高可用性ネットワーキング用の Cisco IOS 管理：ベスト プラクティス ホワイト ペーパー

内容

概要

信頼性の高いCisco IOS®ソフトウェアの導入と維持は、今日のビジネスに不可欠なネットワーク

環境において優先事項であり、ノンストップの可用性を実現するには、シスコとお客様の新たなフォーカスが必要です。シスコとしてソフトウェアの品質保証に重点的に取り組む必要がある一方で、ネットワーク設計グループおよびサポート グループでは、Cisco IOS ソフトウェア管理のベスト プラクティスにも重点を置く必要があります。目標は、より高いアベイラビリティと効率的なソフトウェア管理の実現です。そのために、ソフトウェア管理のベスト プラクティスの共有、学習、および実装を組み合せた方法を採用しています。

このマニュアルでは、企業とサービス プロバイダーの両方のお客様のための Cisco IOS 管理業務に有効な運用フレームワークについて説明します。このフレームワークは、ソフトウェアの信頼性の向上、ネットワークの複雑さの軽減、およびネットワークのアベイラビリティの強化を支援します。また、このフレームワークによって、ソフトウェア管理のテストや検証における、シスコのリリース工程とシスコ カスタマー ベースのそれぞれの責任範囲および重複部分が明確化されるため、ソフトウェア管理の効率性を改善するためにも役立ちます。

Cisco IOS ベスト プラクティスの概要

次の表に、Cisco IOS ベスト プラクティスの概要を示します。これらの表は、定義されているベスト プラクティスの管理の概要、最新の Cisco IOS 管理方法を確認するギャップ分析チェックリスト、または Cisco IOS 管理に付随したプロセスを作成するフレームワークとして使用できます。

これらの表では、Cisco IOS 管理における 4 つのライフサイクル コンポーネントを定義しています。各表は、特定のライフサイクル領域に関する戦略とツール概要で始まっています。その後に、それぞれの定義されたライフサイクル領域だけに適用される、特定のベスト プラクティスが続きます。

計画 - Cisco IOS の管理フレームワークの構築：計画は、Cisco IOS の管理における最初のフェーズであり、ソフトウェア アップグレードの時期、アップグレードする場所、およびイメージのテストと検証に使用するプロセスを企業が判断するために必要とされます。

ベスト プラクティス	詳細（Outcall Billing Detail）
Cisco IOS の計画における戦略とツール	Cisco IOS の管理計画は、現在の業務についての率直な評価、達成可能な目標の策定、およびプロジェクトの計画で始まります。
ソフトウェア バージョン トラックの定義	ソフトウェアの一貫性をどこで維持できるかを特定します。ソフトウェア トラックは、個々の地理的分布、プラットフォーム、モジュール、または機能要件によって他の領域から区別される、ソフトウェア バージョンの一意のグループとして定義できます。
アップグレード サイクルとその定義	アップグレード サイクルの定義は、ソフトウェアおよび変更管理における基本的な品質手順として定義でき、ソフトウェアのアップグレード サイクルの開始時期を判断するために使用されます。
認証プロセス	認証プロセスの手順には、トラックの特定、アップグレード サイクルの定義、候補管理、

	テストと検証、およびある程度以上の試験的な実稼動使用を含める必要があります。

設計 - IOS バージョンの選択と検証：Cisco IOS のバージョンを選択および検証するための明確なプロセスを用意することにより、アップグレードの失敗やソフトウェアの予想外の不具合に起因する、想定外のダウンタイムを削減できます。

ベスト プラクティス	詳細（Outcall Billing Detail）
Cisco IOS の選択および検証における戦略とツール	新しい Cisco IOS バージョンを選択、テスト、および検証するためのプロセスを定義します。これには、実稼動ネットワークを模倣したネットワークのテスト ラボも含まれます。
候補管理	候補管理では、特定のハードウェアおよび有効な機能セットに対する、ソフトウェア バージョン要件および潜在的なリスクを識別します。
テストと検証	テストと検証は、ソフトウェア管理およびハイ アベイラビリティ ネットワーキングの重要な側面です。適切なラボ テストを実行すると、生産のダウンタイムが大幅に削減され、ネットワーク サポート担当者のトレーニングに役立ち、ネットワーク実装プロセスの簡素化にも役立ちます。

