

Always-Onを実現するための 4つのステップ

Always-On : いつもそこに、動き続ける世界のためのプラットフォーム



目次

2	エグゼクティブ・サマリー
2	業界の傾向
3	企業のニーズと課題
5	ステップ 1: 査定と評価
7	ステップ 2: 計画と設計
10	ステップ 3: 実装とテスト
10	ステップ 4: 管理と維持
11	IBM をお勧めする理由
11	詳細情報

エグゼクティブ・サマリー

今日の情報に満ちた社会は、クラウド、分析、モバイル、ソーシャルに依存しています。移動する利用者は、ほとんど常にインターネット接続デバイスを利用できる範囲にいます。ユーザーは、24 時間体制でビジネス上重要なデータやアプリケーションに素早く一貫してアクセスできることを期待しています。そのため、計画的であっても計画外であっても停止は収入、利用者の期待、企業評価に悪影響を与えます。

エンド・ユーザー・エクスペリエンスを損なわず、サービスを中断させないAlways-Onのプラットフォームは、企業の強力な競争的差別化要因となります。Always-Onのプラットフォームとは、どのようなものでしょうか。Always-Onとは、利用されているビジ

ネス・サービスを中断せず、ダウン時間がエンド・ユーザーに認識されない連続稼働を実現することです。Always-Onにより、以下が可能になります。

- ネットワーキング・コンポーネント、ハードウェア、オペレーティング・システムなどに影響を与えるコンポーネントの障害に耐える
- ハリケーン、竜巻、洪水、その他の自然災害などの大災害に耐える
- 定期的な更新やセキュリティ・パッチなどの変更を中断なしに導入する

しかし、このような目標を達成できるAlways-Onのプラットフォームの計画、設計、実装、管理は困難な作業となることがあります。企業は、ユーザー、プロセス、アプリケーション、テクノロジーを網羅する適切なアプローチを用意する必要があります。Always-Onを成功させるには、Always-Onを企業の不可欠な一部に組み込む必要があり、企業レベルのコミットメントが必要です。

本ホワイト・ペーパーの目的は、Always-Onのプラットフォームのニーズに影響を与えている現在の市場の傾向と企業の要件を説明することです。その後、従来の高可用性 (High Availability) と災害復旧 (Disaster Recovery) のソリューションからAlways-Onのデザインに移行するための実証済みの 4 つのステップの詳細を説明します。

業界の傾向

デジタル形式のエンゲージメントの発展がAlways-Onの需要を促進
「自社の IT インフラストラクチャーがモバイル・テクノロジー、ソーシャル・メディア、ビッグデータ、クラウド・コンピューティングの需要に対応するために完全に準備できていると報告している企業は 10% 未満です。」¹

従来、企業は、企業データを管理して保管するために SoR (Systems of Record、定型業務処理システム) に依存していました。しかし、ビッグデータ、クラウド、モバイル、ソーシャルの導入に伴い、新しいデジタルの SoE (Systems of Engagement、協働のための情報活用システム) が企業の機能に影響を与え、急速にデータを生成していることが明らかになっています。従業員、利用者、パートナーはスマートフォン、タブレット、ノートブック、デスクトップなどの多様なデバイス間で自由に移動しています。ビジネス・アプリケーションは常時アクセスされているため、デジタル・チャンネルは、販売、ソリューション、サポートのライフサイクル全体の申核となっています。

