

## シスコ アドバンスド サービスが推奨する ハイアベイラビリティ ネットワークの実現 (ライフサイクルで考える高可用性)

2008年12月



### 1. はじめに

昨今の世界的な厳しい経済状況により、ITは、その役割として一層のコストダウンを実現するとともに効率化による生産性の向上をますます強く求められている。この傾向は世界規模で企業活動を行っているグローバル企業において顕著であり、ITは戦略的な武器として活用し、難しい局面を乗り切るためのツールと位置づけられている。たとえばリアルタイムのビデオ会議システムなど、テクノロジーによって時間と距離の問題を解決し、迅速な意思決定を可能とするとともに、移動にかかるコストを削減している。このように戦略的なIT活用こそが、競合他社との差別化を図る鍵となる。

実際の投資においては、ビジネスと直結したROIを算出しやすいアプリケーションが優先されがちで、インフラストラクチャであるサーバー環境やネットワーク環境などは後回しにされてしまう傾向がある。確かにビジネスは常に拡大しなければならないものであり、そのためには収益を直接生み出すアプリケーションに注力しなければならない。ビジネスを常に成長させ続けないと競争優位が確立できず、企業そのものが淘汰されてしまう可能性もある。

たとえばアプリケーションを車のエンジンとすると、インフラストラクチャは筐体に相当する。エンジンの出力を上げることにばかり集中して、筐体が弱いまま放置すると、いつか支えきれずに崩壊してしまうのは自明である。インフラストラクチャは常に高出力を追い求めるアプリケーションを支え続ける、堅牢かつビジネスに貢献するものでなければならない。

しかし、残念ながら多くの企業において、ネットワークはビジネスと切り離して考えられることが多く、必要な都度拡張を続けてきた歴史がある。結果として多種多様な要件による一貫性のないネットワークとなり、拡張や運用がここにきて限界にきているケースが多く存在する。

継ぎはぎとなってしまったネットワークのもたらす弊害は、たとえば運用コストの上昇、トラブルシューティング時間の長期化、設計を熟知している担当者への過負荷などがあり、結果としてアベイラビリティ<sup>※1</sup>の低下を招くこととなる。

実際のデータにおいても、18%のユーザーが100時間以上の計画外ダウンタイムを経験している。

<sup>※1</sup> アベイラビリティ: 可用性のこと。表記には年間の稼働率が用いられる。

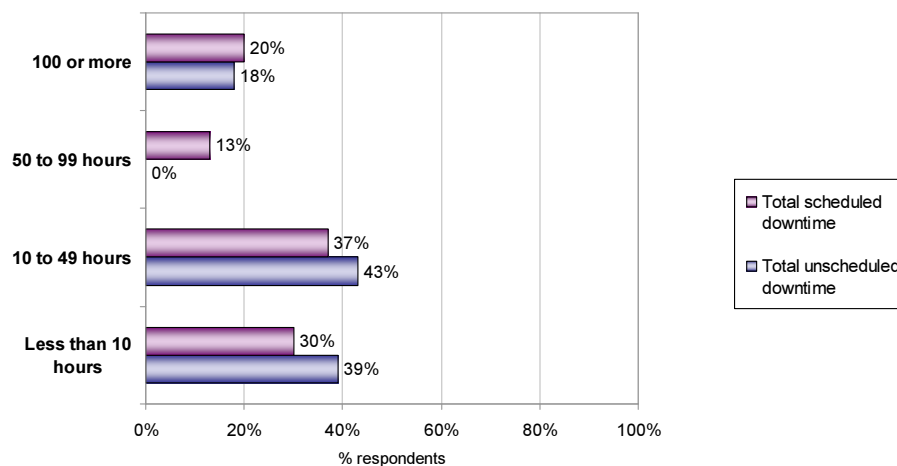


図1-1 計画外ダウンタイム時間<sup>※2</sup>

ネットワークはすべてのシステム アプリケーションの基礎となるものであり、堅牢でなければならない。ネットワークの停止は、システムが使用できないことによる売り上げ機会の損失に止まらず、従業員の生産性低下や復旧にかかるコスト、さらには顧客の信用失墜などのダメージをもたらす。企業のインフラストラクチャは、強力なアプリケーションを支えられる堅牢なものかどうかを、最初に確認することを強く推奨する。

コンプライアンスの観点からも、インフラストラクチャのアベイラビリティは強く求められている。ネットワークと関連がある主な法令としては、すべての上場企業が対象となる金融商品取引法(日本版SOX法。以下J-SOX)がある。

