

# プラットフォーム変革手引書案について

令和2年6月

国立研究開発法人新エネルギー・産業技術総合開発機構

独立行政法人情報処理推進機構

本書は、国立研究開発法人 新エネルギー・産業技術総合開発機構（NEDO）の「**Connected Industries**推進のための協調領域データ共有・AIシステム開発促進事業／データ利活用推進のためのシステム構築促進に向けた方策の検討」により作成しました。

## 目次

1. はじめに .....	1
1.1. 背景 .....	1
1.2. 本ドキュメントの目的 .....	2
1.3. 関連文書 .....	2
2. プラットフォーム変革手引書（案）の策定.....	3
2.1. 「はじめに」の記載概要.....	3
2.2. 「1章：プラットフォーム変革プログラムマネジメント手法」の記載概要.....	4
2.3. 「2章：現状システムの分析再整理手法」の記載概要.....	5
2.4. 「3章：新たな IT システムのあるべき姿」の記載概要.....	6
2.5. 「4章：新たな IT システムの設計開発手法」の記載概要 .....	6
3. DX を実現するための IT システムのあるべき姿の整理 .....	7
3.1. 必要な業務の適正な情報のみが必要なタイミングで取り出せる .....	8
3.2. 個々の変化に応じ独立に迅速かつ安全に IT システムを更新できる.....	9
3.3. 構造が柔軟で全体最適のために外部の有用なサービスを活用できる .....	10
3.4. 小さなサービスから始め、価値を確かめながら拡張していくことができる.....	11

# 1. はじめに

## 1.1. 背景

あらゆる産業において、新たなデジタル技術を使ってこれまでにないビジネスモデルを展開する新規参入者が登場し、ゲームチェンジが起ころうとしている。こうした中で、各企業は、競争力維持・強化のために、デジタルトランスフォーメーション（DX: Digital Transformation）をスピーディーに進めていくことが求められている。

このような中で、我が国企業においては、自ら DX を進めるべく、デジタル部門を設置する等の取り組みが見られる。しかしながら、PoC（Proof of Concept:概念実証。戦略仮説・コンセプトの検証工程）を繰り返す等、ある程度の投資は行われるものの実際のビジネス変革には繋がっていないというのが多くの企業の現状である。

今後 DX を本格的に展開していく上では、DX によりビジネスをどう変えるかといった経営戦略の方向性を定めていくという課題もあるが、これまでの既存システムが老朽化・複雑化・ブラックボックス化する中では、新しいデジタル技術を導入したとしても、データの利活用・連携が限定的であるため、その効果も限定的となってしまうといった問題が指摘されている。また、既存システムの維持、保守に資金や人材を割かれ、新たなデジタル技術を活用した IT 投資にリソースを振り向けることができないといった問題も指摘されている。

さらに、これを放置した場合、今後、ますます維持・保守コストが高騰する（技術的負債の増大）とともに、既存システムを維持・保守できる人材が枯渇し、セキュリティ上のリスクも高まることも懸念される。

もちろん、既に既存システムのブラックボックス状態を解消している企業や、そもそも大規模な IT システムを有していない企業、IT システムを導入していない分野でデジタル化を進めている企業等、上記のような問題を抱えていない企業も存在するが、全体を見た場合、これらの問題を抱えている企業は少なくないものと考えられる。

（出所 経済産業省 産業界におけるデジタルトランスフォーメーションの推進

[https://www.meti.go.jp/policy/it\\_policy/dx/dx.html](https://www.meti.go.jp/policy/it_policy/dx/dx.html)）

## 1.2. 本ドキュメントの目的

前節の背景のもと、独立行政法人情報処理推進機構（IPA）は、各企業が DX の実現に向けて IT システムを構築する際に参照するドキュメントとして、プラットフォーム変革手引書の案を策定した。今後、このプラットフォーム変革手引書案がより実用的なものになるように検討し、各企業が実際に DX の実現に向けて IT システムの構築に取り組むために、本書ではプラットフォーム変革手引書案の概要を紹介することを目的とする。