実装 - 迅速で正常な Cisco IOS の導入：明確に定義された実装プロセスを使用すれば、組織は、新しい Cisco IOS バージョンを迅速かつ正常に導入することができます。

ベスト プラクティス	詳細（Outcall Billing Detail）
Cisco IOS の配備における戦略とツール	Cisco IOS の配備における基本的な戦略は、アップグレード ツールと明確な実装プロセスを使用して、試験プロセスおよび迅速な配備を通して、最終的な認証を行うことです。
試験プロセス	障害の発生する危険を最小限に抑え、実稼動における未解決の問題をより確実に把握するために、ソフトウェアの試験的導入を行うことをお勧めします。個々の試験計画ごとに、試験の選択、試験の期間、および測定方法を検討する必要があります。
実装	試験フェーズが完了した後に、Cisco IOS の実装

	フェーズを開始する必要があります。実装フェーズには、スロースタート、最終認証、アップグレードの準備、アップグレードの自動化、および最終検証などの、ソフトウェア アップグレードの成功および効率性を確認するためのいくつかの手順が含まれます。

運用 - ハイ アベイラビリティ Cisco IOS の実装の管理：Cisco IOS の運用のベスト プラクティスには、ソフトウェア バージョン管理、Cisco IOS Syslog 管理、問題管理、設定の標準化、およびアベイラビリティ管理が含まれます。

ベスト プラクティス	詳細（Outcall Billing Detail）
Cisco IOS の運用における戦略とツール	Cisco IOS の運用における最初の戦略は、環境をできるだけ簡素化し、設定と Cisco IOS のバージョンのばらつきをなくすことです。2 番目の戦略は、ネットワークの不具合を識別し、迅速に解決する能力です。
ソフトウェア バージョン管理	ソフトウェア バージョン管理は、標準化されたソフトウェア バージョンだけを実装して、ネットワークを監視するプロセスです。このプロセスでは、ソフトウェアを検証し、バージョンに準拠しないソフトウェアを変更する場合もあります。
予防的な Syslog 管理	Syslog の収集、監視、および分析は、その他の手段では識別が困難または不可能な、Cisco IOS に特有のより多くのネットワーク問題を解決するために推奨される、障害管理プロセスです。
問題管理	問題判別、情報収集、および十分に分析されたソリューション パスを定義する詳細な問題管理プロセス。このデータは、根本的な原因を判断するために使用できます。
設定の標準化	設定標準は、デバイスやサービスなどを通して標準のグローバル設定パラメータを作成して維持することで企業全体のグローバルな設定の整合性を確保する業務を表します。
アベイラビリティ管理	アベイラビリティ管理は品質向上メトリックとしてネットワークの可用性を使用して品質向上するプロセスです。

ソフトウェア ライフサイクル管理プロセスの概要

Cisco IOS ソフトウェア ライフサイクル管理は、信頼できるソフトウェア実装と高可用性ネットワーキングのために推奨される計画、設計、実装、および運用プロセスのセットとして定義されます。Cisco IOS ソフトウェアのライフサイクル管理には、ネットワーク内の Cisco IOS バージョンを選択、検証、および保守するプロセスが含まれます。

Cisco IOS ソフトウェアのライフサイクル管理の目標は、実稼動下でソフトウェアの不具合またはソフトウェア関連の変更やアップグレードの障害が確認される可能性を低減し、ネットワーク アベイラビリティを高めることです。この文書内で定義するベスト プラクティスは、シスコの多くのお客様および Cisco アドバンスト サービス チームの実際の経験に基づいて、同様の不具合および変更障害を減らすために公開されてきました。ソフトウェア ライフサイクル管理を導入すると、初期費用が増加する可能性があります。ただし、停止回数がより少なく、配備とサポートのメカニズムがより簡素化されているために、全体的な所有コストは減少します。