財務諸表の正確性保証を主たる目的とするJ-SOXにおいて、企業の経営サイドおよびIT部門は、財務諸表と直接関連する財務アプリケーションなどの業務処理統制に注力して対応を進めている。しかしインフラストラクチャなしで動作するアプリケーションは存在せず、インフラストラクチャにおける可用性は業務処理統制に多大な影響を及ぼす。よってJ-SOXにおいてはインフラストラクチャも別途リスクの精査、遺漏のない手順、他に影響を与えない環境などIT全般統制が必須とされている。

ネットワークも含めてシステムという認識を持ち、アベイラビリティを高く維持する以外に、J-SOXで求めている内部統制を満たすことは難しい。

USでの上場廃止事例から分かるように、コンプライアンスも重要なビジネス要件のひとつである。ネットワークにおける可用性を企業全体の問題として捉え、バランスの良い投資をすることが非常に大切となる。

### ハイ アベイラビリティの実現

それでは、アベイラビリティを高く保つ、すなわちハイ アベイラビリティ(High Availability; HA)を実現するにはどうしたらよいか。

アベイラビリティを測定するには稼働率が用いられ、一般に次の式で表される。

<sup>※2</sup> 出展: IP Service Provider Downtime Study Analysis of Downtime Causes, Costs and Containment Strategies, August 17, 2001, Prepared for Cisco SPLOB

**アベイラビリティ:**

$$\text{稼働率} = \text{MTBF}^{※3} / (\text{MTBF} + \text{MTTR}^{※4})$$

よってHAを実現させるには、MTBFを延長させ、かつMTTRを短縮すれば良いこととなる。

MTBFを上げるための要素のうち、代表的なものを以下に示す。

1. 冗長構成によるSingle Point of Failure (以下SPoF)の駆逐(冗長部の切り替えにおいても通信が継続できる場合)
2. 設計を実装にぶれずに反映させ展開するための構成や設定のテンプレート化
3. テンプレートを最新に保ち、維持/管理すること

同様にMTTRを短縮する要素を以下に示す。

1. 保守性の良いシンプルな設計
2. 冗長構成によるSPoFの駆逐(冗長部の切り替えによる通信断を無視できない場合)
3. 障害検知から復旧までの効率よく漏れのないプロセス
4. 要員のトレーニング

MTBFを伸ばす要素の1および2は設計に依存する部分であり、3は運用プロセスに関わる部分である。

MTTRを短縮する要素の1および2は設計に依存する部分であり、3および4は運用プロセスに関する部分である。

よってHAを実現するためには、HAを考慮した設計と、整った運用プロセスが不可欠であることが分かる。

また直列システムと並列システムのアベイラビリティの算出式では、

**直列システム:**

$$\text{稼働率} = \text{装置1の稼働率} \times \text{装置2の稼働率}$$

**並列システム:**

$$\text{稼働率} = 1 - (1 - \text{装置1の稼働率}) \times (1 - \text{装置2の稼働率})$$

となり、仮に装置1および装置2の稼働率を0.9とすると、直列システムの81%に対し並列システムにおいては99%の稼働率となる。以上のことから、装置単体の稼働率に頼ったネットワーク設計をせず、装置に障害が発生しても動き続ける冗長性を確保したネットワーク設計としなければならない。

以降の章では、設計および運用に関してHAを実現するための方法を、シスコが提唱するネットワークのライフサイクルである準備(Prepare)、計画(Plan)、設計(Design)、導入(Implement)、運用(Operate)、最適化(Optimize)に合わせて解説する。

※3 MTBF:ある機器やシステムが故障するまでの時間、平均故障間隔

※4 MTTR:ある機器やシステムが故障から復旧するまでの平均値、平均復旧時間

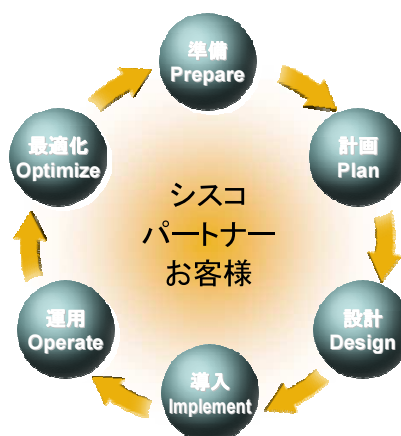


図1-2 シスコ ネットワーク ライフサイクル

## 2. 準備 (Prepare)

第1章「はじめに」で述べたように、ビジネスとITが乖離しているとビジネスの観点からネットワークに要求される方針が徹底されず、ネットワークの複雑化を招き、結果としてアベイラビリティを低下させてしまう。その事態を防ぐためには、ビジネスとITの結びつきを目的としたエンタープライズアーキテクチャ<sup>\*5</sup>（以下EA）の手法をネットワークの設計にも取り入れることが非常に効果的である。

