

事業拡大 ビジネススピードを加速するIT基盤(第14回)

## クラウドネイティブ時代のアジャイル開発事始め

2018.04.06 Fri

アジャイル開発やマイクロサービス、PaaSにコンテナと、現在のソフトウェア開発の現場では新たなキーワードが飛び交っています。これらの用語を解説しつつ、クラウド環境を最大限に生かしたクラウドネイティブ時代のソフトウェア開発について紹介していきます。

### アジャイル開発が注目される理由

とりわけIoT

やデジタル・トランスフォーメーションにかかわるソフトウェア開発の現場において、頻繁に聞かれるキーワードの1つに「[アジャイル](#)」があります。これは1990年代後半に提唱され始めたソフトウェアの開発手法であり、従来の「[ウォーターフォール](#)」型のソフトウェア開発の対極に位置するものと捉えられます。

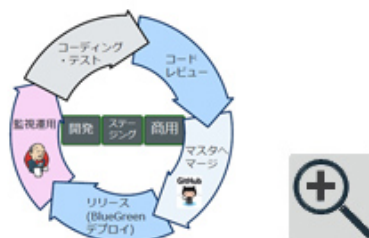
要件定義から基本設計、詳細設計、プログラミング、各種テストといった工程でプロジェクトを進めるウォーターフォールは、プロジェクトの計画を立てやすいこと、進捗管理を行いやすいことなどがメリットとして挙げられます。しかし難点となっているのは、プロジェクトの最初期の段階で要件が固まっていた、それが変わらないことが前提となっていることです。もしプロジェクトの途中で要件が変わるといことになれば、膨大な手戻りコストが発生し、場合によっては大幅なプロジェクトの遅延といった事態を招きます。

これに対し、IoTやデジタル・トランスフォーメーションの領域はビジネス環境や顧客ニーズの変化により、要件が変わることが珍しくありません。たとえばソフトウェア開発に1年間かかると考えた場合、1年後のビジネス環境や顧客ニーズを正確に捉えて要件を定義することは至難の業でしょう。こうした課題に対応する上で、最適な開発手法として注目されているのがアジャイル開発です。

アジャイル開発の大きな特徴として、1週間から2週間程度の開発期間を反復して繰り返すことが挙げられます。この期間の中で開発する機能の設計と実装、テストまでを行い、開発期間終了時にはソフトウェアとしてリリースします。これにより、ソフトウェアの機能を段階的にチェックすることが可能です。このような形でプロジェクトを進めることにより、アジャイル開発は途中で要件が変わってもウォーターフォールよりも対応しやすく、手戻り工数も抑えられるという利点があります。

### ビジネススピードを加速させるアジャイル開発

**DevOpsによる迅速なソフトウェア開発へ**  
Gartnerの予測によると、2020年までに企業のコアアプリの75%は、購入ではなく個別開発されるようになる。よって、各企業がソフトウェア開発の専門家になる必要があり、企業はソフトウェア開発志向になっていく。(ソフトウェアが企業を定義する)



このアジャイル開発とともにソフトウェアの開発現場で広まり始めているのが「[マイクロサービスアーキテクチャ](#)」です。これは一枚岩のソフトウェアの中に複数の機能を詰め込むのではなく、個々の機能を個別のソフトウェア（サービス）として開発し、それらを[API](#)

などで連携させることによってシステムを構成するという考え方になります。機能単位で開発を行うためアジャイル開発と相性がよく、また機能単位での修正やアップデートが行えることから、要件の変更にも柔軟に対応することが可能です。

### マイクロサービスとは

サービスを小さな単位に分割し、  
各サービス間をAPI等で連携させるアーキテクチャ



**メリット** : 各機能を独立して開発リリース可能  
→ システムの変更、拡張、発展が柔軟に

**デメリット** : 分割の粒度が小さくなりすぎると、  
サービス間の関連が複雑になり管理が煩  
→ 適切なポリシーが必要

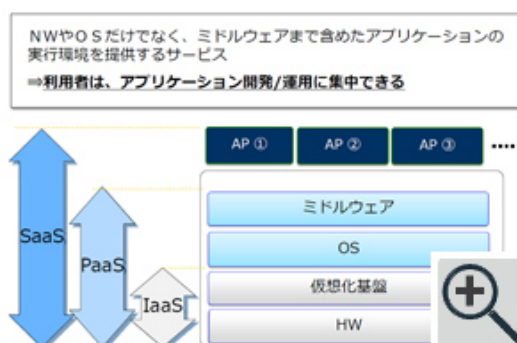


## アジャイル開発に最適なプラットフォームとなるPaaS

このアジャイル開発やマイクロサービスアーキテクチャの考え方を採り入れる上で、意識しておきたいのは最適なインフラの構築です。ソフトウェアのレイヤーがどれだけ柔軟になっても、その下のサーバーやネットワークといったインフラが硬直的では、せっかくのアジャイル開発やマイクロサービスアーキテクチャのメリットを生かせないためです。

