



# CIO のための アプリケーション モダナイゼーション ガイド



2020年9月

## 知っておくべきこと

世界に広がるパンデミックにより、あらゆる業種の企業が、年初には誰も予想していなかったペースで変化を迫られています。それにより、CIO は次のような喫緊の課題を突き付けられています。

- 分散化が進み、かつてなくめまぐるしく変化するシステムを的確に可視化し、運用を管理するにはどうすればよいか。
- 費用を削減しつつ、よりアジャイルでレスポンスな IT システムを作り上げるにはどうすればよいか。
- 既存のデータを有効活用しつつ、新しいデータを迅速に手に入れて、変化を続ける市場の状況を的確に把握するにはどうすればよいか。
- 変更管理の必要性を最小限に抑えて、従業員の作業効率を向上させるにはどうすればよいか。また、大きなリスクを負うことなく、成長できる環境を整えるにはどうすればよいか。
- ユーザーのニーズにより良く応えるためにアプリケーションのスマート化を進めるにはどうすればよいか。

多くの場合、クラウド コンピューティング テクノロジーを段階的にデプロイし、最優先のニーズから優先的に対応することでこれらの課題に対応できます。

包括的なサービスを提供するクラウド プロバイダを利用することで、企業が持つコアアセットに配慮し、引き続き活用しながら、迅速なモダナイゼーションを実現するクラウド システムを構築できます。特に、アジャイルで耐障害性に優れたモダン アプリケーション アーキテクチャを費用効率良く構築したい場合に最適です。

## このガイドをおすすめする理由

このガイドでは、ニーズの特定、アクションに基づくロードマップの策定、IT 運用の最も重要な部分でどのような有意義な変革が可能かを見極めて実現する方法など、アプリケーションとデータのモダナイゼーションについてさまざまな角度から説明しています。

また、このガイドの最後では、企業が迅速に結果を出すのに役立つ、Google Cloud とテクノロジー パートナーが提供する主なソリューションの一覧も紹介しています。

## はじめに

現在の危機的な状況が生じる前から、IT 組織はアジリティを高め、イノベーションを推進しなければならないというプレッシャーにさらされてきました。顧客のユーザー属性や期待は常に変化しています。思わぬ場所から次々と競争相手が現れます。ビジネスモデルを見直す必要もあります。このような課題に対応するにあたり、中心となったのがデジタルテクノロジーです。あらゆる企業にとって、デジタルテクノロジーの導入が鍵となっていました。

また、企業には変化を妨げる要因が以前から存在し、課題となっています。特にレガシー IT システムは、多くの予算が注がれる一方、変革を妨げる要因となっています。多くの企業のオンプレミス インフラストラクチャでは、複雑で老朽化したレガシー アプリケーションが稼働しており、いまだに中心的な役割を果たしています。機能を追加するのが大変で、メンテナンス、セキュリティ、ライセンス、管理などにかかる費用は膨大です。アプリケーションのクラウドへの移行は一部で進んでいますが、[McKinsey の調査](#)によると、CIO の 80% が、アプリケーションのモダナイゼーションに期待していたアジリティやビジネスの成果を実現できていないと回答しています。企業が必要としているのは、差別化を図ることができる、新しいユーザー エクスペリエンスをアプリケーション チームが迅速かつ効率的に提供できるような、柔軟性が高くスケーラブルなソフトウェア サービスの基盤です。

現在生じている危機的な状況により、このような従来から存在した課題への対応が待たなしとなっています。リモートワークに対応した労働力の再編、e コマース、オンライン ロジスティクスなどの分野では、デジタルテクノロジーの導入が急速に進んでいます。リーダーは、これらの分野で実現した早いペースでの変革を組織の他の部分にも広めようとしています。

そのために策定された計画の多くでは、パブリック クラウド コンピューティングテクノロジーの短期間での導入を目指しています。運用費用を支払うだけで、パブリック クラウド テクノロジーのプロバイダが保有するデータセンター資産を利用でき、利用状況も一元的に把握できるため、インフラストラクチャの費用を削減し、管理のオーバーヘッドを低減できます。利用状況に応じたスケールアップやスケールダウンも簡単です。インフラストラクチャの管理に専念するクラウド プロバイダが新しいイノベーションを提供してくれるので、最新のテクノロジーやセキュリティのアップグレードをリアルタイムで導入できます。低コストで、柔軟性が高く、俊敏性も高まります。

変化する市場のニーズに企業が迅速に対応できるよう、IT 組織はアプリケーションから生成されたデータとイベントを収集、処理、分析し、分析した情報を共有できる仕組みを用意する必要があります。IT 組織は、ログファイル、トランザクションストア、クリック インタラクションなどから発生するこれらのデータを活用し、アプリケーション環境全体で一元的な意思決定と分析処理を行える機能を必要としています。

クラウドを導入すると、フルマネージド型のクラウド ネイティブ データベース サービスなどのテクノロジーや、「サーバーレス」デプロイなどの手法とともに、このような優れた仕組みや機能が自動的に提供されるので、運用費用を削減しつつ、イノベーションの推進に重点的に投資できます。そのため、企業の最も貴重なリソースである人材と時間の有効活用につながります。