ビジネス要件、アプリケーション要件、技術要件から、ネットワークに求められるTo Beの姿を描き出せば、そのネットワークは何のために存在するのか明確になり、アプリケーションと合わせたROIの算出も可能となる。また方針を遵守するガバナンスを徹底することにより、シンプルなネットワークを維持することが可能となり、結果HAネットワークが実現される。

準備フェーズの目的は、プロジェクトのビジネス貢献度、想定されるリスク、回収期間などを元に優先順位を付け、戦略的な投資の判断を元にプロジェクトを立ち上げることにある。投資判断ではEAにおけるAs-Is分析などの現状分析が必要となる。ネットワーク関連のプロジェクトでは、既存のネットワーク環境における課題/問題点を洗い出し、次期ネットワークで解決すべき目標として定義することが実際の活動となる。

既存ネットワーク環境の調査は、HAネットワークを実現する上でも非常に大切な要素である。新規にネットワークを導入するという理由で現状調査や課題抽出を省いてしまうと、新ネットワークにおいても既存のネットワーク同様の課題や問題が再発するリスクを内在したままとなる。

既存のネットワークには、設計面や設定面での顕在化していないリスクも複数存在するのが一般的である。通常運用での課題把握だけに止まらず、あるべき姿を描く上での重要なインプットとして、既存の環境を徹底的に調査することを強く推奨する。

EAの手法に則ったネットワーク設計の概念を以下の図に示す。

<sup>\*5</sup> エンタープライズ アーキテクチャ: 全体最適の視点で業務やITの標準化を促進し、かつビジネスとITを結びつける方法論



図2-1 EAIによるネットワーク設計

既存ネットワークのリスク洗い出しについては、以下の内容を網羅する必要がある。

- 過去の障害履歴、障害の傾向分析
- 既に認識している問題点、実装上の課題などの整理
- ベストプラクティスとの対比によるネットワーク設計の強みと弱みの分析
- ネットワーク運用プロセスの精査と分析

既存環境で発見されたリスクを、次のネットワーク設計では解決されるべき課題や問題点として整理しておく。障害や設計面だけにとどまらず、必ず運用プロセスに関する問題点および課題を整理することが必要である。

### 3. 計画 (Plan)

計画フェーズの主な目的は、言葉のとおりプロジェクトの計画を立てることにあり、詳細なスコープ定義、WBS<sup>\*6</sup>作成、スケジューリング、リソース確保など、プロジェクト実行に必要な一連の作業を行う。

また本フェーズにおいては、次工程である設計フェーズが遅滞なく行われるように、設計上重要となる要件の洗い出しを進めておくことが望ましい。

中期経営計画などに代表される今後の経営戦略、拠点の拡張計画などのビジネス上の要件、既存および導入予定のアプリケーションがどのようなトラフィック特性を持ち、どのようなネットワークに対する品質要求があるのかなどのアプリケーション要件を把握しなければ、今後設計するネットワークがそれらの要件を十分に満たす保証はないに等しい。

また、すべてに該当する冗長策を講じたり、回線速度を豊富に割り当てたりする技術面からスタートしたボトムアップの手法では、不要な部分にもリソースを割り当てることとなり、無駄な投資となる可能性が非常に高い。

ビジネス要件およびアプリケーション要件からトップダウンで設計するからこそ、投資の妥当性評価が可能となり、実際にお金を生み出すアプリケーションの成長予測を元にしたネットワークのROIを算出し、投資計画にフィードバックすることができる。

技術面での調査も欠かすことができない。数年後を見据えてどのような技術が主流となるか、自社にあってはいる技術は何か、その技術の柔軟性は、既存の技術との親和性などを調査/解析し、ビジネス要件やアプリケーション要件と合わせてテクノロジー要件として整理する必要がある。メーカーが提供するベストプラクティス情報や、先進企業の事例に関する情報収集も必要である。

<sup>\*6</sup> WBS: プロジェクトに必要なタスクを分解し、階層的に構造化する手法

以上の要件定義が、次フェーズで行う設計の礎となる。

また本フェーズの別の側面では、今後のネットワークをどのような標準に準拠して設計するかを判断することが挙げられる。

業界内で時間をかけて洗練されてきた標準を採用することで、1からプロセスを開発する無駄な投資をなくし、かつ作業ミスや抜け・漏れといったリスクを排除した円滑な設計および運用が行えることとなり、HA実現の一助となる。また上場企業であればコンプライアンスのために従わなければならない標準もあるため、ネットワークのライフサイクルを、各々のフェーズやレイヤーに合わせたフレームワークやガイドラインに則して整えることにより、無駄な投資を抑え、かつ監査もスムーズに受けられることとなる。