下記は、今日のビジネス・プロセスとソリューションを方向付けているいくつかの傾向の定義です。以下のようなものがあります。

- **クラウド。**クラウド・コンピューティングは、インターネット経由でコンバージド・インフラストラクチャーとシェアード・サービスをサポートします。クラウドは、オンプレミス、オフプレミス、ハイブリッドに導入できます。
- **ソーシャル・メディア・チャンネル。**Facebook や Twitter などのソーシャル・メディア・チャンネルは、世界中のユーザー間のリアルタイムの対話をサポートします。
- **モバイル・アプリケーションと変更の速度。**利用者は、自分の携帯電話に定期的に更新が適用されることに慣れており、いつでも、どこでも、卓越したユーザー・エクスペリエンスを得られることを期待しています。そのため、モバイル・アプリケーションの活用はますます普及しています。
- **Software Defined Environment (SDE)。**SDE は、ストレージ、計算処理、ネットワークを最適化できます。IT 部門は、可能な限り効率的な方法で事業部門 (LOB) にサービスを提供できます。
- **エキスパート・インテグレートッド・システム。**これらのシステムは、ハードウェアとソフトウェアの両方を含む構成済みソリューションで構成されており、専門知識を取り込み、設計段階から統合されており、作業を簡素化します。例えば、IBM PureSystems が挙げられます。

- **継続的デリバリー/開発と運用 (Dev/ops)。**このプロセスにより、アジャイル開発者は、計画停止時間中に大規模な「ウォーターフォール」型のようなリリースを行うのではなく、継続的かつ反復的なリリースを行うことができます。このプロセスにより、開発チームと運用チームは緊密に協力して、ペースの速い変更をサポートすることができます。
- **継続的運用。**計画的ダウン時間や計画外ダウン時間の影響を受けずに、常に運用ビジネス・サービスを維持できることです。

このような傾向に対応するには、企業は、IT サービスを進化させて、さらに俊敏に利用者や企業の要求をサポートする必要があります。今こそ、従来の SoR ではなく、従来とは異なる方法で (さらに効率的に) SoE の管理を開始すべきです。

企業のニーズと課題

前のセクションで説明したような特徴を持つ新しいデジタルの傾向や SoE を踏まえると、ビジネスに対する停止の影響は大きくなっています。デジタル・チャンネルは、市場シェアを増やしており、収入創出を促進しています。停止が原因でサービスを失うと、収入や生産性の損失などの多くの影響が生じ、今日ではダウン時間は許容できないものとなっています。利用者は、すぐに利用を中止して、ニーズを満たすために競合サービスを使用するようになります。

利用者のニーズを第一に

可用性に対するユーザーの認識は、ビジネスの基本理念です。例えば、ユーザーが預金するためにモバイル・バンキング・アプリケーションにアクセスしたい場合、いつでもログインしたいときに、そのアプリケーションを利用でき、そのアプリケーションが機能することを期待します。グローバル化が進む今日の経済においては、サービスを中断させる「計画保守時間」は許されなくなっています。ユーザーがサービスにログインしようとして、システムが停止していることが分かった場合、そのベ

ンダーに不満を持つ可能性があります。ユーザーはソーシャル・ネットワークにログインして、その状況についての不満を友人に伝えるかもしれません。そのダウン時間のニュースはソーシャル・メディアを介してすぐに広まります。

コストの課題への対応

「(SoR などの) 基幹業務アプリケーションの全般的なサービス品質は向上しましたが、ダウン時間に起因するコストが増加しているために、大半の企業にとってはサービス可用性はまだ『十分に良い』とは言えません。」²

Always-Onのソリューションに対するニーズが存在することは明白です。しかし、コストに関連する阻害要因がいくつか存在します。企業は、関連コストを誤解していたり、Always-Onによって実現できる価値を定量化する上で問題を抱えている可能性があります。適切な計画とライフサイクルに基づくアプローチを採用せずに、その場しのぎでAlways-Onのプラットフォームを構築すると、大量の保守が必要になったり、サービス・レベル・アグリーメント (SLA) を達成できなくなったりする可能性があります。Always-Onは、ハードウェアとソフトウェアを超えるものです。設備コストとリソース要件は、多くの場合は初期実装コストを超過します。

コストの課題に対応するには、Always-Onについて総所有コスト (TCO) の観点から検討する必要があります。また、Always-Onを採用しないことのリスクも検討する必要があります。ダウン時間に関連する影響としては、収入や e-コマースの損失、ダウン時間や期限切れに対する規制の罰金、顧客への補償、企業評価の損害、従業員の生産性低下などが挙げられます。