世界が大きく変わる一方で、組織を変えることはこれまでと同様困難が伴います。さまざまなビジネス関係者が、それぞれに競合するニーズを抱えています。ビジネス クリティカルな運用を中断するわけにはいきません。変革を成し遂げるための万能の解決策はありません。大きな変革を一夜にして実現することもできません。

アプリケーションのモダナイゼーションを進める取り組みにおいては、データ アーキテクチャが重要であると認識されています。クラウドのデータ アーキテクチャは、ハイブリッドで、マルチクラウドをサポートし、データのサイロの統一やリアルタイムでのデータの利用を実現し、移行を推進して、デベロッパーを支援します。さらに、セキュリティ、コンプライアンス、パフォーマンス、稼働率の改善に役立ちます。



進むべき道筋ははっきりしています。クラウド プロバイダと連携して、以下を行う必要があります。

- ✓ アプリケーションのポートフォリオを明らかにし、ビジネス上の優先度を割り当てて、リスクを評価する
- ✓ ビジネスの成果を中心に据えたモダナイゼーションのプロセスを策定する
- ✓ 既存のサービスを再利用して新たな収益を生み出す方法を考案する
- ✓ モダナイゼーションの実施中と実施後にスタッフをサポートする
- ✓ クラウド アーキテクチャとモダナイゼーションのロードマップを策定する
- ✓ 長期にわたり効率的な最適化を行う



## 取り組みを始める

アプリケーションのモダナイゼーションでは、ほとんどの場合まずビジネスの目標と要件を定義してから、アプリケーションとデータの検出へと進みます。既存のアプリケーションポートフォリオを合理化することで、効率性の向上、複雑性の低減、[TCO](#)の削減につながります。

アプリケーション環境を把握し、アプリケーションをクラウドに移行した場合のリスクと、クラウドに移行する重要性を評価します。企業のアプリケーションポートフォリオには、4種類のアプリケーションが存在します。

**顧客向けアプリケーション:** 最も目につきやすいアプリケーションです。このアプリケーションに不具合があると、収益を失うだけでなく、ブランドの信用も傷つける恐れがあります。

**ビジネス アプリケーション:** 企業の業務運営を支援する、バックエンドのエンタープライズ アプリケーションです。ERP、SCM、財務アプリケーションなどがあります。

**社内アプリケーション:** 社内業務の基盤となる、メール、人事、給与計算などのアプリケーションです。

そして最後に、**モニタリング、レポート、分析を行うアプリケーション**があります。

これらすべてのアプリケーションは、常に大量のデータを利用しながら動いています。アプリケーションが利用するデータは急激に増加しており、データ形式やデータのソースも増え続けています。既存のデータアーキテクチャでは、このような大量のワークロードや、変化し続ける要件に対応できない可能性があります。インフラストラクチャとアプリケーションをクラウドに移行する場合、現代の企業が抱えるデータの課題に対応できる、クラウドネイティブなデータベースサービスを導入してデータアーキテクチャをモダナイズすることを検討します。最新のデータアーキテクチャでは、生成および処理されるデータにインテリジェンス（機械学習のアルゴリズムなど）が継続的に適用されるとともに、そのスケールが動的に調整されます。また、強固なセキュリティが確保され、常にデータが利用できる状態に保たれます。

最初に、アプリケーションポートフォリオを把握し、コアアプリケーションとコアアプリケーションの基盤となるデータ、そしてそのオーナーを特定します。その後、これらのアプリケーションの使用状況とライフサイクルを明確に理解します。

次のステップはアプリケーションの合理化です。合理化を行う方法には、[Gartner の TIME モデル](#)や、[Forrester Wave の方法論](#)など数多くの方法が存在します。しかし、これらの方法は、今では古くなった IT テクノロジーが最新の手法としてもはやされていたころに開発されたもので、IT 中心の視点で変化をとらえているため、より大きな組織的な目標に合わせて優先順位付けを行うための明確な戦略を欠いています。そのため、これらの方法で合理化を行った場合、時間と手間がかかり、他の関係者と利害が対立することにもなりかねません。

最新のデジタルテクノロジーを使用することで、すべてを一度に変更することなくさまざまな側面を調整できる、一連の柔軟なサービスで構成された IT 環境を構築できます。それら一連の機能を組み合わせてアプリケーションを編成し、企業が抱える重要なニーズにも対応できます。エンドツーエンドの自動化や AI に基づく意思決定などの次世代の機能を構築するには、高度なセキュリティを備え、いつでもどこでもデータを利用できる柔軟なアーキテクチャを実装する必要があります。





## ビジネス サービス オリエンテーション

ビジネス サービス オリエンテーションでは、企業の IT ニーズを分析するとともに、障害となる要素を特定して解決します。それにより、一段上の効率性を目指します。CIO は、ビジネス上の価値を重視するアプローチを通して、モダナイゼーションのロードマップを明確に描きます。ビジネス全体で共通して抱える制限事項を把握し、それを解決することで ROI や SLA の向上、そしてイノベーションの推進につなげます。