ネットワークにおいて、関連の深い代表的なフレームワークおよびガイドラインを以下に示す。

- COSO

コーポレートガバナンス、効果的な内部統制、ビジネス倫理を通じて財務報告の品質の向上を行うUSのトレッドウェイ委員会組織委員会が定めたフレームワークであり、J-SOXにおいて定められる内部統制の元となっている。

上場会社すべてがJ-SOXの対象となり、財務報告の正確性を求められるため、上場企業は対応が必須である。

- COBIT

ITガバナンスの成熟度を測るためのフレームワークであり、調達や開発から運用まで、組織のプロセスの成熟度が判定されている。上記COSOのフレームワークに則って構築した内部統制環境のIT部分において、具体的に設計/調達/運用などの手順や環境などを取り決めるための指標となる。

- ISMS (ISO27001)

企業におけるリスクマネジメント体系であり、情報を扱う際の基本的な方針や、具体的な計画、計画の実施・運用、一定期間ごとの方針や計画の見直しなど、セキュリティに関する企業全体の体制が評価される。今日の情報システムにおいてセキュリティに対する対応は切り離せないものであり、セキュリティ事故はそのまま企業イメージの失墜や取引先からの信用の低下にもつながり、HAの観点ではITシステムのアベイラビリティを低下させる要因となるため、対応は必須である。

- ITIL (ISO20000)

ITシステムの運用や、管理業務の構築や維持をする際の体系的なガイドラインであり、ベストプラクティス集となっている。実際の運用業務において抜けや漏れなどを防ぐために不可欠なプロセスを定義しており、円滑な運用プロセスの実現のために必要となる。

以上のさまざまなフレームワークおよびガイドライン同士の関係をイメージで表すと、次の図となる。

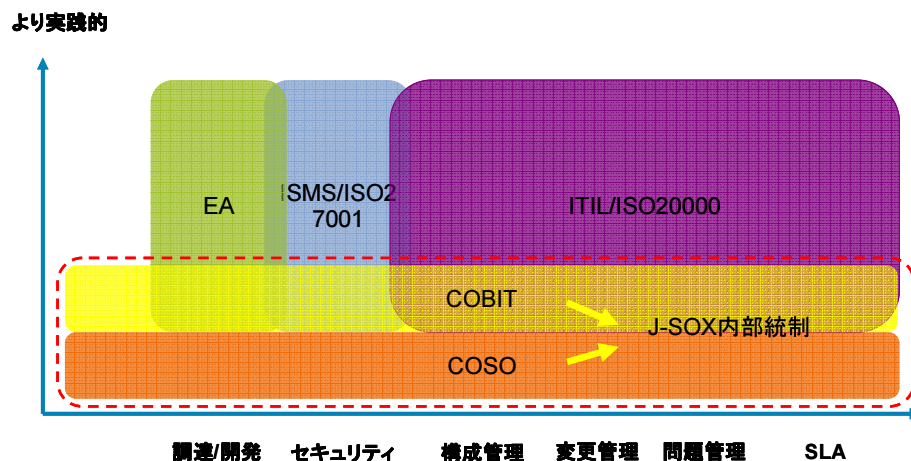


図3-1 各種フレームワーク、ガイドラインの関連

以上から分かるように、それぞれ対象とする範囲やフェーズ、深さなどが異なる。よって、全体のフレームワーク作りや実際のプロセスの策定など、実施する内容に合わせて適宜選択し、適用する必要がある。

#### 4. 設計 (Design)

設計は大きく分けて5つのフェーズに分かれる。

##### 1. コンセプト設計

コンセプト設計は、準備フェーズにて洗い出した既存ネットワークにおける課題、将来の計画も含めたビジネス要件、新規導入予定も含めたアプリケーション要件、今後の技術トレンドも鑑みたテクノロジー要件、ベストプラクティスや他社事例をインプットとしてそれらを満足させるTo-Beモデルを策定することが目的となる。

コンセプト設計では、To-Beモデルがどのように要件を満たし、かつ課題を解決するかの関連付けがなされている必要があり、目標とするアベイラビリティ値の設定なども重要な要素となる。既存ネットワークのアベイラビリティが数値として算出されていれば、より具体性のある目標値へのインプットとなり、IT部門、ユーザー部門双方に納得度の高いSLA (Service Level Agreement) 締結のための重要な要素となる。

上述したさまざまな要件を満たすネットワーク設計を、求められる機能、サービスレベル、セキュリティ強度などによりモジュールとして定義し、モジュール同士の接続要件などを定め、概要レベルの設計を行う。