「複数の業界にわたる Fortune 1000 企業を対象とした IDC の調査によると、1 時間当たりの重要なアプリケーション障害の平均コストは 50 万ドルから 100 万ドルです。」³

検討すべき質問

今日のビジネス・ドライバーを検討する際、ご自身の要件を詳細に検討することが重要です。検討すべき重要な質問のいくつかを下記に示します。

1. 計画保守時間を回避するために、停止時間ゼロの変更を実施する必要がありますか？
2. 重要なサービスの災害復旧ではなく、少なくとも高速なフェイルオーバーを必要としていますか？
3. 顧客はAlways-Onのサービスを期待していますか？
4. 大災害が発生した場合、顧客に引き続き中断なしにサービスを提供する必要がありますか？
5. コンポーネント障害が原因で、長期のダウン時間やビジネス・サービスの中断が発生していますか？
6. デジタル・チャンネルが増大していて、デジタル・プレゼンスが高まり、収入が増えていますか？
7. 事業部門 (LOB) が IT プラットフォーム内の変更を増やしていて、運用を中断させていますか？

上記のいずれかの質問が該当する場合は、今こそアプローチを変えるべきです。この分野における IBM の経験を踏まえ、IBM は、IT とビジネスの一貫した同期を維持するよう設計した体系的な方法でユーザー、プロセス、情報、テクノロジーを統合するAlways-Onのライフサイクル方法論を提案します。この方

法論は、4つの段階で構成されています。まず、査定と評価から始めて、その後、計画と設計、実装とテスト、管理と維持の段階に移行します。定期的に各段階を振り返り、Always-Onのプラットフォームが進化するビジネス要件や急速に変化するテクノロジーと同期した状態を維持していることを確認する必要があります。



図 1. Always-Onに移行するための4つのステップ。

ステップ 1: 査定と評価

第1のステップでは、現行の環境の査定と評価を行います。持続的に利用できるようにしておく必要があるビジネス・サービスに対する理解を深め、ビジネス目標を達成するための現在の能力を査定する必要があります。下記に、開始方法の例をいくつか示します。

- **現行のプロセスを確認します。** 査定には、企業の現行のプロセスと、可用性に対する要件の評価を含める必要があります。また、以下も実施する必要があります。
 - 可用性と継続的運用の要件に基づいてアプリケーションを査定して分類します。

- 運用業務の成熟度を査定します。
- 可用性に関する組織的な成熟度レベル（説明責任など）を評価します。
- ビジネス・インパクトに基づいてコストを検討します。
- **各アプリケーション・ドメインを詳細に検討します。** 各アプリケーション・ドメインを慎重に確認して、アプリケーションの機能を判断します。下記に例を挙げます。
 - アクティブ/スタンバイ。これは、障害発生時に利用できる「ウォーム」スタンバイ環境が準備されている従来型のITの復旧方式です。
 - パーティション・アクティブ。各サイトのアプリケーション・クラスターが、データベースと同様に独立して稼働します。一定の割合のユーザー・アクティビティはあるアプリケーション・クラスターに送られ、その他は別のアプリケーション・クラスターに送られます。

- アクティブ/クエリー。各サイトのアプリケーション・クラスターが稼働していて、読み取りはローカル・データベースから実行され、書き込みは1次データベースでのみ実行されます。
- アクティブ/アクティブ。これは、最良のシナリオと考えられています。両方のデータセンターが同じサービスを提供して、どちらのロケーションでもデータの読み取り/書き込みが実行され、両方のロケーションが同期されています。
- **継続的なサービスとアップグレードを検討します。**計画と設計のステップで、新しいアプリケーションとサービスの可用性要件を定義して文書化する必要があります。
- **ノイズを除外します。**すべてのアプリケーションにAlways-Onの可用性が必要なわけではありません。その点について査定時に判断する必要があります。例えば、一部の社内アプリケーションは、少なくとも初期段階では、このように高度な要件のターゲットとならない可能性があります。同時に、一般には、組織的な優先順位に基づき、大半のアプリケーションが時間とともにこのモデルに移行するだろうと考えられています。コストの観点からみると、すべてのビジネス・サービス・アプリケーションをAlways-Onにすることは現実的ではありません。例えば、ビジネス・サービス・アプリケーションをAlways-Onのプラットフォームで機能するように最新化する必要がある場合、資金調達、テスト、統合された運用モデルが必要となり、中断を伴う急激な変化となる可能性があります。
- **ギャップと阻害要因を特定します。**Always-Onのプラットフォームへの移行を妨げる可能性があるアプリケーション、運用上のギャップ、組織的なギャップを特定します。
- **リスクを分析します。**現在のリスクと潜在的なリスクを分析します。