最初に、外部の顧客向け機能から、セキュリティやデータ分析といった社内ツールまで、あらゆるビジネス機能について検討します。たとえば注文管理は、さまざまなビジネス サービスによって行われる、注文の作成、注文の照会、注文の更新などのビジネス上の機能です。多くの場合、これらの機能は、別のクラウド サービスを含むさまざまな環境のさまざまなプラットフォームで実行されています。IT チームが組織的に導入したプラットフォーム、各部門で導入したプラットフォーム、企業買収により継承したプラットフォームなどさまざまです。

すべての機能とサービスを特定して、詳細を把握できることが理想です。そうすることで、それらを戦略化し、次のアクションにつなげることができます。この作業により、機能とサービスをそれぞれ分離されたアプリケーションとして適切に管理できます。デジタル化に関するいくつかのテーマを重点的に進め、エンドツーエンドのカスタマー ジャーニーを再定義しようと考えている企業もあるでしょう。その場合、新たなビジネスモデルの定義や、機能横断型チームとデータのコラボレーションが必要になることもあります。

## アプリケーションの 把握と分離

多くのビジネス サービスは、巨大で複雑なレガシー アプリケーションに存在しています。このようなアプリケーションは、「モノリス」と呼ばれることがあります。アプリケーションの分離とは、このモノリスの一部分を取り出して、独立したサービスとして実行できるようにするプロセスです。このプロセスは、ビジネスニーズの観点から捉えることができます。そのようなモノリスを分割するためのリファクタリングや再設計の方法として、機能分割があります。API ベースのアプローチによりアプリケーションをビジネス サービスに分離することで、サービス オーナー、サービス利用者、その他のビジネスチームなど、組織の各部分が、それぞれのペースで業務を進めることができるようになります。また、組織全体でこれらのサービスがどのように使用されているかも明確化できます。どのような方法でアプリケーションを分離するかにかかわらず、アプリケーションのモダナイゼーションにおいて、データの依存関係を特定し、データの整合性の問題を解決することが、モダナイゼーション プロジェクトを成功させる鍵となります。これは、[クラウドネイティブに向けた再設計](#)という大きなプロセスの一部に含まれます。また、マイクロサービス アーキテクチャを使用して、ロギング、モニタリング、レポートなどの共通サービスをサポートすることも検討します。

機能分割が完了すると、各構成要素とその依存関係を明確に把握できます。これらの分割した各ワークロードは、独立してスケーリングおよびパーティショニングできます。高い可用性と耐障害性が求められるサービスでは、分散システムに単一のサービスのレプリカを複数用意する必要があります。バックエンドも、連鎖的に生じる障害への対策として、分散化が必要になります。厳しい規制要件が課せられるサービスでは、単一の地域内にデータを保持する必要があるかもしれません。逆に、世界中にデータを分散することでレイテンシを短くすることが望ましいサービスもあります。頻繁にスケールアップが必要なサービスの依存関係を分離することで、関連性のない依存関係やサービスまでスケールアップする必要がなくなり、費用とパフォーマンスをより適切に管理できます。これらの分離されたアプリケーション サービスが必要とする関連データは、世界中に分散したフルマネージド データサービスによって提供されます。

ビジネスの基礎となる構成要素を把握することで、適切にサービスを分離できます。たとえば、単一のモノリスとして存在する e コマース システムには、顧客認証、アカウント番号検索、顧客リワード プログラム、ロイヤルティ ポイント付与、商品のプロモーション システム、顧客の購入パターン認識機能などのサービスが存在します。これらを相互に連携する別々のサービスとして分離するとどうなるかを検討することで、さまざまな依存関係や障害発生ドメインを明らかにできます。これにより、サービスや依存関係をどのように論理的に分割するか、あるいはどのサービスを分割しないかを適切に判断できます。

# モダナイゼーションのプロセス

## プラットフォームの設計

現状のアプリケーションとデータサービスの全容を把握したら、社内の他のチームや関係者向けにソフトウェアを提供する環境について計画します。先ほど行ったアプリケーションの分離によって、社内向けの IT の範囲を明確にする作業は大部分が完了しています。フェイルオーバー、高可用性、ビジネス継続性、自動化、セキュリティ、スケーラビリティなどを通して、継続的に優れた成果を上げることに重点を置きます。

既存のアプリケーションがコンテナ化されていない場合や、手動のプロセスが残っていて自動化が進んでいない場合など、最初のうちは既存のアプリケーションやデータサービスがパフォーマンスの足かせになることがあります。新たなテクノロジーやプロセスを導入して、これらの機能を強化します。足かせになっている機能をすべて置き換えることも、古いシステムと並行して新しい機能を実行することもできます。

改善の状況を測定できるように、ベースライン指標を用意することが重要です。ソフトウェアのモダナイゼーションと提供に関して、3つのカテゴリから成る**汎用的な指標**があります。