計画 - Cisco IOS の管理フレームワークの構築

計画は、Cisco IOS の管理における最初のフェーズであり、ソフトウェア アップグレードの時期、アップグレードする場所、およびイメージのテストと検証に使用するプロセスを判断するために必要とされます。

ベスト プラクティスには、ソフトウェア バージョン トラックの定義、アップグレード サイクルとその定義、および内部ソフトウェア認証プロセスの作成が含まれます。

Cisco IOS の計画における戦略とツール

現在の業務についての率直な評価、達成可能な目標の策定、およびプロジェクトの計画を行うことによって Cisco IOS の管理計画を開始します。この文書内のベスト プラクティスとお客様の組織内のプロセスを比較して、自己評価を実行する必要があります。基本的な質問には以下が含まれます。

- 組織で、ソフトウェア テスト/検証を含むソフトウェア認証プロセスが使用されているか。
- 組織で、 ネットワーク上で稼働する Cisco IOS バージョンの数が制限されるような Cisco IOS ソフトウェア標準が使用されているか。
- 組織で、Cisco IOS ソフトウェアのアップグレード時期を決定するのが困難か。
- 組織で、新しい Cisco IOS ソフトウェアを効率的かつ効果的に導入するのが困難か。
- 組織で、導入後にダウンタイムのコストに深刻な影響を与える Cisco IOS の安定性の問題が発生していないか。

組織での評価を終えたら、次は Cisco IOS ソフトウェア管理の目標を定義する必要があります。技術、実装、および運用のそれぞれのアーキテクチャを計画するグループから、職能上の枠を越えてマネージャまたはリーダーのどちらか（あるいはその両方）を集めて、Cisco IOS の目標およびプロセス改善プロジェクトの定義を開始します。最初の会議の目的は、全体的な目標、役割、および責任を決定し、アクション項目を割り当て、プロジェクトの初期スケジュールを定義することです。また、ソフトウェア管理のメリットを判定するための重要成功要因とメトリックも定義します。可能性のあるメトリックには、次のようなものがあります。

- ソフトウェアの問題に起因するアベイラビリティ
- ソフトウェアのアップグレードにかかるコスト
- アップグレードに必要な時間
- 実働環境で稼動しているソフトウェア バージョンの数
- ソフトウェアのアップグレード変更における成功と失敗の比率

Cisco IOS 管理フレームワーク全体の計画に加えて、組織によっては、毎月または四半期ごとの定期ソフトウェア計画会議も定義します。これらの会議の目的は、現状のソフトウェア配備を確認し、任意の新規ソフトウェア要件の計画を開始することです。計画には、現在のソフトウェア管理プロセスの再確認や変更が含まれる場合と、ソフトウェア管理の別のフェーズにおける役割および責任の定義だけが含まれる場合があります。

計画フェーズで使用するツールは、ソフトウェア インベントリ管理ツールだけで構成されています。この領域で使用する主要なツールは、CiscoWorks 2000 Resource Manager Essentials（RME）Inventory Manager です。CiscoWorks2000 RME Inventory Manager を使用すると、Web ベースのレポート ツールによって、Cisco ルータおよびスイッチのバージョン管理が大幅に簡素化されます。この Web ベースのレポート ツールでは、ソフトウェア バージョン、デバイスのプラットフォーム、メモリ サイズ、およびデバイス名に基づいて、Cisco IOS のデバイスが報告およびソートされます。

ソフトウェア バージョン トラックの定義

Cisco IOS ソフトウェア管理計画の最初のベスト プラクティスでは、ソフトウェアの一貫性をどこで維持できるかを特定します。ソフトウェア トラックは、個々の地理的分布、プラットフォーム、モジュール、または機能要件によって他の領域から区別される、ソフトウェア バージョンの一意のグループとして定義されます。理想的なのは、1 つのネットワークで 1 つのソフトウェア バージョンだけが実行されている状態です。その場合、ソフトウェア管理に関連するコストが大幅に削減され、一貫性があり、簡単に管理できる環境が実現できます。ただし、ほとんどの組織では、特定の領域内の機能、プラットフォーム、移行、および可用性の問題のためにネットワーク上で複数のバージョンを実行しなければならないのが現実です。異種プラットフォーム上で同じバージョンが実行されることはあまりありません。また、1 つのバージョンが組織のすべての要件をサポートするようになるまで待っているわけにはいかないということもあります。テストと検証、認証、およびアップグレード要件を検討し、そのネットワークで使用するソフトウェア トラック数をできるだけ減らすことが目標です。多くの場合、組織は、テスト/検証、認証、およびアップグレードの全体コストを抑えるために少し多めのトラックを使用しています。