このステップの実施中、テクニカル・コミュニティの中で信頼関係を構築して、企業でAlways-Onの観点 (POV) について他の意思決定担当者を教育できる経営幹部の支持者を見つけることが重要です。最高マーケティング責任者 (CMO) や最高財務責任者 (CFO) などの経営幹部のエグゼクティブとの会話を進めることができると、ビジネスの利害関係者と IT 部門の関係を強化できます。要件やニーズを生み出しているのはビジネスであり、IT 部門は、望まれる最終結果を提供していることを覚えておいてください。このような会話の内容として、売上創出について考えられる結果や、売上の段階的増加につながるデジタル・エクスペリエンスの改善といったビジネス価値を提案します。また、Always-Onを提供しないことによって生じるコストや企業評価への影響についても話します。IBM は、企業がカスタマー・エクスペリエンスを第一に考えられるように肯定的なアプローチをとることを提案します。継続的可用性が競争上の差別化要因となるというアプローチです。

また、モバイルへの移行に伴うユーザーのサービスの利用法や、極めて可用性が高いアプリケーションやデータに関連する期待についての議論も開始できます。このような考慮事項を念頭に置くと、企業はロードマップの計画を開始する準備ができます。

ステップ 2: 計画と設計

Always-Onを設計するには、アーキテクチャーに対する考え方を必要があります。変更時、計画停止時、計画外停止時も含めて、常に可用性を維持するビジネス・サービスという観点から考える必要があります。計画と設計のステップでは、査定と評価のステップ (ステップ 1) で収集した情報を使用して、実施可能な計画の基礎を形成します。ブリュワーの CAP 定理に支えられている常時オンの一般概念から始めることをお勧めします。

分散システムに対するブリュワーの CAP 定理では、テクノロジー・ソリューションは、次の 3 つの保証のうち、2 つしか提供できないとしています。

1. 一貫性: すべての分散ノードにおいて同時にすべてのデータの単一の最新のコピーが存在します。
2. 可用性: すべての要求は成功または失敗の応答を受け取ります。
3. 分断耐性: システムは、任意のメッセージ損失またはシステムの一部の障害に対して継続して動作します。(例えば、ネットワークがサーバー間でのメッセージ配信を停止する場合。)

ブリュワーの限界を踏まえると、すべてのビジネス・サービスのビジネス要件を正確に査定して、ビジネス要件に基づき、特定のサービスに適用すべきアーキテクチャー・パターンを判断することが極めて重要になります。ビジネス要件に加えて、アプリケーション・アーキテクチャーとプラットフォームを査定し、ブリュワーの 3 つの保証のうち、提供できないものを軽減する方法を判断する必要があります。⁴