### スピード

コードを本番環境にデプロイするまでにどの程度の時間がかかりますか。

どの程度の頻度で変更を展開しますか。

それらの変更を展開する際にどの程度のダウンタイムが発生しますか。



### 安定性

反映した変更のうち、障害につながる頻度はどの程度ですか。

どの程度の時間で復旧できますか。



### 信頼性

顧客はサービスを利用できますか。

予定外の停止からどの程度学び、改善につなげることができますか。

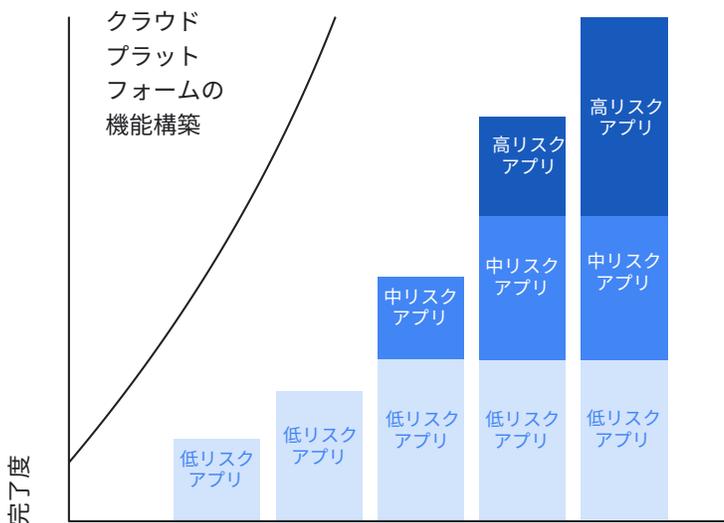
これらのベースライン指標を基準に、プラットフォームに導入した新機能の効果や、ソフトウェアテクノロジー、プロセス、人にとっての改善の度合いを判断できます。



## プラットフォームの構築

現状から目標に進む際は、実現できる価値の大きさ順に優先順位を付けます。新たな機能を導入すると、今までにないメリットを得ることができ、アプリケーションやサービスのモダナイゼーションにつながります。しかし、あまり利用されないバッチサービスなどの一部のアプリケーションやサービスには、それほどリソースを用意する必要はありません。費用やメリットを考慮してモダナイゼーションのニーズに優先順位を付けることで、実現すべきタスクを具体的に洗い出すとともに、明確な成功基準によってその効果を示すことができます。

移行についても、リスク、使用状況、必要性の緊急度などの要素に基づき進めることができます。障害、ロールバック、再試行による影響が少なく、運用上のリスクが最も少ないアプリケーションから始め、学びながら移行を進めることができます。その後、利用率やビジネス上の重要性などに基づき最も影響の大きいサービスまで段階的に移行を進めます。考慮すべきリスクには、顧客への直接の影響、収益の損失、ブランドへの悪影響などがあります。

