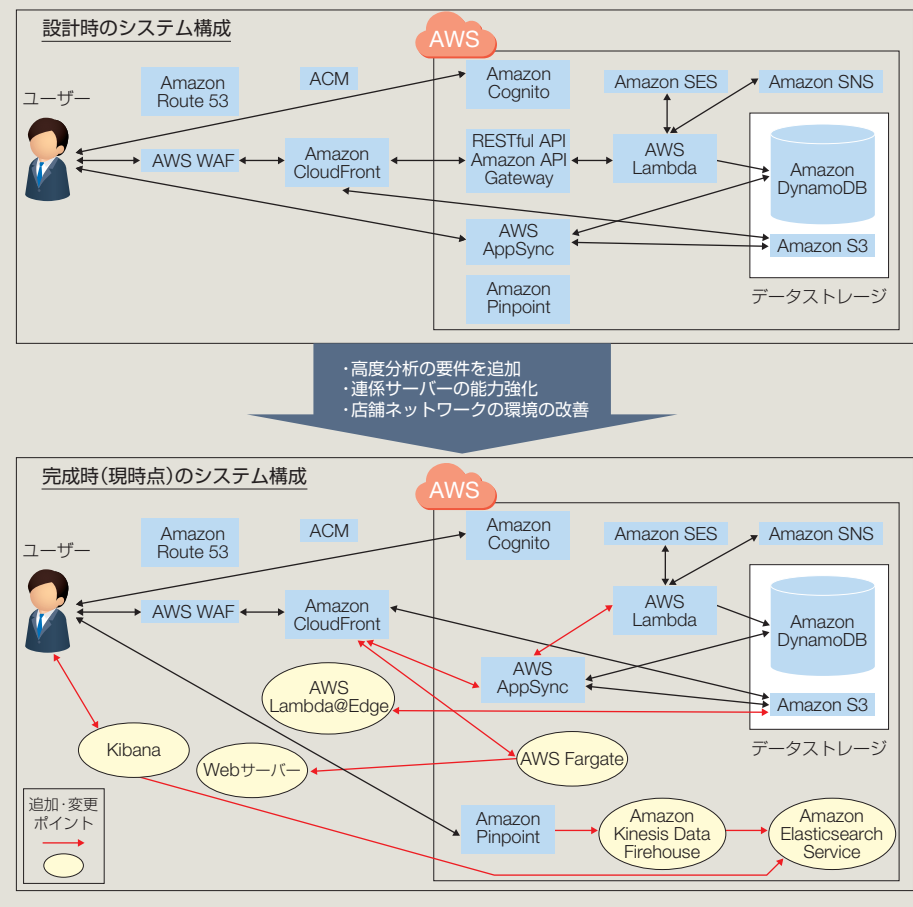
このような理由により、アジリティー（俊

敏さ）とコスト効率に優れたクラウドネイティブ（クラウドサービスの利用を前提にした設計）のアーキテクチャーを採用することとした。次々に出てくる顧客のアイデアを取捨選択しながら最適なサービスを構築したり、ビジネスの規模に応じて柔軟にリソースを増減したりできる点は、スケーラビリティに優れたクラウドネイティブならではの強みである。（図1参照）

## (2) マネージドサービスの利用

クラウドネイティブな新システムの特徴は、サーバーレスであることに加えてマネージドサービス（サービスの利用に必要な機器の運用・管理・設定などをまとめて提供

図1 要件の追加・変更に対応可能なシステムを実現



するサービス）を全面的に利用している点である。この方式の主な利点は、①サービスリリースまでの速さ②スケーラブルでバースト（短時間での処理）可能なシステム基盤リソースの確保③利用に応じた低コストな課金体系④セキュリティが高く自動化された運用環境⑤DevOps（開発と運用が密接に連携して開発する体制）による運用保守の自動化である。

今回の新システムでは、AWS（Amazon Web Services）社のさまざまなサービスの機能を外部のアプリケーションから利用できるようにするために、新しいAPI（アプリケーション・プログラミング・インターフェー