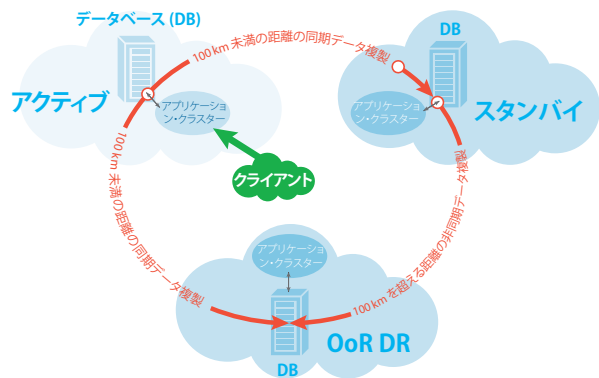
計画と設計の段階に以下のアクティビティーを取り込むことをお勧めします。

- Always-Onの設計アーキテクチャーに指針を適用します。
 - 考え方を変えましょう。この指針は、実践者、アーキテクト、エンジニアが過去に学んだ従来の HA と DR の手法を超えて、回復力と柔軟性を実現する革新的な概念を受け入れるためのガイドとなります。
 - 例えば、以前は用途が災害発生時に厳密に限られていたシステムやインフラストラクチャー (地域外 [OoR] データセンターなど) を組み込むことが必要になる場合があります。Always-Onに移行すると、OoR データセンターを稼働中のワークフローに統合して、計画保守停止を実施するのではなく、変更の実行時における回復力のために使用できます。
 - 詳細情報については、次の IBM Redpaper で IBM が説明している 21 の指針を確認してください。「[Always On: Assess, Design, Implement, and Manage Continuous availability](#)」

- **Always-Onを実現するためのテクノロジーの使用方法の計画を策定します。**
 - 製品の機能を判断します。例えば、グリッド・テクノロジーを使用してマルチサイト・クラスタリングをサポートする Java アプリケーションを確認します。
 - アプリケーション設計の原則を検討します。例として、サービス指向アーキテクチャー (SOA)、SOA とアプリケーション・プログラミング・インターフェース (API) を介した SoR からの疎結合 SoE、クラウド対応アプリケーション、クラウド・ネイティブ・アプリケーションが挙げられます。
- **運用上の課題への対応を作成します。**
 - 継続的デプロイメントをサポートします。
 - (スタックを使用して) 自動化を統合します。自動テストは、安定度と品質を向上させ、手動による手順をなくし、デリバリー・サイクルを加速します。
 - 停止発生後に機能するバージョンに戻すために必要なロールバックの計画を作成します (停止の発生がユーザーに認識されないことが望ましい)。さらに、開発者は、バグを修正するために環境を再現する機能を必要としています。
 - 並行バージョン管理を容易に行えるようにします。こうすると、企業は、新しいアプリケーション・バージョンの段階的な展開をサポートできます。多くの場合、バグがあった場合の実稼働環境での潜在的な問題を減らすために、アプリケーションの新規バージョンは段階的に実動に実装されます。
 - ビジネス (LOB) アプリケーション開発やプラットフォーム・プロバイダーとの関係と連携を構築します。
- **組織的な課題を考慮に入れます。**
 - 変更を受け入れて発展を主導する上級管理者のスポンサーや IT リーダーなど、変化をもたらす担当員を特定する必要があります。
 - 企業全体にわたってスキルに対する理解を深めることも必要です。例えば、アプリケーション・インフラストラクチャーで Always-On と無中断のアップグレードが可能になっても、アプリケーションの設計とビルドの方法が変わらなければ、アプリケーション所有者は使用できません。
 - エンドツーエンドの説明責任 (エンゲージメントから定常状態) をサポートします。
 - 継続的改善を容易に行えるようにします。
 - 部品の可用性ではなく、ビジネス・サービスの可用性に集中します。
- **継続的または Always-On の可用性の 4 つのパターンを計画します。** サービスを常時オンに発展させるためには、現行のアーキテクチャー設計のすべての部分を検討することが重要です。例えば、地域外 (OoR) データセンターを Always-On の設計に組み込むことを検討します。IBM は、現行のレジリエンシー・アーキテクチャーを Always-On に発展させるための 4 つの開発パターンを推奨しています。下記と図 2 の例を参照してください。

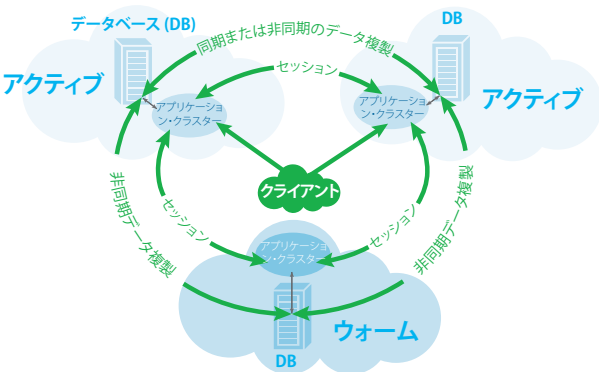
状態 1: 地域外 (OoR) 災害復旧 (DR) を使用するアクティブ/スタンバイ