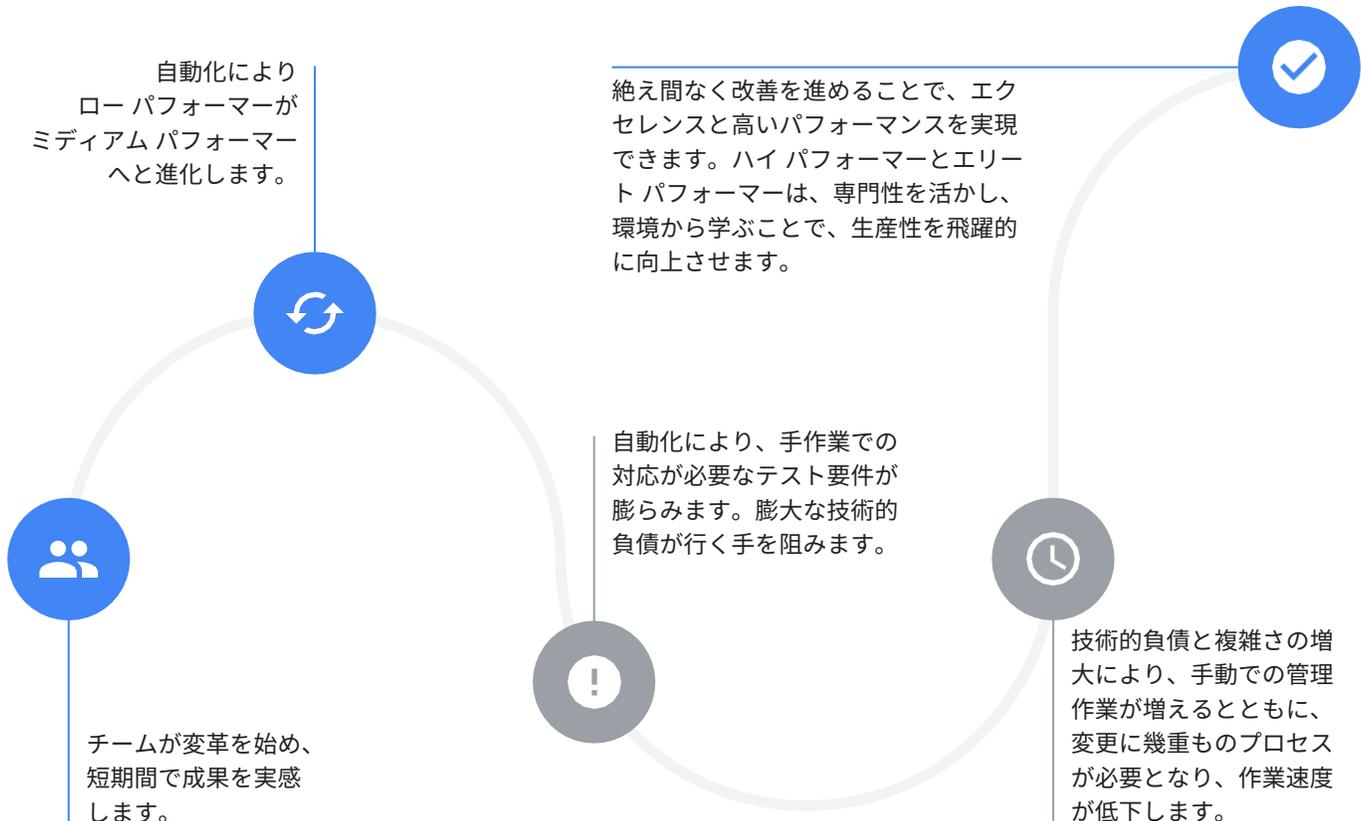


アプリケーションの段階的な移行



最新のアーキテクチャは柔軟なモジュール式の構造になっているので、新しい機能を自分のペースで簡単に導入できます。多くの作業を並行して行うことも可能です。そのため、リスクを抑えながら新しい機能について学習を進め、徐々にリスクの高いサービスの移行へと移ることができます。モダナイゼーションを進める IT エンジニアのチームや、プロダクト担当チームと連携してモダナイゼーションを進めるチームでは、[ペース レイヤリング](#)や [Proxy パターン](#)などのサービス適応の方法論を学習する必要があります。

従来の IT 変革と比べた際のもう一つのメリットとして、長期的な効果を視野においてプラットフォームへの投資を進めるうえで、短期的な効果や中期的な効果も享受できることがあります。Google の DevOps Research and Assessment<sup>9</sup> (DORA) では、これを「変革の J カーブ」と呼んでいます。



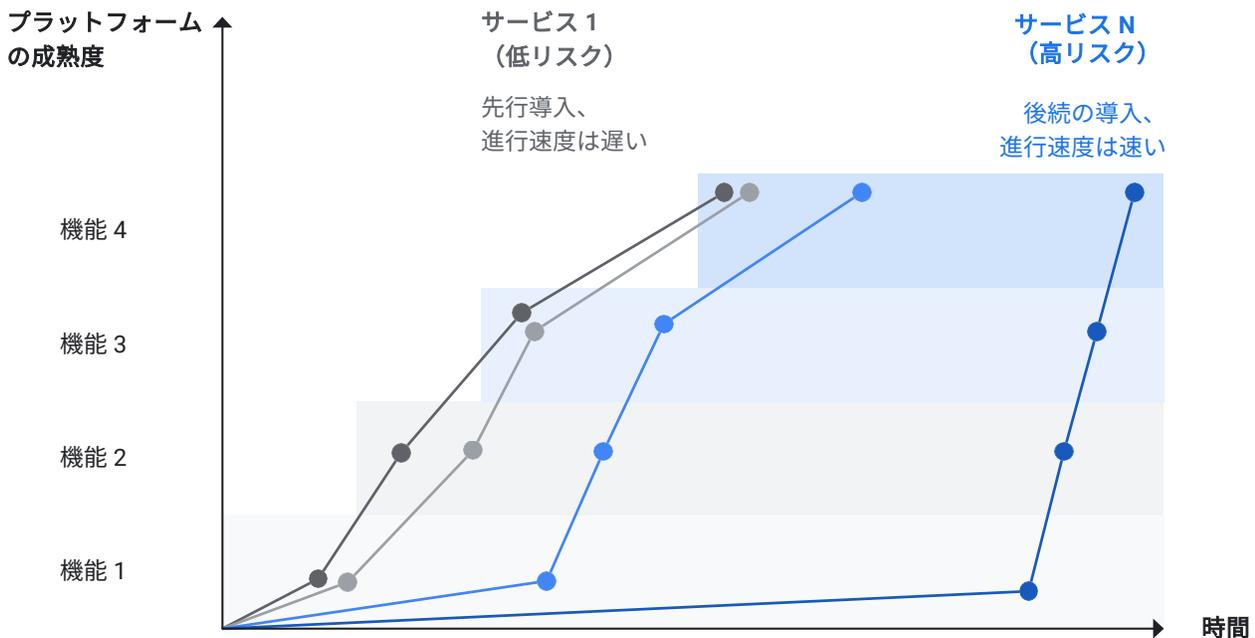
また、DORA では、よりパフォーマンスの高いソフトウェア デリバリーと運用を実現するいくつかの技術、プロセス、文化に関する機能<sup>10</sup>を明らかにしています。それらを以下にまとめます。