各ソフトウェア トラックを区別する最初の基準は、プラットフォームのサポートです。通常、LAN スイッチ、WAN スイッチ、コア ルータ、およびエッジ ルータには、それぞれ個別のソフトウェア トラックがあります。また、特にこの要件がネットワーク内でローカライズ可能な場合には、data-link switching（DLSw; データリンク スイッチング）、Quality of Service（QoS）、または IP テレフォニーなどの特定の機能またはサービスのために、他のソフトウェア トラックが必要になる場合もあります。

もう 1 つの基準は信頼性です。多くの組織では、ネットワーク エッジに対してはより新しく高度な機能やハードウェア サポートを提供しますが、ネットワーク コアおよびデータ センターに対しては最も信頼できるソフトウェアを使用するよう努めます。その一方で、多くの場合、コアやデータ センターは、スケーラビリティや帯域幅の機能を最も必要とする環境です。異なる WAN ルータ プラットフォームを持つ大規模なディストリビューション サイトなど、特定のプラットフォーム用に他のトラックが必要になる場合もあります。次の表は、大規模な企業組織用のソフトウェア トラック定義の例です。

トラック	領域	ハードウェア プラットフォーム	機能	Cisco IOS のバージョン	サーティフィケーション ステータス
1	LAN コア スイッチング	6500	QoS	12.1 E(A 8)	Testing
0	LAN アクセス スイッチ	2924 XL 2948	Unidirectional Link Detection Protocol（UDL	12.0 (5.2)XU	認定 3/1/01

		XL	D; 単方向リンク検出プロトコル）、Spanning Tree Protocol（STP; スパニング ツリー プロトコル）		
3	LAN ディストリビューション/アクセス	5500 6509	Supervisor 3	5.4(4)	認定 7/1/01
4	ディストリビューション スイッチ ルート スイッチ モジュール（RSM）	RSM	Open Shortest Path First（OSPF）ルーティング	12.0(11)	認定 3/4/02
5	WAN ヘッドエンド ディストリビューション	7505 7507 7204 7206	OSPF フレームリレー	12.0(11)	2001 年 11 月 1 日済み
6	WAN アクセス	2600	OSPF フレームリレー	12.1(8)	認定 6/1/01
7	IBM との接続	3600	Synchronous Data Link Control（SDLC; 同期データリンク制御）ヘッドエンド	11.3(8)T1	認定 11/1/00

トラックの割り当ても、時間の経過とともに変化します。多くの場合、機能やハードウェアのサポートはメインラインのソフトウェア バージョンに統合され、最終的には異なるトラックを一緒に移行できるようになります。トラックの定義が終了すると、定義されている他のプロセスを使用して、新しいバージョンの一貫性や検証の段階へと進むことができます。トラック定義もまた、継続した取り組みです。新しい機能、サービス、ハードウェア、またはモジュールの要件が生じたら、そのつど新しいトラックを検討する必要があります。

トラックのプロセスを開始する際には、トラック要件を新たに定義することから始めるか、既存のネットワークに対する安定化のプロジェクトから始める必要があります。既存ソフトウェア バージョンとの間に特定できる共通点があるために、現在のトラック定義が使用できる場合もあります。ほとんどの場合、お客様のネットワークに十分な安定性が備わっている場合は、特定されたバージョンに急いで移行する必要はありません。通常は、ネットワーク アーキテクチャ グループ、または技術グループが、トラック定義プロセスの責任を負います。場合によっては、1 人の担当者がトラック定義の責任を負っていることもあります。また、それぞれのプロジェクトごとに、プロジェクト リーダーがソフトウェア要件および新しいトラック定義の策定に対する責任を負う場合もあります。新しいトラックが必要かどうかや、従来のトラックを統合またはアップグレードする必要があるかどうかを判断するために、四半期ごとにトラック定義を確認するのもよい方法です。