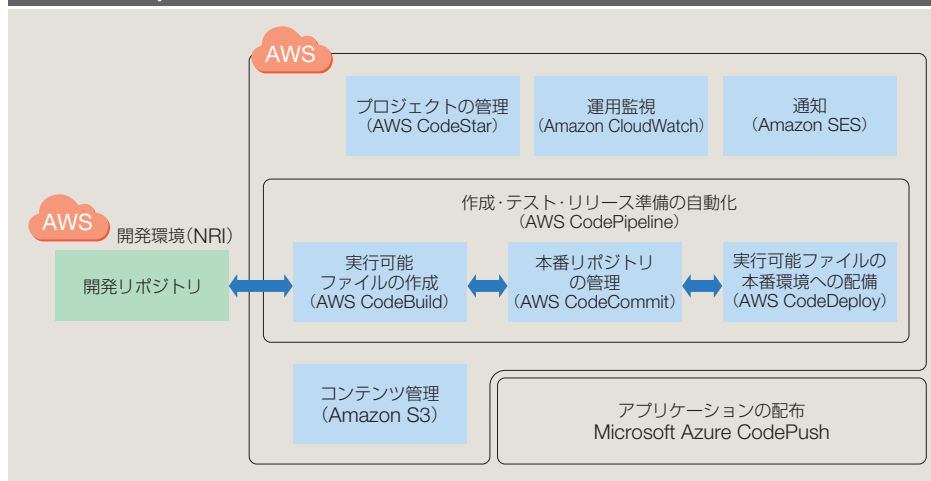
ス。汎用的なプログラムの機能を利用するための仕様) サービスであるAWS AppSyncを採用している。これによりアプリケーションデータの同期やアプリケーション構築が容易に行えるようになる。

AWS AppSyncは、次世代のAPI規格であ

るGraphQLを利用したマネージドサービスである。WebアプリケーションのデータベースとAPIの統合処理はほぼコーディングなしで実現でき、バックエンド開発の生産性は劇的に向上する。開発者はGraphQLのデータ定義さえ行えば、APIやデータアクセスなどを実装する必要もない。データストアには、スケーラブルでクラウドネイティブなシステムとの親和性が高いAmazon DynamoDBを利用するのが標準的だ。このように、バックエンド開発はサーバーレスどころかコードレスな開発スタイルに変わりつつある。

AWSのAPIサービスでは、Amazon API Gateway (API呼び出しの処理、トラフィック管理、認証・アクセス制御など) がよく利用されている。今回も、当初はこれを使うことを検討した。その場合には、Amazon API Gateway、AWS Lambda (ビジネスロジックの実行)、Amazon DynamoDB (データベース) の3階層でバックエンドを構成し、AWS Lambdaをビジネスレイヤーとして利用するのが一般的だ。しかし、新システムは当初、外部関係も持たない設計で、ほとんどのビ

図2 DevOpsツールによる運用の自動化



ジネスロジックはフロントエンドで保持できる。そのため、開発の効率化を優先させてAWS AppSyncを採用したわけである。

### (3) 自動化されたシステム運用

新システムはAWSのDevOpsツールを全面的に採用している(図2参照)。アプリケーションの作成から配備を担うのは、AWS CodeCommit (ソースコードを格納したりリポジトリを管理)、AWS CodeBuild (ソースコードからアプリケーションを作成)、AWS CodeDeploy (アプリケーションの配備を自動化)、AWS CodePipeline (ソースコード変更時のアプリケーション作成・テスト・リリース準備の自動化) である。

NRIの環境と顧客の本番環境が連携され、アプリケーションのリリースから配備、運用までが自動化されており、ソースコードが更新されるたびに自動でテストが行われ、エラーがあれば開発者に通知が届く。このように完全に自動化された環境で運用されるため、ハードウェア障害への対応やアップデートなどとも無縁で、開発の品質は高いレベルで維持される。



この他、アプリケーション開発プロジェクトの管理はAWS CodeStar、AWSのリソースと実行されているアプリケーションのモニタリングはAmazon CloudWatch、メール送受信のプラットフォームはAmazon SES(Simple Email Service) がそれぞれ担う。コンテンツの管理にはストレージサービスのAmazon S3を採用している。これらのサービスにはテンプレート(ひな形)が準備されており、設定を簡単にカスタマイズできるため、導入と学習のコストが低く再利用性も高い。アプリケーションの配布にはMicrosoft Azure CodePushを利用している。

開発や運用のノウハウは属人化することなくDevOpsの一部として管理され、ドキュメント作成や新規メンバーの学習負荷も軽減される。新システムでは、開発からリリース・運用までのプロセスでマニュアル操作が全く必要ないNoOpsを実現している。これにより、クラウドのマネジメントコンソール(管理画面)から本番環境にアクセスする作業が不要となり、セキュリティが向上するメリットも大きい。