計画	開発とテスト	リリースとデプロイ	監視と最適化
小さいバッチサイズ	アジャイル開発	標準化されたプラットフォームとプロセス	ほぼリアルタイムのフィードバックループ
専任の機能横断型チーム	継続的インテグレーション	自動化された環境のプロビジョニング (IaC)	アプリケーションを結びつける疎結合アーキテクチャ (API)
継続的計画	テストの自動化	自動リリースとデプロイ (ツールチェーン)	指標と信頼性

各チームは、これらの機能を活かし、チーム内でより効果的なソフトウェア デリバリーを実現できます。このソフトウェア デリバリーには、プラットフォームの開発も含まれます。より効果的なプロセスとテクノロジーを使用できるように、今後導入するプラットフォーム機能を定義して実装できます。

## プラットフォームの使用

ビジネス機能に低リスクのサービスを導入すると、一度に複数の機能が実現されます。



例として、現在以下のようなプラットフォームを持つ架空の企業を考えます。

- VMware を使用して VM と SAN を単一のオンプレミス コロケーション施設に保有している
- Windows サーバーで単一のモノリス ウェブ アプリケーションが実行されており、週 1 回のメンテナンスの時間枠にデプロイされる
- 大規模な MS SQL Server と Oracle データベースがウェブサービスと API を通して公開されている

この企業では、まず理想的な移行先の環境を特定します。ここでは、以下のような環境を移行先として選びます。

- Anthos Service Mesh を使用して、複数のグローバル リージョンの GKE でコンテナを使用
- マネージド データベース: Cloud Datastore
- マネージド リレーショナル データベース: MS SQL Server の代わりに Cloud SQL、Oracle データベースの代わりに Bare Metal Solution

移行を行うために、以下の機能を導入しますが、まずは小さいインスタンスや単一のサービスからデプロイし、運用に慣れるに従って数を増やすというパターンに従います。

特定された制限	導入する機能またはテクノロジー	得られるメリット	最初のステップ
開発環境と本番環境があまりに異なっており、バグにつながっている	サービスを「コンテナ化」する機能	開発環境、ステージング環境、本番環境の予測可能性の向上	1つのサービスをコンテナ化してデプロイし、ツールや手順に慣れ親しむ  長期的にすべてのサービスをコンテナ化する計画を策定
データベースが複雑で管理の手間がかかる	Google Cloud のフルマネージド データベース サービス	管理は Google に任せてアプリケーション開発に集中	リスクの低いワークロードを特定し、適切なデータベース サービスに移行
手動でリリースしているためエラーが発生しやすく、時間もかかる	コンテナのシンプルな CI / CD パイプライン	自動化されたテストとデプロイ	まず基本的な CI、次に CD  時間とともに徐々に対象とするサービスを増やす  時間とともに CI と CD にステップを追加
現在 VM で多くのコンテナを手動で実行しており、リスクが高い	GCP で単一の GKE クラスタを導入	サービスのオーケストレーションと簡単な自動スケーリング	1つのサービスを GKE に移行  対象とするサービスを徐々に増やす
新たな分散システムの監視や一貫性のあるポリシーの適用に多くの手作業が必要となる	Anthos Service Mesh と Anthos Config Management を追加	動的なサービス ディスカバリー、コードとしての構成、サービスレベルのテレメトリー	GKE に複数のサービスを移行したら、サービスメッシュを追加してトラフィックを監視し、ポリシーをデプロイ  デプロイするポリシーを徐々に増やす
一部のサービスの既存ユーザーにビジネス中断の影響を与えないことが必要	API 管理の導入 (Apigee)	コンテキスト アウェア ルーティング、ペイロード変換、ユーザー固有の API ファサード	サービスごとに、重要性の低いユーザーを最新バージョンのサービスにルーティングして、フィードバックを取得  その後、残りのユーザーをしっかりと管理しながらルーティング

特定された制限	導入する機能またはテクノロジー	得られるメリット	最初のステップ
<p>一部のサービスはレガシー データベースから 1 ミリ秒以内で到達できる必要がある</p> <p>一部のサービスはオンプレミスの安全なハードウェアを使用する必要がある</p>	GKE On-Prem クラスタを導入	レガシー サービスやデータと同じネットワーク内にデプロイできる機能	<p>GKE On-Prem クラスタを立ち上げ、そこに既存のオンプレミス サービスをデプロイ</p> <p>サービスやクラスタの数を徐々に増やす</p>
他のリージョンのユーザーのパフォーマンスが低い	GCP で 2 つ目の GKE クラスタを導入	リージョン レベルでの復元性と、グローバルなユーザーに対する低レイテンシの実現	複数クラスタの運用モデルを導入し、デプロイ、インシデント対応、負荷分散などを調整
大量の非リレーショナルデータが高価なリレーショナルデータベースに格納されている	まずは非リレーショナル データをマネージド データストアに移行	非常にシンプルなスケーリング	1 つのサービスのデータをデータストアに移行し、それに合わせて新しいコードをデプロイ
リレーショナル データベースが高価で、メンテナンスが難しく、スケラビリティに欠ける	残りのリレーショナル データをマネージド リレーショナル データベース サービスに移行	管理費用の大幅な削減	1 つのシャード、テナント、または顧客アカウントをクラウド リレーショナル データベースに移行し、その作業を繰り返す
市販の商用アプリケーションをいくつか公開しているが、メンテナンスや利用が複雑	API を通してベンダー非依存のエンドポイントを導入し、Velostrata、Migrate for Anthos などの方法でコードを GCP に移行	ベンダー固有のインターフェースをユーザーに公開する必要がなくなる	ベンダー非依存の API エンドポイントを作成してアプリケーションをクラウドに移行
ほとんどのサービスをクラウドに移行したが、オンプレミス環境を保持するのに費用がかかるほか、強固なセキュリティを実現できず、柔軟性も低い	リスクの高い、大量のデータを利用するアプリケーションを GCP に移行	オンプレミスのコロケーション施設を停止	ビジネス継続計画とデータの復元がコロケーション施設に依存していないことを確認し、コロケーション施設からすべてのトラフィックを除去する。ただし、確実な検証が終わるまでの一定期間は「コールド状態」（稼働させない状態）で環境を保持
主要なアプリケーションデータとその他のデータストアの統一的な可視性が必要	一元化されたデータ ウェアハウスとデータ統合を導入	ビジネス上の分析情報への迅速なアクセス、IT チームが短時間で簡単にレポートを作成	既存のデータストアから一元化されたデータウェアハウスへの変更データ キャプチャ (CDC) プロセスを設定