標準的な従来型のアクティブ/スタンバイであり、災害を宣言して OoR に切り替える必要があります。ここで使用しているのは、企業が継続的可用性に移行するための開始点の候補を示すためです。



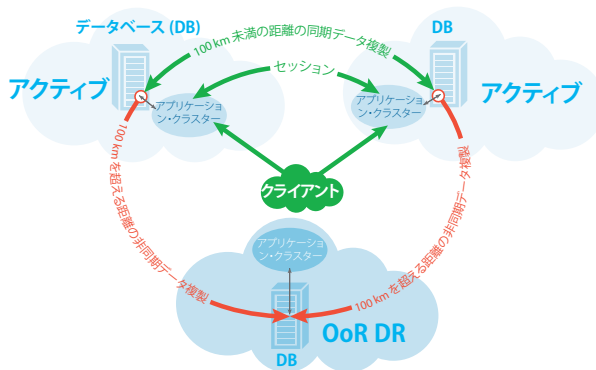
状態 3: 2つの地域外 (OoR) のアクティブ/クエリー

このパターンは、OoR DR を使用する 2つのアクティブという上記のパターンを成熟させたバージョンです。DR のシナリオで OoR の「クラウド」やデータセンターを使用するのではなく、日常的な運用に組み込まれており、分析、レポート作成、バッチ処理、読み取り専用照会に使用できます。実際、都市圏内の両方のアクティブ・パターンに影響を与えるコンポーネントで保守が必要となった場合に、アクティブ・コンポーネントとして使用できます。



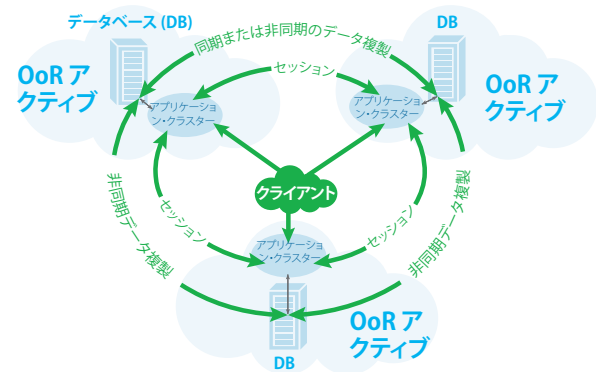
状態 2: 地域外 (OoR) 災害復旧 (DR) を使用する 2つのアクティブ

これは一般的なパターンであり、通常は成熟した金融セクターで使用されます。このような環境では、営業時間中に継続的可用性が必要であり、目標復旧時点 (RPO) は 0、データ整合性要件は原子性、一貫性、独立性、永続性 (ACID) のデータ整合性です。



状態 4: 地域外 (OoR) を使用する 3つのアクティブ

これは、ibm.com を 2001 年 6月から Always-On で維持するために使用されている 3アクティブ・モデルです。このパターンを実現している重要なビジネス上の意思決定は、最終的なデータ整合性です。データは、3つの「クラウド」のいずれにも書き込むことができ、ソースで取り込まれ、データセンター間の距離に基づく複製遅延で 2つのピアに適用されます。



→ 片方向のデータ転送 ↔ 両方向のデータ転送 | 内側の円 (セッション) = SoE | 外側の円 = SoR

図 2.4 2つの継続的可用性パターンの全体像。

このプロセス全体を通して、ライフサイクルの運用と時間に伴うアプローチの変化を検討することが重要です。可用性と継続的デリバリーの機能を設計して、アプリケーションとインフラストラクチャー・アーキテクチャーに組み込みます。ロードマップでは、組織的な成熟度を高め、必要に応じて繰り返し利用するために必要となるステップに対応する必要があります。