厳密なバージョン管理によってソフトウェア トラックの確認および保守を行っている組織は、ネ

ットワーク内で稼動するソフトウェア バージョン数を削減することに成功しています。その結果、一般的に、ソフトウェアの安定性が向上し、ネットワークの全体的な信頼性が実現します。

アップグレード サイクルとその定義

アップグレード サイクルの定義は、ソフトウェアおよび変更管理における基本的な品質手順として定義され、ソフトウェアのアップグレード サイクルの開始時期を判断するために使用されます。アップグレード サイクル定義を使用すると、ソフトウェアのアップグレード サイクルを適切に計画し、必要なリソースを割り当てることができます。アップグレード サイクル定義を使用しないと、多くの場合、現在の安定したバージョンに対して機能要件が発生することによって、ソフトウェアの信頼性に関する問題が増加します。また、実稼動環境で使用する前に、新しいバージョンを適切にテストおよび検証する機会を逸することにもなりかねません。

この作業で重要なのは、ソフトウェア計画プロセスを開始する時期と、その程度を特定することです。なぜならば、必要な注意を払わずに実稼動環境で機能、サービス、またはハードウェア機能を有効にしたり、ソフトウェア管理について検討せずに Cisco IOS を新しいバージョンにアップグレードしたりすることが、ソフトウェアに関する問題の主な原因となっているからです。もう 1 つの問題は、アップグレードを行わないことです。正常なソフトウェア サイクルおよび要件を無視し、複数の異なるメジャー リリースを飛ばしてソフトウェアをアップグレードすることにより、問題に直面するお客様も多くおられます。このような問題の原因は、イメージ サイズ、デフォルト動作の変更、Command Level Interpreter（CLI; コマンド レベル インタープリタ）の変更、およびプロトコルの変更です。

新しい主要機能、サービス、またはハードウェア サポートが必要になった場合には、この文書で定義するベスト プラクティスに基づいて明確に定義したアップグレード サイクルを開始することをお勧めします。正確なテスト要件と検証要件を判断するために、認証およびテストと検証の程度を（リスクに基づいて）分析する必要があります。地理的な場所、論理的な場所（コア、ディストリビューション、またはアクセス レイヤ）、または影響されることが予想される人数やお客様の数によって、リスク分析を実行できます。主要な機能またはハードウェア機能が最新のリリースに含まれている場合、何らかの簡素化されたアップグレード サイクル プロセスも開始する必要があります。機能が比較的マイナーであれば、リスクを検討し、どのプロセスを開始する必要があるのかを判断します。また、組織を比較的最新の状態に保ち、アップグレード プロセスが非常に面倒なものになるのを避けるためには、ソフトウェアを 2 年以内にアップグレードする必要があります。

End Of Life（EOL; 廃止）ステータスを過ぎたソフトウェア トレインに対しては、不具合の修正が行われないという事実も考慮する必要があります。多くの環境では、テストと検証のプロセスをほとんどまたはまったく行わずに新しく機能を追加し、その結果ダウンタイムが発生しても、容認されたり、場合によっては歓迎されたりしています。そのため、ビジネス要件に関しても考慮する必要があります。テスト要件を検討する際には、Cisco のリリース工程において収集された、最新のデータも検討する必要があります。不具合とその根本原因の分析の示すところによれば、不具合の根本原因の大半は、影響を受けているソフトウェア領域内のコーディングにあるものです。つまり、特定の機能またはモジュールをネットワーク内の既存リリースに追加する場合、その機能またはモジュールに関連した不具合が発生する可能性はありますが、新しい機能、ハードウェア、またはモジュールが他の領域に影響を与える可能性は非常に低いということです。このデータに従えば、既存のリリースでサポートされている新しい機能またはモジュールを追加する場合、新しいサービスまたは機能だけを他の有効なサービスと組み合せてテストすることで、テスト要件を低くすることが可能になります。ネットワーク内でいくつかの深刻な不具合が検出されたためにソフトウェアをアップグレードする場合にも、このデータを検討する必要があります。