## 1.3. 関連文書

本書に関連する文書を表 1 に示す。

表 1 本書の関連文書

ドキュメント名	概要
DX レポート	DX を実現していくうえでの IT システムに関する現状の課題や、その対応策を中心に議論し、まとめたレポート。
DX 推進ガイドライン	DX を実現していくうえでのアプローチや、必要なアクションについての認識の共有が図られるように、経営のあり方、仕組み、IT システムの構築について取りまとめられたガイドライン。
DX 推進指標とそのガイダンス	各組織が、現状や課題の認識を共有する議論をして自己診断を行い、次のアクションにつなげる気付きを得るための指標及びその活用方法に関して取りまとめたガイダンス。
プラットフォームデジタル化指標について	IT システムに問題がありそうと経営者が判断 (DX 推進指標の IT システムに関する指標値が低い等) した企業が、既存 IT システムに対し、専門家の技術的支援を得て、技術的負債や DX 対応度合いを可視化することで、対策が必要なシステムを特定し、対策活動へつなげるための指標及びそのガイダンス。
プラットフォーム変革手引書案について (※本書)	DX 推進指標、プラットフォームデジタル化指標によって評価した結果、対策の必要があるとされたシステムを最適化する際の技術を整理して体系化し、個別の条件を加味して、実際の計画、設計を策定して、実行につなげるための手引書案。

## 2. プラットフォーム変革手引書（案）の策定

IT 担当者が IT システムのプラットフォーム変革・構築の際に参照する、ガイダンス「プラットフォーム変革手引書」の案を作成した。全体の目次については「表2 プラットフォーム変革手引書の目次」の通り。

表2 プラットフォーム変革手引書の目次

目次
はじめに
1章 プラットフォーム変革プログラムマネジメント手法
1.0 全体の関連、目的
1.1 現状システムの全体把握
1.2 あるべきシステム像の明確化
1.3 新しいプログラムマネジメント
2章 現状システムの分析整理手法
2.1 サブシステムタイプ別に分析再整理すべき設計情報 【前半】
2.1.1 Web型での分析再整理すべき設計情報
2.1.2 オンライン型での分析再整理すべき設計情報
2.1.3 バッチ型での分析再整理すべき設計情報
2.1.4 ゲートウェイ型での分析再整理すべき設計情報
2.1.5 概要設計での分析再整理方法
2.2 設計手法、テスト手法の整理 【後半】
3章 新たなITシステムのあるべき姿
3.1 あるべきシステムの機能要件の整理
3.2 機能要件を満たすためのあるべきアーキテクチャの整理
3.3 現行システムからの移行方法の整理
4章 新たなITシステムの設計開発手法
4.1 新たな開発手法に求められるシステム要件の整理（アプリ・ミドル・運用など）
4.2 新たな設計開発に求められる標準化

各章の記載概要については以下の通り。

### 2.1. 「はじめに」の記載概要

(1) プラットフォーム変革手引書の想定読者について

プラットフォーム変革手引書は企業などの経営に従事する立場の CIO(Chief Information

Officer)の方々、CTO(Chief Technology Officer)、CDO(Chief Digital Officer)、CDXO(Chief Digital Transformation Officer)および実際の現場の方々を対象としている。また指標による診断を進め、IT資産を分析したうえで、ITを活用してDXに見合う将来を目指すことができることを目的としているため、こうした推進を行う上で、ともにアセスメントを実施したりアドバイスを言ったりするコンサルタントやベンダーの皆さまにも参考としていただきたい。

## (2) プラットフォーム変革手引書の位置づけについて

DXを推進するためには、組織が保有するデータの活用が必要となってくる。しかし、既存のITシステムの構造やブラックボックス化・肥大化などにより、必要なデータを必要なタイミングで必要な精度で取得することが困難になることや、データを取得するためにITシステムの大幅な改造が必要になることがある。このような組織では、現状保有するITシステムを変革し、DXに対応できるようにしていく必要がある。

プラットフォーム変革手引書は、このようなITシステムの変革を行う際の、事前調査やあるべき姿の定め方、あるべき姿に向けた変革プロセスの進め方を手引きするものである。

## (3) DXにおけるITの重要性

スピード感のある経営革新のためにはITをうまく活用することが重要である。そのためには、たとえばPFデジタル化指標で明らかにした問題点を着実に対応していく、様々なシステムを精緻に見つめ、強化することなどが必要となる。

ところがこれまでのような手法や手段だけでは、十分な強化ができなかったり、求められているスピードに追いつけなかったり、多大な運用コストが発生するなどの課題を払拭しきれない。そこで本手引書ではDXを進める上で必要となる様々なIT戦略や実施手法について、その考え方を示していくことにする。

## 2.2 「1章：プラットフォーム変革プログラムマネジメント手法」の

### 記載概要

本章では大規模システムを多く保有している場合や、保有するIT資産が膨大な場合も対象とし、必要な情報を記載する。本書は、現状保有するシステムに対し、再構築を進めていくという場合にも活用できる手引書となると考える。