ステップ 3: 実装とテスト

計画が整ったら、開始します。このステップで重要な要因の1つは、現実的なテストと、大規模なサービス中断が発生した場合に「ビジネスを運営」できるという最終結果について考えることです。以下をお勧めします。

- **Always-Onの計画の実装を開始します。**
 - Always-Onの計画を達成するために必要なテクノロジー、システム、プロセスを実装します。(例えば、一度に1つずつ「クラウド」またはテスト・サイトを構築して、その後で相互接続します。)
 - 現行と構想中の企業アーキテクチャーに沿って継続的デリバリーと継続的可用性をサポートするためのスキルに投資して、そのスキルを維持します。
 - 継続的運用について検討します。継続的デリバリー (継続的運用と停止時間ゼロの変更) と継続的デプロイメント (実動で使用するためのプラットフォームでのリリース) を組み込んで拡張することもできます。
 - 移行中と準備テストでのライフサイクル運用を検討します。
- **Always-Onの戦略をテストして、顧客に受け入れてもらいます。**
 - 運用で軽減技法を使いこなして自動化できるように、「どのように機能して、どのように失敗するか」を理解します。
 - 実動障害テストの一部として、俊敏なインシデント管理プロセスを実装します。

- サービスが指定の目標を達成するように、テストを設計して実行します。
- 戦略がAlways-Onのサービスをサポートしていることを確認します。
- 改善と準備における弱点を特定するために、実地検証を実施します。

ステップ 4: 管理と維持

Always-Onの実現は、「1回で終わる」プロジェクトではありません。事前対応型のサービス管理は、実装を成功させる上で貴重な指針となります。

以下をお勧めします。

- **現行の状態を監視して、リスクを検出し、対応します。**
 - 誤動作や問題が発生する前に、監視してイベントの傾向を判断し、異常を検出することが重要です。反応的なサービス管理から事前対応型のサービス管理に移行すると、長期的にコストと時間を節約できます。
- **規制要件へのコンプライアンスを管理します。**
- **パフォーマンスについて報告します。**
 - 報告により、企業投資の準備と結果を実証できます。
- **再査定します。**
 - 持続的にポリシーを振り返ることで、ビジネス要件のプロセスの一部として、ビジネス・インパクトに基づき、可用性の要件とコストを継続的に定義して文書化することができます。ビジネス・インパクトと査定の持続的なプロセスの一環として、可用性要件を更新できます。

IBM をお勧めする理由

IBM は、過去 15 年間にわたり、Always-Onの環境を設計して、実装し、管理してきました。長年にわたる専門知識を活用して、Always-Onのプラットフォームへの移行を支援できます。IBM は、可用性に関するビジネス・サービス要件を理解して、現行のビジネス・サービス・アプリケーション・アーキテクチャーを確認し、現行の運用構造と組織構造を理解して、事業と IT の関係性を判断することで、お客様の最も重要なビジネス・サービスを判別できます。

IBM の専門家、査定アクティビティで収集された情報を使用して、Always-Onの実現の阻害要因を分析して特定し、ロードマップを最後まで策定します。対話式のワークショップを通じて、最新の指針とパターンを使用し、最も適切なパターンに合わせてビジネス・サービスを調整します。この作業により、どのビジネス・サービスがAlways-Onを実現でき、実現できないのかに対する洞察を得られます。また、アプリケーション、インフラストラクチャー、ユーザー、プロセスを調整するために必要な投資を特定して、Always-Onのビジネス・サービスへの個々の方向性を定義します。