次の表に、ハイ アベイラビリティを実現している主な企業組織における推奨アップグレード要件

を示します。

ソフトウェア管理におけるトリガ	ソフトウェアのライフサイクル要件
新しいネットワーク サービス。たとえば、新しい ATM バックボーンまたは新しい VPN サービス。	新機能の（他の有効なサービスとの関連での）テスト、コラプスト トポロジ テスト、what-if パフォーマンス分析、およびアプリケーション プロファイル テストからなる、ソフトウェア ライフサイクルの全面的な検証を行います。
新しいネットワーク機能が、現在使用しているソフトウェア リリースでサポートされていない。たとえば、QoS と Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）などの例があります。	新機能の（他の有効なサービスとの関連での）テスト、コラプスト トポロジ テスト、what-if パフォーマンス分析、およびアプリケーション プロファイル テストからなる、ソフトウェア ライフサイクルの全面的な検証を行います。
現在のリリースに追加される新しい主要機能またはハードウェア モジュール。たとえば、新しい GigE モジュール、マルチキャスト サポート、または DLSW の追加。	候補管理のプロセス。リリース要件に基づいた、完全な検証。または、候補管理によって現在のリリースが使用可能だと確認された場合には、限定的なテストと検証。
マイナーな機能追加。たとえば、アクセス制御用の TACACS デバイス。	機能のリスクに基づいて、候補管理を検討します。リスクに基づいて、新機能のテストまたは試験導入を行います。
2 年間ソフトウェアが実稼動している場合、または四半期ごとのソフトウェアの確認。	現在サポート可能なリリースを特定するための全面的なライフサイクルの管理に関連した候補管理およびビジネス決定。

緊急アップグレード

致命的な不具合が原因で、ソフトウェアのアップグレードが必要になる場合があります。このような場合、緊急アップグレードの方法論が用意されていないと、問題の発生につながる可能性もあります。ソフトウェア アップグレードが管理されておらず、ライフサイクル管理を行わずにソ

フトウェアをアップグレードしてしまうような場合から、ネットワーク デバイスがクラッシュし続けているにもかかわらず、候補となっているリリースの認証とテストが完了してないためにアップグレードが行われない状況まで、ソフトウェアに関する問題は多岐に渡ります。このような場合には、シスコは緊急アップグレード プロセスをお勧めします。緊急アップグレード プロセスでは、ネットワーク内の、ビジネスにおける重要性がそれほど高くない領域では、限定されたテストおよび試験導入だけが実行されます。

明確な回避策の用意されていない致命的なエラーが発生した際に、その問題がソフトウェアの不具合に関連している場合は、不具合を分離し、修正が使用可能なのか、またはいつ使用可能になるのかを判断するために、シスコのサポートを最大限利用されるようお勧めします。修正が使用可能な場合には、限られたダウンタイム内で問題を修正できるかどうかを迅速に判断するために、緊急アップグレード サイクルの適用をお勧めします。ほとんどの組織ではサポートされたバージョンのコードを実行しているため、そのソフトウェアの新しい暫定バージョンに上げることで問題を修正できます。

潜在的な緊急アップグレードを、お客様の組織で準備することもできます。準備には、サポートされている Cisco IOS リリースへの移行と、認証されているバージョンと同じ Cisco IOS トレインに含まれる、候補を差し替えるためのバージョンの特定または開発が含まれます。ソフトウェアがサポートされているということは重要です。それは、シスコの開発が、そのソフトウェア トレインへの不具合修正を継続していることを意味するためです。サポートされているソフトウェアをネットワーク内で維持することによって、より使い慣れた、安定したコード ベースにより、検証時間を短縮できます。通常、候補が差し替えられるのは、同じ Cisco IOS トレインに含まれる、機能やハードウェア サポートが追加されていない新しい暫定ソフトウェア イメージです。候補を差し替える戦略は、特定のソフトウェア トレインを採用するための、初期フェーズにある組織にとっては特に重要です。