#### (1) DX関連取り組みとの関連

これまで、多くのシステム開発は実装したい業務ごとに企画・計画・設計・構築・運用されてきたと言っていいだろう。しかしこれから推進するDXにおいては、こうした「個別最適」からスタートすることは望ましくない。まずは企業や組織が所有するあらゆるシステムを把握することから始めるべきである。これら現状システムの全体把握を行った上で、プラットフォームデジタル化指標（以下「PFデジタル化指標」）を用いてシステム規模レベルⅢ程度（機能システム）にま

で分解し、その単位毎に評価を行う。

なお、機能システムとは、1つの企業の一事業部門が保有するシステムすべてを包含する規模レベル。目安として100万～200万ステップ。規模の目安が10～30万ステップのサブシステムを、数～10程度集めたもの。

#### (2) 現状システムの全体把握

本節では現状システムの全体把握について述べていく。現在複数の事業を持つような多くの企業や組織では、事業本部別など事業単位でのシステムの把握が行われていることが多い。これは事業本部などの単位で投資・回収がほぼ独立に行われているからである。しかしDXを推進するうえで、これら事業本部などに閉じず、組織を超えた最適化が求められる。

このため現状のシステム全体を把握した上で、それらの関係性の分解・分析を行う。この分解・分析はシステムレベルⅢ（機能システムレベル）まで行う。分解したシステム毎にPFデジタル化指標を用いて、DX対応状況の評価、さらに事業特性に応じた重要度を考慮して総合評価を明確にする。なお、分解を行ったすべてのシステムについて、それぞれ再構築を行うという意味ではない。

#### (3) あるべき姿のシステム像の明確化

各企業があるべき姿を描くために、現行システムの要件に加え、新たな要件や目指すべきアーキテクチャを加えて構成する手順を明確にしておく必要がある。ここでは、企業や組織全体としてのITシステムのあるべき姿を明確化する。

#### (4) 新しいプログラムマネジメント

ロードマップの策定、実行計画の立案と推進プロセスを定義、実行する上での設計開発手法の適用など、あるべきシステム像の実現に向けて、どのような手順でプロジェクト計画を立案していくかを明確にしておく必要がある。

## 2.3. 「2章：現状システムの分析再整理手法」の記載概要

本章では、全体システムを把握した結果、再構築が必要と判断されたシステムレベルⅢ（機能システムレベル）の機能システムに対して、現行システムから再構築の概要設計に必要な情報を分析再整理する方法を示す。

#### (1) サブシステムタイプ別に分析再整理すべき設計情報

一言でシステムといっても、様々な目的で様々なアーキテクチャに基づいてつくられたものが存在する。本手引書では、これらのサブシステムの多様性をカバーするために、4つの型に分類し、それぞれの型について、サブシステムごとの概要設計として必要な情報を整理する。

#### (2) 再構築における設計手法、テスト手法

前節までに分析した情報とユーザの求める新たな要件、目指すべきアーキテクチャから新たなシステムの設計を行う。その際、現行機能の保証範囲やその検証方法を明確にしておく必要がある。

## 2.4. 「3章：新たな IT システムのあるべき姿」の記載概要

新たな IT システムのあるべき姿としては、以下の要件が求められると考える。

- ・ 必要な業務の適正な情報のみが必要なタイミングで取り出せる
- ・ 個々の変化に応じ独立に迅速かつ安全に IT システムを更新できる
- ・ 構造が柔軟で全体最適のために外部の有用なサービスを活用できる
- ・ 小さなサービスから始め、価値を確かめながら拡張していくことができる

これらを満たすために必要となるアーキテクチャを整理する。また、現行システムからの移行について、新たな IT システムに採用するアーキテクチャと少なからず関連してくるため、併せて整理をする。

なお、IT システムのあるべき姿については、本資料の「3. DX を実現するための IT システムのあるべき姿の整理」に詳細を記載する。

## 2.5. 「4章：新たな IT システムの設計開発手法」の記載概要

新たな IT システム構築のため、設計開発手法としても新たなものが求められることになる。例えば、開発手法としては、アジャイル開発や API 等を活用した手法等。こうした開発手法の確立を行う。また、適用可能な実装技術、構築技術などのテクノロジー、手法についても、実践事例を挙げ、実際に活用する場面で、有効に使うことができるようにする。