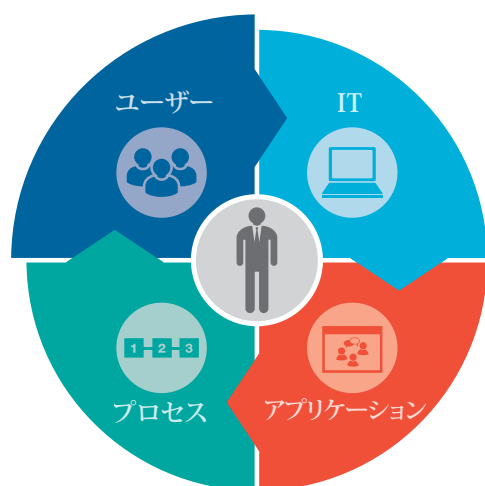


図 3. Always-Onのアプローチは、ユーザー、プロセス、アプリケーション、テクノロジーを網羅している必要があります。

40 年以上にわたり事業継続と災害復旧に携わってきた IBM は、効果的なレジリエンシー・テスト・プログラムの作成を支援できます。回復力に富むビジネスを構築するための IBM のアプローチは堅固で、世界中のリカバリーとマネージド・サービス・センターでの数千のレジリエンシー・テストの調整と管理を組み込んでいます。さらに IBM 社のポータル [ibm.com](https://www.ibm.com) は、2001 年 6 月から Always-On になっています。IBM がこれを実現できた理由は、運用モデル (ユーザーとプロセス)、アーキテクチャー・モデル、そして堅固なプラットフォームにビジネス・アプリケーションを搭載したい開発チームにアプリケーション要件を委任できたことにあります。

IBM のコンサルタントは、テスト済みの方式やツールを用いて、お客様が豊富な機能を備えたレジリエンシー・ソリューションを定義、実装、管理するための支援をします。

詳細情報

IBM の事業継続とレジリエンシー・サービスの詳細については、IBM 担当者にお問い合わせいただくか、次の Web サイトをご覧ください。[ibm.com/services/jp/ja/it-services/business-continuity/](https://www.ibm.com/services/jp/ja/it-services/business-continuity/)

さらに、IBM グローバル・ファイナンスは、お客様がビジネスに必要な IT ソリューションを可能な限り費用対効果に優れた戦略的な方法で取得できるように支援します。信用資格のあるお客様のために、ビジネス要件に合った IT ファイナンス・ソリューションをカスタマイズして、効果的なキャッシュ管理を可能にして、総所有コストを改善します。IBM グローバル・ファイナンスは、重要な IT 投資の資金を調達してビジネスを前進させるための最もスマートな選択肢です。詳細情報については、[ibm.com/financing/jp/](https://www.ibm.com/financing/jp/) にアクセスしてください。



日本アイ・ビー・エム株式会社
〒103-8510
東京都中央区日本橋箱崎町19-21

IBM ホーム・ページ:
ibm.com/jp

IBM、IBM ロゴ、ibm.com、PureSystems および Redpaper は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、
ibm.com/legal/copytrade.shtml

をご覧ください。本書の情報は最初の発行日の時点で得られるものであり、予告なしに変更される場合があります。すべての製品が、IBM が営業を行っているすべての国において利用可能なものではありません。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

本書に掲載されている情報は特定物として現存するままの状態を提供され、第三者の権利の不侵害の保証、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されています。IBM 製品は、IBM 所定の契約書の条項に基づき保証されます。

お客様は自己の責任で関連法規を遵守しなければならないものとします。IBM は法律上の助言を提供することはいたしません。また、IBM のサービスまたは製品が、お客様がいかなる法規も遵守されていることの裏付けとなると表明するものでも、保証するものでもありません。

- ¹ IBM Institute of Business Value「The IT infrastructure conversation」(2014年7月)。<http://www-03.ibm.com/systems/infrastructure/us/en/it-infrastructure-matters/it-infrastructure-report.html>
- ² Gartner「Hype Cycle for IT Service 1 Continuity Management, 2014」(2014年9月10日、John P. Morency、Carl Claunch、Pushan Rinnen、#G00263215)。
- ³ IDC「DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified」(Stephen Elliot、2014年12月、IDC #253155)。
- ⁴ IBM RedBooks「Always on: Assess, Design, Implement and Manage Continuous Availability」(Herbie Pearthree)。<http://www.redbooks.ibm.com/abstracts/redp5109.html?Open>

© Copyright IBM Corporation 2016



Please Recycle