認証プロセス

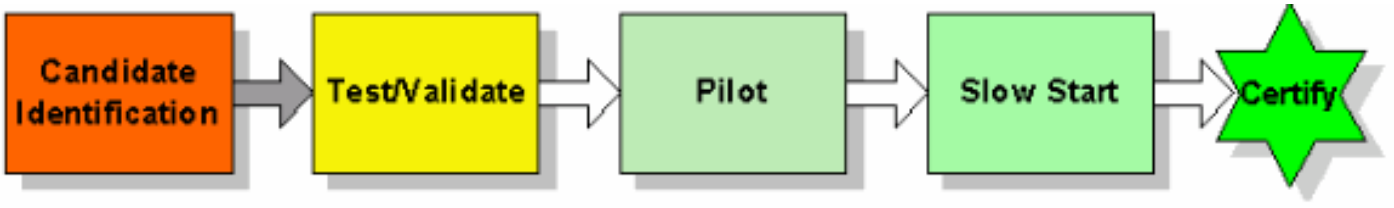
認証プロセスは、検証されたソフトウェアが、実稼動環境内に一貫性をもって配備されていることを確認するために役立ちます。認証プロセスの手順には、トラックの特定、アップグレード サイクルの定義、候補管理、テストと検証、およびある程度の試験的な実稼動使用を含める必要があります。しかし、簡単な認証プロセスであっても、特定のトラック内に一貫したソフトウェア バージョンが配備されていることの確認には役立ちます。

認証プロセスを開始するには、アーキテクチャ、技術または配備、および運用の各グループから、認証プロセスの設計管理を行う担当者を集めます。このグループはまず、認証プロセスの継続的な成功を実現するために、ビジネス目標およびリソースの能力を検討する必要があります。次に、担当者またはグループに対して、認証プロセス内の主要手順に関する全体的な責任を割り当てます。これらの手順には、トラック管理、ライフサイクルのアップグレード定義、テストと検証、および試験的導入が含まれます。これらの各領域は、組織内において定義され、承認されて、正式に連絡される必要があります。

また、認証プロセスの各フェーズにおける品質または承認のためのガイドラインも必要です。これは、品質ゲート プロセスと呼ばれることがあります。プロセスが次の手順に進む前に、特定の品質基準を満たす必要があるためです。このプロセスは、認証プロセスが効果的であり、リソースを割り当てる価値があること確認するために役立ちます。一般的に、ある領域で品質に問題が発見されると、プロセスは 1 つ前の手順に戻されます。

ソフトウェアの品質または予想外の動作により、候補であるソフトウェアが、定義されている認証基準を満たさない場合があります。環境に影響を与える問題が発見された場合には、今後の暫定リリースを認証するための、より簡素化されたプロセスを用意する必要があります。これは、変更された内容や解決された不具合を組織が理解できる場合には、リソース要件の軽減に役立ち

、一般的に効果があります。最初の候補に問題が発生したために、その後の暫定 Cisco IOS リリースを認証することは珍しくありません。また、問題があったために限定的な認証を行うか注意事項を発行しておいて、その後、新しい暫定リリースが検証された際に、完全に認証されたリリースにアップグレードする場合もあります。次のフローチャートは、基本的な認証プロセスであり、品質ゲート（各ブロックに続く確認）が含まれています。



設計 - Cisco IOS バージョンの選択と検証

Cisco IOS のバージョンを選択および検証するための明確な方法論を用意することにより、アップグレードの失敗やソフトウェアの予想外の不具合に起因する、想定外のダウンタイムを削減できます。

設計フェーズには、候補管理およびテストと検証が含まれます。候補管理は、定義されているソフトウェア トラックの特定のバージョンを識別するためのプロセスです。テストと検証は、認証プロセスの一部であり、それらのソフトウェア バージョンが、必要なトラック内で正常に動作することを確認します。テストと検証は、コラプスト トポロジのラボ環境、および実稼働環境に類似した設定のラボ環境で実行する必要があります。