### 3. DX を実現するための IT システムのあるべき姿の整理

DX 推進指標では、IT システムに求められる要素として「データ活用」「スピード・アジリティ」「全社最適」の3つを挙げている。詳細を整理すると「表3 DX 推進指標における IT システム要件」のようになる。

表3 DX 推進指標における IT システム要件

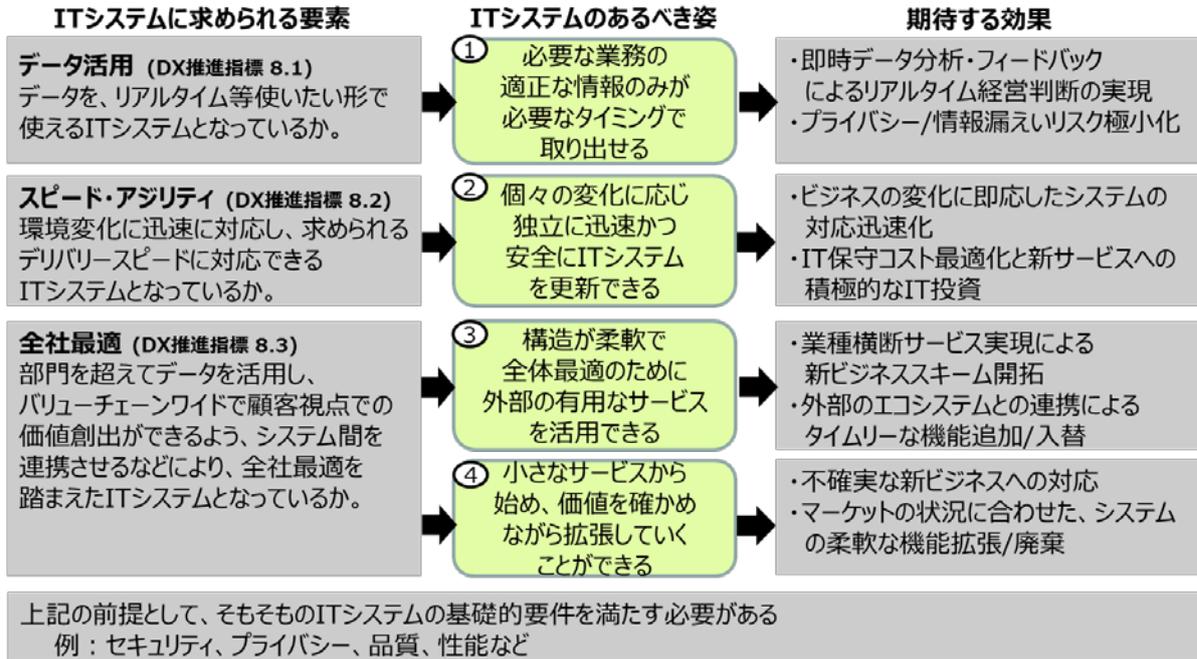
#### ITシステムに求められる要素 (DX推進指標 8)

<b>データ活用 (DX推進指標 8.1)</b> データを、リアルタイム等使いたい形で使えるITシステムとなっているか。	DXを推進する上では、そもそもどんなデータを持っているのか、その中でリアルタイムで使いたいデータは何かを認識し、実際にそれが使えているかが極めて重要であり、ITシステムはそれを実現できるものとなっていることが求められる。
<b>スピード・アジリティ (DX推進指標 8.2)</b> 環境変化に迅速に対応し、求められるデリバリースピードに対応できるITシステムとなっているか。	既存のITシステムの運用を前提としてしまうと、新規サービスのリリース頻度などを加速できなくなってしまい、結果として、加速する環境変化のスピードに取り残されてしまう。環境変化に迅速に対応できるようなデリバリーが可能になっているかがDX実現の上で重要である。
<b>全社最適 (DX推進指標 8.3)</b> 部門を超えてデータを活用し、バリューチェーンワイドで顧客視点での価値創出ができるよう、システム間を連携させるなどにより、全社最適を踏まえたITシステムとなっているか。	ディストラプターが顧客視点のビジネスモデルで攻めてきている中、それに対抗するためには、自社内外のバリューチェーンワイドでの組み替え等により価値を創出することが重要である。したがって、その基盤となるITシステムは、システム間連携などにより、部門を超えてのデータ活用が可能であり、また、データのオープンな流通を実現するエコシステムとの連携も容易に実現できるものとなっていることが必要となる。