柔軟なインフラを実現するための具体的なサービスとしては、[PaaS](#)や[コンテナ](#)が挙げられます。PaaSはソフトウェアの実行環境を提供するクラウドサービスであり、開発者はサーバーやOSの存在を意識する必要はありません。提供される環境の中で開発しなければならないという制約はありますが、その一方でOSやミドルウェアのインストールや設定といった手間が省けることは魅力でしょう。

PaaS(Platform-as-a-Service)とは



PaaSによっては、[ブルーグリーンデプロイメント](#)

を容易に実現する機能が提供されていることもあります。これはブルー系とグリーン系と呼ばれる、2つの本番環境を交互に切り替えられるようにする仕組みです。

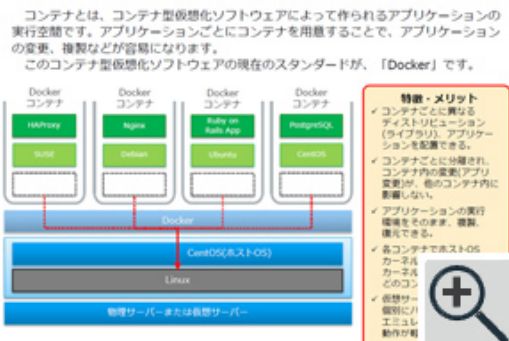
たとえばブルー系を使ってサービスを提供していて、ソフトウェアのバージョンアップを行いたいとします。通常であれば、一度サービスを止めてバージョンアップを行うことになります。しかしブルーグリーンデプロイメントの仕組みが使えると、待機系となっているグリーン系であらかじめ新しいバージョンを実行し、ブルー系から切り替えることで瞬時にソフトウェアを切り替えられます。これにより、システム停止時間を最小限に抑えられるほか、ブルー系とグリーン系がまったく同じ環境であれば、環境の差異による不具合も気にする必要がありません。

## すでに多くのクラウドベンダーがPaaSやコンテナ環境を提供

一方のコンテナは、1つのOS(ホストOS)上で、ソフトウェアを実行するための独立した空間を提供する仕組みです。イメージはサーバー仮想化技術と似ていますが、コンテナでは仮想化したハードウェアを使うのではなく、開発したソフトウェアとそ

れに必要な部品(ランタイム)をほかのソフトウェアから隔離しつつ直接実行します。このため、仮想によるオーバーヘッドがないほか、サーバー仮想化よりも物理リソースの利用効率がよく、1つの物理サーバー上でより多くのコンテナを実行できるというメリットがあります。

## コンテナ/Dockerとは



環境依存を意識することがないこともコンテナの大きな利点です。このため開発環境のコンテナを使ってソフトウェアを開発した後、それを本番環境のコンテナに移すといった際に環境に合わせて設定などを変更するといった必要がなく、環境依存を気にせずに開発することができます。

PaaSとコンテナに共通する特長として、ソフトウェアを実行するための環境を手早く整えられるため、スクラップ＆ビルドが容易であることも挙げられます。これもアジャイル開発の利点を生かす上で重要なポイントであり、また開発スピードの向上にもつながるでしょう。

国内では、NTTコミュニケーションズが「[Enterprise Cloud](#)」において、オープンソースで開発されているPaaSプラットフォームである「[Cloud Foundry](#)」や、DockerコンテナやDockerホストを容易に管理する「[Rancher](#)」を利用したサービスを用意しています。グーグルの「Google Cloud Platform」やマイクロソフトの「Microsoft Azure」、アマゾンの「Amazon Web Services」など、各社のクラウドサービスではすでにPaaSやコンテナ環境を提供しており、手軽に試すことができます。またまずはこれらのサービスを試用し、PaaSやコンテナのメリットを確かめてみるとよいでしょう。

## クラウド活用において重要なポイントとなるネットワークインフラ

クラウドサービスを積極的に活用する、いわゆる“クラウドネイティブ”なソフトウェア開発では、PaaSやコンテナ以外のサービスも積極的に利用したいところでしょう。具体的には、ビッグデータ分析のための基盤、あるいは機械学習を可能にするサービスなどが考えられます。