こちらの企業では、このモデルに従ってソフトウェア デリバリーおよび運用プラットフォームに段階的に機能を追加して、着実にビジネスに新たな機能を追加しています。各コンポーネントを追加することで、より多くのアプリケーションをモダナイズできるほか、すでにモダナイズされているアプリケーションに新たな価値を提供できます。

あらゆる IT 変革の取り組みでは、人、プロセス、テクノロジーの3つの要素への対応が求められます。ビジネス サービス オリエンテーションのプロセスについては先ほど説明しました。しかし、人についても適切な構造を導入しなければ、変革のプロセスを成功に導くことはできません。次に、この側面について検討しましょう。

## プロダクト デリバリー モデル

ほとんどの IT 組織では、プロジェクト ベースのモデルを通してソフトウェアを提供しています。このモデルでは、時間やスケジュールなど、部門ごとの目標に重点が置かれ、ビジネスのニーズが考慮されないことがよくあります。プロダクト デリバリー モデルを導入することで、ビジネスの成功という視点に基づきソフトウェアを提供できます。

プロダクト デリバリー モデルでは、各チームがソフトウェアの定義、提供、サポートまで、ライフサイクル全体に責任を持ちます。メジャー リリースやマイナーリリース、新機能、サポートの終了のライフサイクルについても決定します。

プロダクト チームと連携することで、プロジェクトの目標だけにとらわれず、長期的戦略に基づく視点からソフトウェアのライフサイクルを管理できます。プロダクト チームは、プロダクトを誰が使用するかによって、社内や社外の幅広い関係者の利害を調整しながら業務を進めています。そのため、ビジネス成功に向けてライフサイクルを反復的に実施し、求められる新機能を特定して、ビジネス目標に沿ったプロダクトリリースを実現できます。

このモデルでは、機能別にチームが編成されます。大きく分けて、2 種類のチームがあります。



### ビジネス機能チーム

ビジネス関係者と緊密に連携し、ビジネス サービスの提供を念頭に業務を進めます。サービス デリバリーおよび機能のロードマップは、各事業部門の目標に沿ったものになります。これらのチームは、社外と社内の両方のユーザーのニーズに応えることに重点を置きます。前述の例では、注文管理、フルフィルメント、分析、モバイルアプリなどを担当するチームが含まれます。

これらいずれの種類のチームも、ビジネス上の成果を重視し、長期的な視野から計画、構築、メンテナンスのライフサイクル全体を管理するという意味で、プロダクト モードで業務を行います。上位のチームが必要なサービスを提供するという点で、下位のチームにとって上位のチームはユーザーにあたります。このようなチーム編成により、分離された運用モデルを使用しながら、共通の目標に向かって業務を進めることができます。プロダクト デリバリー モデルとプロジェクト デリバリー モデルについて詳しくは、[こちら](#)を参照してください。



### プラットフォーム機能チーム

さまざまな事業分野で共通して利用できるプラットフォームを提供します。たとえば、データ分析チームは、データ ウェアハウス、データ処理パイプラインなどの管理を担当します。階層の上位にあたるチームの問題のあるビルドに対応する責任は負いません。上位のチームは、1つ以上の下位のチームが提供するプラットフォームを使用します。



## 成功を積み重ねる

ここまでのセクションでは、ビジネス価値の推進を念頭に、組織のプロセスと人の方向性を定める方法について説明しました。

ビジネスの機能と優先度に照らして組織のアプリケーション環境を把握することが、モダナイゼーションの取り組みを進めるうえでは欠かせません。また、この取り組みを進めるには、各チームがプロダクト デリバリー モデルに沿って業務を進める必要があります。