次に、各要素を IT システムが実現すべき要件やそれによる期待や効果にブレイクダウンしたものを「図1 IT システムのあるべき姿と効果」に示す。

これらの要素は DX 固有の要件であり、これに IT システムとして基礎的な要素を追加したものが、DX を実現するための IT システムのあるべき姿となる。ここでは、DX 固有の要件について詳しく解説する。

図1 ITシステムのあるべき姿と効果



### 3.1. 必要な業務の適正な情報のみが必要なタイミングで取り出せる

これまで多くのシステムは、個別の業務や個別の用途を実現するために開発されてきた。これらのシステムで扱っている情報をもとでの用途以外で用いようとする、アクセスによる更新待ちが多く発生することや、目的の情報を含む大きなデータ群のまとまりでしかアクセスができないということが生じることがある。DXで求める「データ活用」はこうした制限によって阻害されてしまう。この問題を解決するには、ITシステムのアーキテクチャやデータ構造が、基準が明確化されているとともに、正規化されていて、必要な情報だけが（他の余計な情報含まずに）、必要なタイミングで（必要な頻度・鮮度で、必要なスループットで）取り出せるようにする必要がある。

期待効果：

- ・効果 1-1: 即時データ分析・フィードバックによるリアルタイム経営判断の実現

組織内の情報をリアルタイムで取得できることにより、常に最新の情報をもとに経営判断を行うことが可能になる。例えば、経営数値の集計に数か月かかっているとすると、経営者は数か月前の古い情報をもとに経営判断を行うことになる。これがリアルタイム情報をもとにできることによって、経営のスピードを飛躍的に向上させることが可能になる。

- ・効果 1-2: プライバシー/情報漏えいリスク極小化

システムから必要な情報だけを取り出すことによって、プライバシー/情報漏えいリスクを極小化できる。システムで扱う情報には機密性の高いものや個人や組織のプライバシーに関するもの

が含まれる。これらの情報を扱うシステムや担当者はできる限り極小化することが、リスク低減の基本である。例えば、市民が無料となるサービスを実現する際に、住人台帳システムに対して「この人は市民か？」という問いに Yes/No だけで答えることができれば、個人情報である「住所」などにはアクセスしなくてもサービスを構築することができる。こうすることにより、このサービスは個人情報を扱う必要がなくなる。

### 3.2. 個々の変化に応じ独立に迅速かつ安全に IT システムを更新できる

現行システムの多くは、モノリス構造などによれば、互いの機能が密に連携しひとつの IT システムを構成してきた。このような構造を持つシステムは一部の機能改変や追加であっても、影響を及ぼす範囲が大きくなってしまう。さらに、改変や追加だけではなく、そのテスト時には関係する機能全体のテストが必要となるため多くの時間と労力を要してしまう。少なくとも機能の改変や追加には多くの時間とコストを消費することになる。

それだけではない。テストが十分に実施できない場合には、当然品質の低下が想定できる。品質の低下はシステムトラブルとなって表れ、業務に影響を及ぼしてしまう可能性がある。DX で求める「スピード・アジリティ」はこうした制限によって阻害されてしまう。この問題を解決するには、IT システムがもつ様々な機能が細分化され、疎結合に設計・実装し、それぞれが独立にデプロイできるようにしておく必要がある。このような設計・実装にしておくことによって DX で求められる迅速なデリバリや安全な運用が可能となる。

期待効果：

- ・効果 2-1: ビジネスの変化に即応したシステムの対応迅速化

IT システムをビジネスの変化に追従できるように変化し続けさせることができる。前述したモノリス構造の IT システムでは、システムの改修に多くの時間と労力がかかる。そのため、通常、システムの更改は多くても年に数回までといったように制限され、仕様決定後、数か月以上かけてシステムを構築することが多い。このようなシステム修正のリードタイムネックにより、数か月前に決めた仕様ではリリース時にはすでにビジネス上古くなってしまっている可能性が生じてしまう。特にユーザや市場に近いサービスを提供している IT システムではニーズやマーケットの変化が激しいため、この問題が顕著に表れる。IT システムの機能を互いに疎な関係で設計・実装することができれば、入れ替えた機能だけをテストすればよいため、信頼性を担保しつつ迅速な改変を実現できる。これによりビジネス要求をタイムリーに IT システムに反映させることが可能になる。

- ・効果 2-2: IT 保守コスト最適化と新サービスへの積極的な IT 投資

守りの IT 投資である IT システムの保守コストを削減し、その原資を新たなサービスのための攻めの IT 投資に充てることができる。モノリス構造の IT システムは前述したように、信頼性や品質の担保に時間とコストがかかってしまう。十分にそのリソースを確保した場合でも、複雑なシステムは、すべての場合の検証を現実的に行えないことがあり、不具合の可能性を残したまま