このようにクラウドを活用していくと、言語処理はGoogle、機械学習はMicrosoft Azureなど、複数のクラウド基盤をまたいでシステムを構築するといったことも十分に考えられます。そこで重要となるのがネットワークです。

各クラウドを接続するネットワークを個別に構築すれば、インフラの運用負荷が増大する恐れがあるほか、トラフィック量の増減などといった変化への対応にも時間を要する可能性があります。インターネットを利用して接続するのも1つの手ですが、セキュリティ面での不安が残ります。

データセンターやクラウド上で運用している、既存システムとの接続が必要となるケースもあるでしょう。たとえばIoTデバイスから取得したデータとERPに蓄積されている情報を統合して分析するといったソフトウェアを開発するといった場合、クラウドサービスとして提供されているPaaSやコンテナ基盤と自社システムを接続する必要があり、ネットワークの要件はより複雑になります。

このような場面で重宝するのが、複数のクラウドサービスをまたいで閉域網での接続を実現するVPNサービスで、具体的にはNTTコミュニケーションズの「[SD-Exchange](#)」や「[Multi-Cloud Connect](#)」

」などが挙げられます。こうしたサービスであれば、クローズドVPNの1拠点として各クラウドやデータセンター上のコロケーションを接続できるため、ネットワークインフラをシンプルに保ったまま、複数のクラウドやオンプレミスのシステムを連携したシステムを構築することが可能です。

### NTTコミュニケーションズ「SD-Exchange」

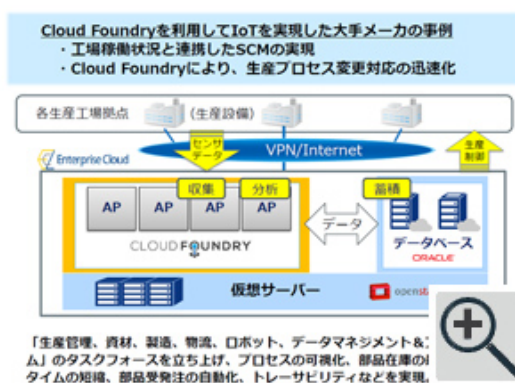
ハイブリッドクラウド向けの相互接続バックボーン「SD-Exchange」を活用することで、データセンター「Nexcenter」、AWSなどの各種クラウドにもセキュアに接続できます。



### アジャイル開発に取り組む前に

それではPaaSやコンテナを利用することでどのようなメリットがあるのか、具体的なユースケースで見てみましょう。まず[IaaS](#)で提供される仮想サーバーに加え、PaaSやコンテナ技術も活用してIoTに取り組み、[SCM](#)を最適化した例です。昨今では、SCMのような従来型のシステムであっても、あらゆる要求に対応できる柔軟性や、スピード感のあるアプリケーション開発が求められることが少なくありません。それを実現するために、PaaSやコンテナ技術を活用するというわけです。

### NTTコミュニケーションズ IoTによる精算プロセス変更事例



ただし、こうしたインフラを活用してアジャイル開発を行うためには、相応のスキルやノウハウが必要となります。特にアジャイル開発は従来のウォーターフォール開発とは異なる独自の手法を学び、それに慣れるプロセスが欠かせません。その際に有用なのが、アジャイル開発のプロセスを学ぶことができるトレーニングコースやワークショップへの参加です。

前述したとおり、アジャイル開発にはさまざまなメリットがありますが、それを最大化するためには特有のプロセスについて理解を深めることが欠かせません。それらをトレーニングやワークショップで学びつつ、さらにPaaSやコンテナ技術の実環境にも触れることができれば、より深くクラウドネイティブなソフトウェア開発を理解できるでしょう。

なおアジャイルやクラウドネイティブなソフトウェア開発を実践する際には、いきなり大規模なプロジェクトで試すのではなく、まずはスモールスタートで感触を確かめるべきです。そうしてノウハウを社内に蓄積し、徐々にプロジェクトの規模を大きくし

ていくという流れです。またアジャイル開発に知見を持ち、PaaSやコンテナ環境をクラウドサービスとして提供しているベンダーをパートナーに選べば、安心してプロジェクトを進められるのではないのでしょうか。

※掲載されている内容は公開日時点のものです。

※掲載されているサービスの名称、内容及び条件は、改善などのために予告なく変更することがあります。