##### 2. 基本設計

基本設計は、コンセプト設計にて策定した概要レベルの方針、モジュール構成などをインプットとし、モジュール内の具体的な構成設計を行い、機種選定まで落とし込める機能定義や、デバイス コンフィグレーションが作成可能なレベルへの各技術エリアに関する方針のブレイクダウンを行うことが主な目的となる。

基本設計のフェーズにおいては、冗長性や拡張性、ルーティングプロトコルなど、決定しなければならない項目は多岐にわたるが、網羅性も非常に大切な要素となり、慎重に確認する必要がある。新ネットワーク構成におけるアベイラビリティが数値で算出されていれば、運用プロセスにおける可用性管理およびサービスレベル管理への優れたインプットとなり、妥当性のある目標値の設定の一助となるため、可能な限り数値で算出することを推奨する。

基本設計書で必要な項目それぞれに対し、技術的な方針を詳細に検討/決定し、機器構成を決定し、最終的に設定のテンプレート案を策定することが基本設計でのゴールとなる。

テンプレート化の意義は、導入フェーズにおける設定作業の工数軽減およびミスの発生を軽減する効果がある。また運用フェーズにおいても、設定ミスによる障害を未然に防ぎHAに貢献する効果がある。またテンプレート化するには複雑さを排除するため、数そのものを少なくする、つまりシンプルなネットワークを実現することを念頭に置く必要がある。

シンプルなネットワークの実現のため、他に留意することは、モジュール内構成を同一機器に揃える、同一機種に使用するソフトウェアは同じバージョンかつ同じフィーチャーセットにするなどが挙げられる。

実際によく見られる負荷分散に関するケースとして、回線を効率的に使用するために、コンセプト設計を顧みず技術的な要件のみ照らし合わせて可能/不可能を論じてしまうことがある。安易なルーティングプロトコルのチューニングは、本来シンプルなネットワークを実現する上で大敵である。コンセプト設計の目的や内容、背景を再確認し、運用面に与える悪影響も合わせて考慮しながら本当に必要な時のみ実施する必要がある。

### 3. 検証

検証においては設計上期待された動作を確認するとともに、パフォーマンスを確認し、そこでのフィードバックを詳細設計にインプットし、最終的なデバイス コンフィグレーションの確定につなげていくこと、また本番環境に与える影響を判定することを主な目的とする。検証の中には単体検証、システム検証、UATの各段階がある。HAを実現するために非常に重要なフェーズであり、本番環境を導入した際に不具合が出ないように、テスト項目および手法などに関して慎重な設計を行う必要がある。開発、検証、UAT環境は本番環境と分離するようにCOBITにおいて規定されている。実運用を考えた場合でも、開発中や検証中のアプリケーションや機器などが本番環境に影響を与えることは重大な問題であり、アベイラビリティを低下させることになる。

コンプライアンスの観点からも実運用の観点からも、基本設計の段階で検証環境の設計を行い、試験が十分に目的を果たせる環境を整えなければならない。検証は本番環境と切り離すように設計した別環境やLabを利用して行い、決して本番環境で行ってはならない。検証結果は常に次に述べる詳細設計へフィードバックし、デバイス コンフィグレーションを確定することを助け、想定外の事象などが発生した際には原因究明し、必要であれば基本設計やコンセプト設計などの上流工程に戻って再設計することが大切である。

### 4. 詳細設計

詳細設計では基本設計で策定した技術的な方針および検証結果をインプットとし、デバイス コンフィグレーションを作成/確定し、テンプレート確定を目的とする。これ以外にもVLANの割り当て表やポート設定リストの作成、ラックのレイアウト図、フロアレイアウト図などの作成、機器保守に必要な情報を網羅したドキュメントの作成、運用プロセスへのインプットなどもまた大きな目的とされる。運用プロセスへの出力となるドキュメントの品質が、そのままMTTRを左右することを忘れてはならない。

### 5. 移行設計

移行設計は、通常非常に大きな工数を伴い、かつ新規システムへのリプレースなどの成否を分ける重要なフェーズであるため、基本設計を実施している時点から概要レベルで検討を同時進行で詳細化し、最終的には手順書のレベルにまでブレークダウンしていく必要がある。本フェーズのみ切り出して別プロジェクトとし、基本設計のプロジェクトと協調しながら実施するケースも多く見られる。

移行の手法に関しても一括で行う、段階的に行うなどさまざまな方法が存在するが、本番環境に影響を与えないことを念頭に、事情に合わせた方式で実施する必要がある。移行設計および手順書の中には、切り戻し条件や判断ポイント、完了条件や承認者などが定められていなければならない。

設計のフェーズは、前述したコンセプト設計～移行設計のようにインプットとアウトプットが決まっており、トップダウン方式で進める。技術面から積み上げたボトムアップ方式では、ビジネスやアプリケーションにネットワークが適合しているか否かの判断は難しいことが理由となる。