リリースしてしまうことも少なくない。リリース後にその不具合が顕在化してしまった場合には、保守のリソースはその対策に追われることになり、さらにリソースが不足するという悪循環が発生してしまう。ITシステムを一つ一つは単純な機能ごとに疎な構造として実装し、それぞれを独立して保守や品質保証を行うことができれば、システム全体の品質リスクを低減させるとともに保守コストを抑えて、それを攻めのIT投資としてビジネス拡大や効率化に向けた新たなサービスに投資することが可能になる。

### 3.3. 構造が柔軟で全体最適のために外部の有用なサービスを活用できる

ITシステムを抱える組織はそれぞれ異なる業務を持っているため、ITシステムはそれに合わせてオーダーメイドやカスタマイズが行われてきた。このため、法改正や新たな技術革新に対応する際には、それぞれの組織が個別に対応しなければいけなくなってしまっている。例えば、会計基準に関する法改正が行われた場合、本質的には同じ仕様変更を、ベースのITシステムが異なるためにそれぞれ独立に改修する必要が生じてしまう。DXで求める「全体最適」はこうした制限によって阻害されてしまう。この問題を解決するには、業務を競争領域、非競争領域に分け、部品化しそれぞれの業務間を疎結合にしたうえで、非競争領域については業務やシステムの仕様を標準化し、すでに存在する他組織のサービスを活用できるようにするとともに、定期的にあるべき姿から逸脱していないかを検証し続ける必要がある。

期待効果：

- ・効果 3-1: 業種横断サービス実現による新ビジネススキーム開拓

非競争領域を既存のサービスを利用できることにより、新たなビジネススキームを柔軟に構築することができる。フルカスタマイズのシステムで業務を行っている場合、すべてのシステム更改を自ら行う必要がある。他の組織と連携をする場合、お互いのシステムを個別につなぐ必要がある。このためシステム更改がボトルネックになりビジネスのスピードを阻害してしまう。そこで、非競争領域の情報やインターフェイスを共通化し、共通化した仕様で連携を行うことができれば、ビジネス的に最適なタイミングで新たなビジネススキームを構築することが可能になる。

- ・効果 3-2 外部のエコシステムとの連携によるタイムリーな機能追加/入替

法令や社会制度など、ビジネスに大きな影響を与える変更が行われることがある。このような変更は、社会全体に広く影響を与える。したがって、個別の組織のシステムで対応すると効率が悪いことがある。このため、外部のエコシステムを活用し、変更を組織横断的に共通化することができれば、社会全体として、タイムリーに機能の追加・入替えをすることが可能である。

### 3.4. 小さなサービスから始め、価値を確かめながら拡張していくことができる

ビジネスの成功の確度が不明確なうちに多大な投資をすることは経営の視点では大変難しい。このため小さなサービスから始め、市場のニーズやサービスの価値を検証しながらビジネスをすすめることが多くなってきている。しかし、従来のウォーターフォールプロセスによる大規模システム開発で開発を行うと、システム全体の企画、要件定義、概要設計、機能仕様設計、実装、単体テスト、総合テスト、受け入れテストなど、運用の各プロセスが終わってから次のプロセスに進む。価値を評価するタイミングではすべての開発が終了しているため、評価結果のフィードバックが困難である。DX で求める「全体最適」はこうした制限によって阻害されてしまう。この問題を解決するには、ビジネスを支える IT システムを部品化して最初は小さく設計構築し、ニーズや提供価値の大きさに応じて次第に拡張できるようにする必要がある。

#### 期待効果

##### ・効果 4-1: 不確実な新ビジネスへの対応

常に新たなサービスや製品を提供しようとする場合、サービスは小さく始め顧客の反応を見ながら拡張するのが良い。IT が企業や組織内の手続きの自動化や効率化を目的として使われていた時代と異なり、現在では企業や組織はより多くの（顧客のための）サービスのために使われるようになってきた。ところが新たな事業やサービスの提供を行う局面では市場や顧客のニーズが明確にわからない場合がある。ここで、従来のようにいきなり大規模システムを開発してしまうと大きなリスクを抱えてしまうことになる。そこで、当初は IT システムへの投資は拡張性にも留意しながら小さく検証からスタートし、ビジネス価値の評価結果に応じて拡張していくことができれば、小さな投資を複数行い、効果を確認できたものから、ビジネスの拡大と同時に IT システムも拡大し成長させていくことが可能になる。