これら両方の側面を実現するためには、チームでの変更管理も必要になります。変更管理により、人やプロセスの変更を、継続的な改善と改良のプロセスへとつなげることができます。市場の状況、プロダクト戦略、競争環境などの要因により、時間とともに企業のビジネス機能の定義は変わります。それに従い、チーム構成も変更が必要です。しかし、運用モデルの一環としてこのような変革のアプローチを整えておくことで、ビジネスの価値の最大化に向けてリソースを有効に活用できます。

## Google Cloud でできること

自力でモダナイゼーションを実現することは非常に困難です。Google Cloud チーム、そして Google Cloud の[信頼できるパートナー](#)がお手伝いします。お客様との強固なパートナーシップを築き、モダナイゼーションのあらゆる段階のニーズに応えるソリューションを自信を持って提供いたします。

データベース ベンダーのロックインやレガシー Java 環境にまつわる課題の解決に取り組んでおられますか。あるいは、開発に最新の原則を導入したいとお考えでしょうか。Google Cloud は、人、プロセス、テクノロジーの3つすべての側面からお客様の変革の取り組みを支援します。

システムでモダナイゼーションが必要になっても、独力ですべての問題を解決する必要はありません。Google Cloud では、企業が直面するさまざまな課題に対応した多数のソリューションをご用意しています。以下に、その一部をご紹介します。

テクノロジーの目標	ソリューションの詳細
重要なビジネス アプリケーションを Google のグローバルで安全性の高いインフラストラクチャで実行する	<a href="#">Google Cloud での SAP</a>
Windows ワークロードを、安全性と費用効率の高いクラウド環境で実行する	<a href="#">Google Cloud での Windows</a>
VMware ワークロードを Google Cloud に移行し、ネイティブに実行する	<a href="#">サービスとしての VMware</a>
Google Cloud で特殊なワークロードを低レイテンシで実行できるハードウェアを使用する	<a href="#">Bare Metal Solution</a>
VM を迅速に Google Cloud に移行する	<a href="#">VM の移行</a>
レガシー アプリケーションを拡張、モダナイズし、新しいクラウド サービスと組み合わせる	<a href="#">レガシー アプリケーションのモダナイゼーションデータベースの移行とモダナイゼーション</a>
コンテナに移行して既存のアプリケーションを段階的にモダナイズし、運用のオーバーヘッドとライセンス費用を削減する	<a href="#">Anthos を使用して Java アプリケーションをモダナイズ</a>
Google Cloud の自動化されたツールと専門家のガイダンスにより、かつてないスピードでユーザーに革新的なソフトウェアを届ける	<a href="#">Anthos による最新の CI / CD</a>
デベロッパーとパートナーのエコシステムに新しいサービスやデータを提供し、ビジネス チャンネルを獲得する	<a href="#">API を使用した新しいビジネス チャンネル</a>
分析処理とデータアクセスを一元化する	<a href="#">データ ウェアハウスのモダナイゼーション</a>

また、Google Cloud には以下のような他社にはない独自の特長があります。



### 信頼性と可用性

Google Cloud のインフラストラクチャは、数十億人のユーザーが毎日複数の Google サービスを利用しても耐える設計になっています。お客様は、Google Cloud が提供するこの世界屈指のインフラストラクチャを利用して、同等レベルの信頼性と可用性を実現できます。



### 高い柔軟性

Google Cloud の Anthos プラットフォームでは、オンプレミス、Google Cloud、またはその他のクラウド プロバイダでアプリケーションを開発し、管理できます。お客様はこのメリットを活かして、オーバーヘッドを最小限に抑えつつ、ハイブリッドまたはマルチクラウドの環境でモダナイゼーション戦略を構築できます。オンプレミスとクラウド インフラストラクチャの両方で同じ Google Cloud のツールを使用できるので、移行中に運用のオーバーヘッドを最小限に抑えることができます。アプリケーションのモダナイゼーションプラットフォームとして Anthos を利用すると、[ROI を最大 4.8 倍に向上](#)できることが示されています。

独力で問題を解決しようと試みる前に、[cloud.google.com/solutions](https://cloud.google.com/solutions) で Google Cloud のソリューションをご覧くださいか、または Google Cloud 担当者までご相談ください。[CIO のためのアプリケーション移行ガイド](#) のホワイトペーパーもご覧ください。



### オープン プラットフォーム

IT 環境のモダナイゼーションを目指すほとんどの企業は、ロックインを避けて、オープンソースのプラットフォームを利用したいと考えています。そのため、Kubernetes がアプリケーションのモダナイゼーションで最も選ばれるプラットフォームとなりました。Google は、Kubernetes プロジェクトのトップコントリビューターです。Anthos を通して、Kubernetes を導入する企業にマネージドプラットフォームを提供しています。



### デフォルトでデータを保護

Google は、セキュリティ プロダクトの開発と関連する研究に年間 10 億ドル近くを投資しています。これらのツールやアプローチが、Google Cloud が提供するインフラストラクチャとアプリケーション スタックのあらゆるレイヤに組み込まれています。保存データと転送中データのデフォルトでの暗号化、クラウドベースの HSM、データの匿名化サービスなど、Google Cloud はモダナイゼーションプラットフォームを安全に運用するために必要なあらゆるセキュリティ ツールと機能を提供します。