設計の各フェーズにおいては、設計する内容はもちろん大切な要素だが、同時にドキュメント化も非常に重要である。エンタープライズ環境では、設計書類が整っていないケースが散見されるが、ドキュメント化するからこそ、ユーザー部門からの新規要件に対し、方針に合致しているか否かの判断が可能となる。また運用プロセスに対する運用や保守資料としてのインプットにもなる。ISOの観点や内部統制においてもドキュメント化は必須の事項となる。

これまで文章で説明した設計の流れを以下に図示する。

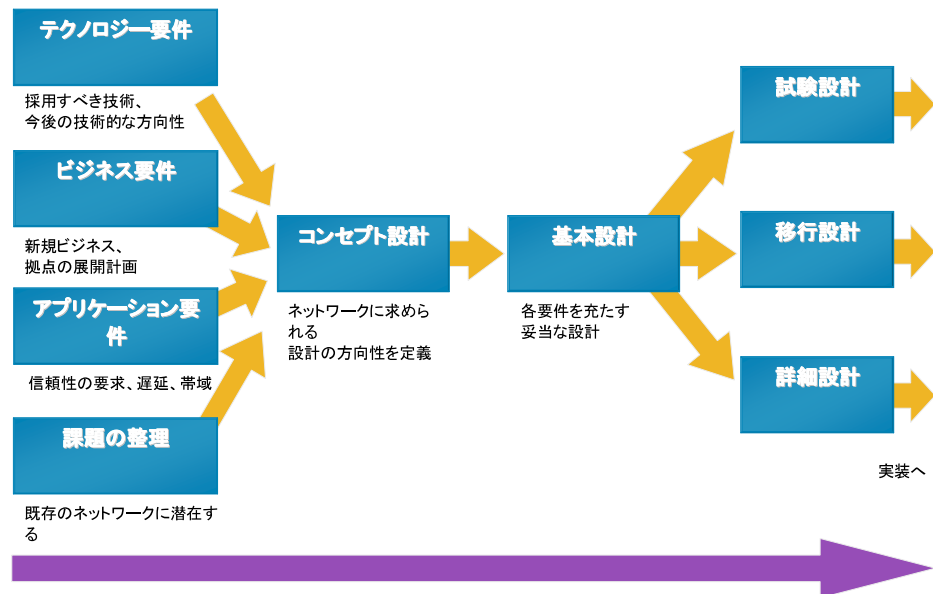


図4-1 ネットワーク設計の流れ

## 5. 導入 (Implement)

導入段階において重要な要素は、いかに影響なく本番環境に機器を導入するかを第一に考える必要がある。一般的にビックバン的な移行は24時間動き続けることを求められるエンタープライズ環境のネットワークでの適用は難しいため、段階的な移行計画を立てるケースが多い。たとえば拠点の展開などにおいて、パイロット拠点を作り暫く状況を観察や確認し、問題ないと判断した後、少しずつ展開を進めていくスロースタートの手法を採用することが望ましい。

実際の機器設定では、設計にて策定したテンプレートを活用して行うが、決して本番環境で設定を投入してはならない。必ず別環境で設定を完了し、試験を実施して問題なく動作することを確認した後、本番環境に機器を実装するプロセスとチェック体制が必要である。

また導入段階においては作業そのものが発生する。その際の注意事項として、作業担当者がその場で判断する事項を作らない手順書を作成しなければならない。作業担当者のスキルを当てにした部分などは含めず、誰が実施しても作業可能なレベルでなければならない。

手順書の中には、移行設計で作成したタイムテーブルや切り戻しポイント、承認者などが含まれている必要があり、手順書以外にもサイトの住所、地図、ビル名、階数、フロアレイアウト図、ラックレイアウト図、機種名、ホスト名などの情報を揃え、間違いなく対象の機器に対する作業ができるようにし、現地での作業ミスを防ぐようにする。

同様の情報は保守の時にも必須となる。問題なく目的の機器の前に到着し、作業を開始できることは、即MTTR短縮につながり、また作業ミスによる2次災害を防ぐ効果も見込めることとなる。