##### ・効果 4-2: マーケットの状況に合わせた、システムの柔軟な機能拡張/廃棄

DX の推進上では、マーケットの状況を把握しタイムリーにサービス改善に活用することが重要である。特に難しいのがユーザや市場に近いサービスである。なぜならユーザや市場に近い領域では「ニーズに変動」があり、その「変動の速度」が速く、また「想定・予測もしにくい」からである。つまりデリバリしたサービスや価値についての顧客の反応や、顧客や市場のニーズをいち早く把握し、こうした変化に対して継続してサービスを改良・改善、場合によっては廃棄しフィットさせ続ける必要がある。

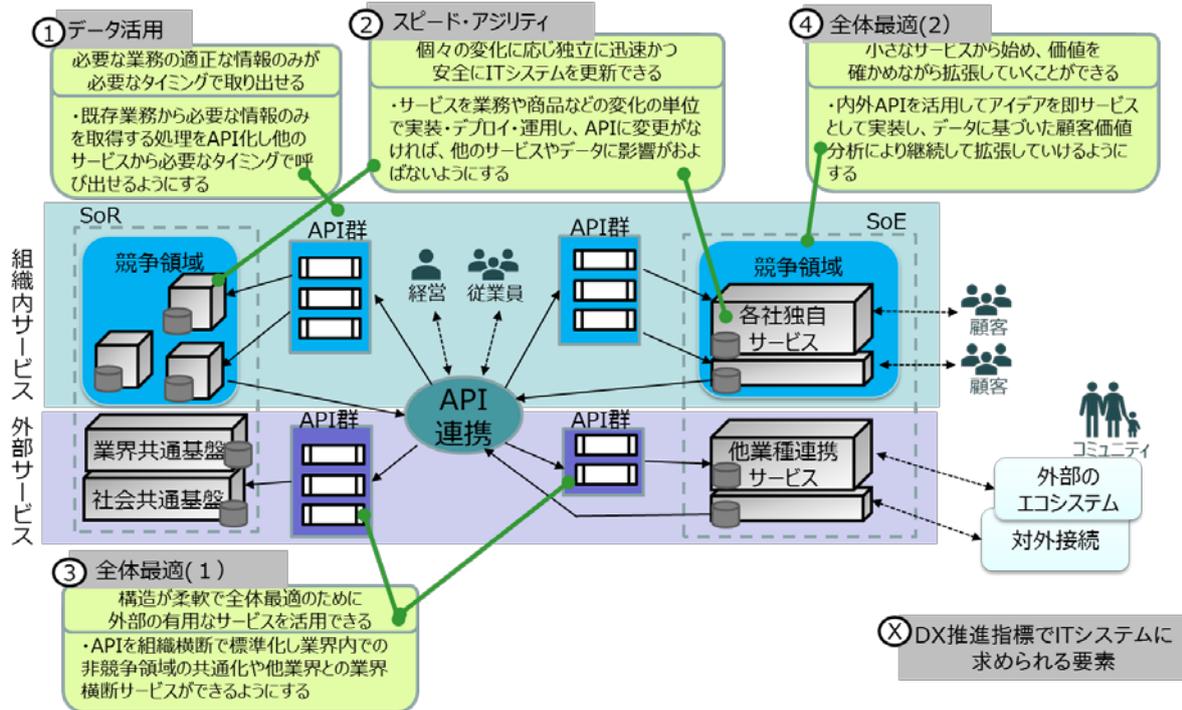
常に変化を察知し、敏感に反応し続けなければ提供サービスや価値とニーズの間にギャップが生まれる。このギャップを放置し続けるとユーザはより良いサービスやより提供価値の大きなサービスへ移行していき、顧客の喪失につながってしまう。サービスを実現している IT システムの持っている情報を活用し、顧客の経験を分析し、より成功に導けるようにサービスを改

善し続けることができれば、顧客の長期的なサービス利用につながり、安定したサービスのビジネス化が可能になる。

## 【あるべき姿の実装例】

これまで、DXに向けたITシステムのあるべき姿とその効果について述べてきた。以降で、そのあるべき姿の一つの実現形態を例として示していく。「図2 あるべき姿の実装例」がそのイメージである。

図2 あるべき姿の実装例



ここで示すものは、例であり必ずこのようにしなければならないものではないが、それぞれの要件が、具体的にどのように実現され、全体としてどのように機能するのかを理解するために活用していただきたい。