なお、クラウドベンダー各社は標準的なDevOpsのサービスを提供しており、各社の提供機能にほとんど違いはない。

#### (4) AWS社との技術連携

一般に、新しいシステムの開発で全く経験のない技術を採用することには大きなリスクが伴うが、NRIはクラウドサービスの黎明期からこれを活用したアジリティのある開発に取り組み、新しいクラウドサービスの導入時にもAWS社と技術連携をしながらプロジェクトを推進してきた。その実績が評価さ

れて、NRIは日本で最初にAWSプレミアムコンサルティングパートナーに認定されている。新システムでも、新サービスの技術評価からAWSの東京データセンターでのサービス開始まで、AWS社と密に連携してフィードバックを積み重ねた。AWS AppSyncは新システムの本番稼働に合わせて正式にサービスが開始されたが、これはNRIとAWS社が連携してきた成果である。

### クラウドネイティブの課題

クラウドネイティブの課題も見えてきた。特にクラウドサービスのフロントエンドライブラリー(クラウドベンダーが提供する開発部品)は成熟度が低く、新システムの開発でもトライアンドエラーを行う必要があった。また、ライブラリー自体のバージョンアップが頻繁に行われ、不具合対応のために新バージョンを取り込むと今度は別の問題が発生するといったことも起きた。そのため、開発者には、OSS(オープンソースソフトウェア)になったライブラリーのソースコードを調査しながら不具合の対応方法を検討するという開発スタイルが必須である。

クラウドネイティブに精通したITアーキテクトを確保することも課題である。クラウドネイティブでは、アプリケーション開発の中心はモバイルやWebなどのフロント開発であり、国内外で開発者が多いため開発コストとスケジュール調整の問題はそれほど大きくない。問題は、クラウドネイティブなシステムはさまざまなサービスをピースとして組み合わせ、パズルのように設計していくこと

が多く、時には新しいサービスを使うこともある点だ。また、モバイルやAPIのアプリケーション要素だけでなく、AIや機械学習、IoT、データレイクなども含めたトータルソリューションとしてシステムを設計する能力も求められる。これに応えられるITアーキテクトは不足している。

また、これはクラウドネイティブ自体の問題ではないが、ネットワーク環境にも注意する必要がある。今回の新システムでは、日本では考えられないようなネットワークインフラへの過負荷が原因の不具合もあった。ネットワークへの負荷は、画像データやWebコンテンツのサイズを最適化するAWSのサービスを利用することで軽減する方法もあり、今回のシステムでも採用している。いずれにせよ、クラウドサービスの利用に際しては現地での入念なテストが欠かせない。

## これからのDXシステムの開発

DXのためのアプリケーション開発においては、次の2つの流れが明確になっている。

1つは、クラウドネイティブへのシフトである。DXを加速させるためには、システム基盤や運用はクラウドサービスに任せ、アプリケーション開発にリソースを集約させることが不可欠である。従来のモノリシック（一枚岩的）なサービスは、スケーラビリティと可用性・拡張性に優れたマイクロサービスアーキテクチャー（サービスを機能単位に小さく分割し、それらを関係させる方式）へと置き換えるべきである。ただし、IoTシステムやAIのエッジデバイス（ネットワークに

直接接続される機器）の開発と同様に、クラウドサービスの中でもフロントエンドの技術は発展途上であり、まだまだ安定しているとはいえない。そのため、いかにシステムベンダーとクラウドベンダーを巻き込んで問題に対応できるかがポイントである。

もう1つはDevOpsの導入である。これにより、DevOpsエンジニアという新しい領域の技術人材が必要になっている。DevOpsエンジニアは、開発プロセスとリリース・運用までの自動化を担当し、ソースコード以外のクラウドリソースを意識することなく開発できる環境を提供する役割を持つ。NRIはそのようなDevOpsエンジニアの人材育成にも力を入れている。

ヤマハ発動機は、リリースされた新システムを使って、海外のディーラーの在庫・販売状況、成約に至るまでのプロセス、製品比較の状況など、以前はディーラーや営業担当に聞くなどして得るしかなかった情報をデータとして取得し、そのデータを分析して新しい「気付き」を得られるようになった。「気付き」は次の機能改善のポイントであり、グローバル企業が長年悩んできた需給計画の最適化につながる。そしてシステムはつくれば終わりではなく、「気付き」に基づいた改善を積み重ねていく必要がある。

DXは未知のものへの挑戦であり、リスクもあるが、クラウドネイティブとすることでリスクが最小化され、挑戦を繰り返すことができるようになる。NRIは、これからもクラウドネイティブなシステムの開発を通じて、お客さまと共に「攻めのIT経営」の実現に挑戦していきたいと考えている。 ■