## 6. 運用 (Operate)

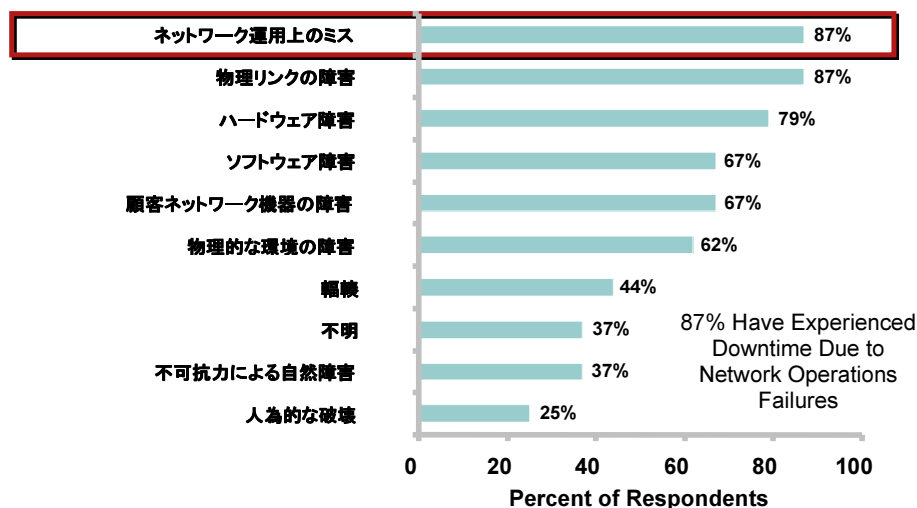


図6-1 ダウンタイムの要因

前記の図を見ても分かるように、ネットワーク ダウンタイムの9割弱はオペレーション上のミスによるものである。

オペレーションミスの原因の主なものとして以下の内容が挙げられる。

- **運用プロセスの不備**

運用プロセスそのものに遺漏があり、チェック体制の不備や、そもそもプロセスそのものがない場合である。対処方法としては、ITILなど業界標準のベストプラクティスに準拠する形で運用プロセスを構築すると効果が高い。整った運用プロセスは、実施項目が多く煩雑に見えるが、実際は洗練されており、ミスを防ぐ工夫などがプロセスの中に盛り込まれている。

- **間違った設定コマンドの投入**

本番環境で設定をダイナミックに変更しようとした際に生じやすい。上述した設定のテンプレート化により、設定コマンドを直接投入することを可能な限り避けることにより発生を低減することが可能となる。

- **タイプミス**

この原因に関しても、本番環境にて設定をダイナミックに変更しようとした際に生じるが、設定以外のコマンドに関しても生じる可能性がある。設定に関しては、テンプレート化とその活用を徹底することにより、発生を低減することが可能となる。

- **要員のスキル不足**

この原因に関しては、上述した間違った設定コマンドの投入やタイプミスなどを誘発する根本原因ともなり得る。ネットワークを運用していく上での必要なスキルを定義し、現状とのギャップ分析を行い、スキル向上や新しい技術を習得する機会を組織として考える必要がある。ネットワーク運用の現場にいるスタッフは常に多忙であり、スキルアップに時間を割けない問題がある。しかし必要な技術は常に進歩しており、最新の技術を取り入れる努力を怠ると、担当者の技術はすぐに陳腐化し、会社としても損失を被ることとなる。会社として、または組織としての強制力を持ちトレーニング計画を進めていくことが必要である。

またミスが起こりやすいユーザーの作業を可能な限り排除することも非常に有効なHA実現の手段である。そのためには自動化ツールを導入して運用をオートメーション化することが効果的である。

手動管理におけるデメリットを以下に示す。

表 6-1 手動管理の意味すること

サービス停止が多い	サービス停止やセキュリティインシデントの80%が人為的な設定ミスに起因する
ネットワーク可用性が低い	ネットワーク予算の80%がネットワークダウンを回避するために割り当てられている
労働集約型の変更管理	ネットワークエンジニアの49%の工数が手動でのネットワーク変更作業に使われている
複雑でコストのかかるコンプライアンス管理	コンプライアンス要件に適合させるための作業には、5倍以上のコストがかかる

自動化ツールの多くはCOBITやITILなどに則して開発されており、コンプライアンス準拠の程度をレポートする製品も存在する。自社ネットワークの規模や運用の煩雑さ、標準準拠の必要性などの条件に照らし合わせ、導入を検討することが望ましい。前記の表によれば手動管理のコンプライアンス要件への対応コストは、自動化されている場合の5倍以上かかり、また別のデータによると監査に関するコストも4倍以上かかると言われている。

また本フェーズにおける他の重要な考慮事項として、老朽対策およびセキュリティの脅威がある。

老朽対策に関して、ネットワーク機器やソフトウェアは製品としてのライフサイクルがあり、End of SupportやEnd of Lifeなどさまざまな段階でメーカーからアナウンスが出されている。サポート対象とならない機器や、部品の入手しづらい機器を使い続けることは、万が一の障害発生時での復旧に非常に時間がかかり、MTTRを増大させるリスクを内在する。自社で使用しているハードウェアおよびソフトウェアの製品サイクルを管理し、計画的な投資と対策をしていくことはHA実現において非常に大切な要素である。