### (1) 必要な業務の適正な情報のみが必要なタイミングで取り出せる

この要件を満たすITシステムを、組織全体で活用するデータを特定し、それを取り出す手段を外部から呼び出すインターフェイス(API: Application Programming Interface)を持つプログラムとして実装することで実現できる。従来システムに多く見られるデータベースの共有によってデータ活用を行う形態では、データ取得の単位や範囲がデータベーススキーマに強く依存してしまうため、必要なデータを取得することが困難であったり、逆に不要なデータが含まれている状態で取得されてしまったりすることがある。データの基準を明確にし、構造を正規化したうえでAPI化することによって、データを必要とするシステムは実際のスキーマを考慮することなく、APIを呼び出すことで必要なタイミングで必要なデータを取得することが可能になる。また、ターゲットとするデータに特化して読み出せる工夫をすすめることで、情報に対するアクセス権限をこえる流通を防ぐことも可能となる。さらに、ITシステムによる情報の持ち方によってデータへのアクセスやレスポンスを高速化し、ITシステムに対する負荷を軽減することも期待できる。

#### (2) 個々の変化に応じ独立に迅速かつ安全に IT システムを更新できる

DX を推進するうえで、IT システムを市場や顧客の変化に迅速かつ柔軟に、また安全に更新し続けたいという要求はますます大きくなっている。こうした要件を満たすためには IT システムを、業務や商品などのビジネス上で変化する単位で独立して開発・配布・運用できるような構造にしておくことが重要である。たとえば MSA (Micro Service Architecture) などがそれを実現する技術思想の一つといえる。従来システムに多く見られる関係するサービス群を一つの大きなシステムで実現するアーキテクチャ (Monolithic Architecture) では、サービスの仕様変更の影響が全体システムに及んでしまうため、その影響調査や影響範囲に含まれる別のサービスの再テストに多くの労力がついやされてしまう。MSA を導入することによって、IT サービスを業務や商品などの変化の単位で変更しても、他のサービスへの影響を極小化することが可能になる。また、マイクロサービスの単位で開発・運用の複数のチームを構成することで、それぞれのチームで個別に判断して修正、リリースを行うことが可能になり、チーム間調整や同期のための時間の節約も期待できる。

#### (3) 構造が柔軟で全体最適のために外部の有用なサービスを活用できる

この要件を満たす IT システムを、業務や業種の共通プラットフォーム基盤を構築することで実現できる。個々の組織で個別にサービスを開発していると、同様ではあるが一部が異なるサービスが複数存在してしまう。これらのサービスを連携させる際や、技術革新による拡張を行う際には、やはり個別に同様な開発をそれぞれで行う必要が発生する。そのためには自組織が他社と差別化する業務 (競争領域) とそうではない部分 (非競争領域) をまず分別する必要がある。この分別をしておくことで、次に非競争領域について共同で開発もしくは、既存サービスの活用が可能となる。複数の組織で行ってきたきわめて類似した業務処理の開発をまとめて行うことが可能になりコストの節減につながる。また、連携が容易になることによって、今までシステム連携がネックになってスピードダウンしていたビジネススキームの再構築の加速化も期待できる。

#### (4) 小さなサービスから始め、価値を確かめながら拡張していくことができる

この要件を満たす IT システムは、開発期間を短い期間 (例えば 1 ~ 数週間) に分け、その期間内で実装可能な小さなサービスから開発・リリースし、利用者の反応を見ながら、段階的に拡張していく開発プロセス (アジャイル開発) やマイクロサービス等の手法により実現できる。従来の要件定義、設計、実装、テスト、運用を順番に行っていくプロセス (ウォーターフォールプロセス) では、要件定義で決めた要件を実現するシステムが運用するまでには数か月時には数年かかることがある。

ビジネス領域の中でスピードを求められる領域では、このような時間感覚では当初の要件が満たされ、IT システムが運用されるころには陳腐化してしまう。また、新しいビジネス領域では要件そのものが次第に変化していくことも少なくない。アジャイル開発によって、サービスを小さく生んで大きく育てていく段階で、常にビジネスニーズを確認し、微修正を繰り返せるようにすることで、常にニーズにフィットしたサービスを提供し続けることが可能になる。また、小さく生むことによって初期投資を抑えることができ、ビジネス的な有効性を検証したのちに拡大するという投資の効率化やリスクの低減にも寄与できる。