次にセキュリティに関して本書では簡単に触れるにとどまるが、アベイラビリティを直接低下させるリスクとして、DoSアタックに代表されるサービス停止攻撃がある。サービス停止は、その内容によっては企業イメージの失墜に直結することもありかねない。自社のセキュリティポリシーに基づいて設計することは当然であるが、セキュリティ強度が高い設計をすれば終わりではない。新たなウィルスの出現やセキュリティホールが発見、アタック手法の変化など、セキュリティリスクは常に拡大していく特性を持っているため、設計時と変わらない対策ではすぐに陳腐化してしまい、気がつくと使えないレベルにまで低下してしまう。

メーカーからの情報を収集して常に高い注意を払い、最新のパッチをあてるなど、常に対策を怠らず、セキュリティ関連情報と対策を最新の状態に保つ努力が必要である。

## 7. 最適化 (Optimize)

ネットワーク停止につながるリスクをプロアクティブに防止するためには、運用プロセスの中に最適化の要素を盛り込む必要がある。

基本設計の章にてチューニングをなるべく避けた設計を心がける旨を述べたが、さらに機器のソフトウェアの選定において、必要な機能を最低限満たしたフィーチャーセットを使用することが重要である。使わない機能をサポートしたフィーチャーセットを使用することは、メモリーやCPUなどのリソースが無駄に消費されるだけでなく、使用していない機能部分の不具合に起因する影響を受け、障害となる可能性があることを認識しなければならない。将来使用するかどうかわからない機能をサポートするソフトウェアを使用するのではなく、シンプルなネットワークをシンプルなソフトウェアでサポートすることがさまざまなリスクからネットワークを守りHAを実現することの第一歩である。

コンセプト設計において機能別にモジュールを定義し、基本設計にてその内部を構成する機器やソフトウェアを揃える形での構成設計を推奨したが、そのメリットとして、部材配備の簡素化によるMTTR向上という面以外に、そのモジュールをひとつの集合とみなせることにより、ハードウェアおよびソフトウェアの管理が容易になることが挙げられる。機器およびソフトウェアが管理可能な状態になっていることで初めてプロアクティブなソフトウェアの選定などの最適化手段を適用することが可能となる。

具体例としては、設計段階のモジュール単位で、使用する機能/機種/ソフトウェアが整理されているので、必要とする機能と照らし合わせながらソフトウェアの不具合情報との照合を行い、使用するにあたり最適なソフトウェアのバージョンおよびフィーチャーセットを決定することがスタートとなる。次に、ソフトウェアは成熟度に合わせてアップグレードしていく必要があるため、次候補としてのソフトウェアを選定し、かつ現在導入しているソフトウェアからのアップグレードのトリガーもあらかじめ決めておく必要がある。

なお、次候補のソフトウェアに関しては、定期的にソフトウェアのスクリーニングを実施し、見直しを図ることにより鮮度を保つことも忘れてはならない。

さらにITILに基づいた考え方により、通常時の変更以外に緊急変更手順も定めておく必要がある。その際の手続きや承認者、承認プロセスを定義して手順に落とし込むことにより、コンプライアンスに準拠しながら、プロアクティブに対処し、アベイラビリティを低下させる要因をあらかじめ根絶しておくことがはじめて可能となる。

## 8. 総括

前章までに述べてきたことが、HAネットワークを実現するための概要である。設計や運用の要素同士は複雑な関係があり、単に設計の中だけ、運用の中だけでは収まらない。例として設計のアベイラビリティの目標値が、サービスレベル管理における妥当な目標値へのインプットとなるように、両者に密接な関係があることをご理解いただいた。

本書で述べてきた内容は、技術的なことはほとんど含まれていないが、設計や運用を進めていく上で考慮すべき重要な事項ばかりである。

本書が、企業ネットワークのアベイラビリティの向上に寄与できれば幸いである。

©2008 Cisco Systems, Inc. All rights reserved.

Cisco, Cisco Systems, およびCisco Systemsロゴは、Cisco Systems, Inc.またはその関連会社の米国およびその他の一定の国における登録商標または商標です。

本書類またはウェブサイトに掲載されているその他の商標はそれぞれの権利者の財産です。

「パートナー」または「partner」という用語の使用はCiscoと他社との間のパートナーシップ関係を意味するものではありません。(0704R)

この資料に記載された仕様は予告なく変更する場合があります。



シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先(シスコ コンタクトセンター)

<http://www.cisco.com/jp/go/contactcenter>

0120-933-122 (通話料無料)、03-6670-2992 (携帯電話、PHS)

電話受付時間：平日10:00～12:00、13:00～17:00

お問い合わせ先