Cisco IOS の選択および検証における戦略とツール

各組織には、Cisco IOS のバージョンを選択するプロセスをはじめとして、ネットワークで使用する Cisco IOS の標準のバージョンを選択して検証するためのプロセスがあります。アーキテクチャ、Engineering と動作の横断のチームは候補管理プロセスを定義し、記録する必要があります。承認されたプロセスは、適切な実行グループに引き渡す必要があります。また、候補の情報が確認されたら更新できる、標準的な候補管理のテンプレートを作成することも推奨されます。

実稼動環境を容易に模擬できるような高度なラボ環境を、すべての組織が用意できるわけではありません。組織によっては、ビジネスに大きな影響を与えずに、ネットワーク内で新しいバージョンを試験するための費用や能力が不足しているために、ラボ テストを行っていません。ただし、ハイ アベイラビリティを実現するためには、実稼動ネットワークを模擬したラボを構築することが推奨されます。また、Cisco IOS の新しいバージョンが広い範囲でテストされるように、テストと検証のプロセスを開発することも推奨されます。ラボの構築には、約 6 か月を割り当てる必要があります。この期間に、ラボを最大限に活用するための、具体的なテスト計画およびプロセスを作成します。Cisco IOS の場合、それぞれの必要なソフトウェア トラックに対する Cisco IOS の具体的なテスト計画を作成します。新製品やソフトウェアの導入を目的としたラボはあまり使用されていないため、これらのプロセスは、より大規模な組織で重要になります。

次のセクションでは、Cisco IOS の選択および検証に使用する、候補管理のツール、およびテストと検証のツールについて簡単に説明します。

候補管理のツール

注：以下に示すツールのほとんどを使用するには、登録ユーザであり、ログインしている必要があります。

- リリース ノート：そのリリースのハードウェア、モジュール、および機能のサポートに関する情報があります。採用する可能性のあるリリースに、必要なハードウェアとソフトウェアのサポートがすべて存在することを確認したり、デフォルト動作の違いやアップグレード要件などの移行に関する問題を理解するために、候補管理の作業中にリリース ノートを確認する必要があります。

テストと検証のツール

テストと検証のツールを使用して、新しいハードウェア、ソフトウェア、およびアプリケーションなどのネットワーク ソリューションをテストおよび検証します。

- **トラフィック ジェネレータ：特定のプロトコルを使用する任意のリンク全体のレートをモデル化するために使用される、マルチプロトコル トラフィック ストリームおよび raw パケット レートを生成します。**ユーザは、送信元、宛先 MAC、およびソケット番号を指定できます。これらの値は、指定した手順で増加するか、スタティック（固定）、またはランダムに増加するように設定できます。トラフィック ジェネレータでは、次のプロトコルのパケットが生成できます。IPInternetwork Packet Exchange（IPX）DECnetAppleXerox Network Systems（XNS）Internet Control Message Protocol（ICMP; インターネット制御メッセージ プロトコル）インターネット グループ管理プロトコル（IGMP）コネクションレス型ネットワーク サービス（CLNS）User Datagram Protocol（UDP; ユーザ データグラム プロトコル）Virtual Integrated Network Service（VINES）データリンク パケットツールは Agilent および Spirent Communications から入手できます。
- **パケット カウンタ/キャプチャ/デコーダ（スニファ）：すべてのパケットおよびデータリンク レイヤでパケットを選択的にキャプチャし、デコードできます。**このツールには、ユーザがフィルタを指定できる機能が用意されているため、指定したプロトコルのデータだけをキャプチャすることができます。フィルタを使用すると、特定の IP アドレス、ポート番号、または MAC アドレスに一致するパケットをキャプチャするように指定できます。ツールは、Sniffer Technologies から入手可能です。
- **ネットワーク シミュレータ/エミュレータ：実稼動ネットワーク要件に基づいて、特定のルータのルーティング テーブルにデータを入力できます。**IP の Routing Information Protocol（RIP; ルーティング情報プロトコル）、OSPF、Intermediate System-to-Intermediate System（IS-IS）、Interior Gateway Routing Protocol（IGRP）、Enhanced IGRP（EIGRP）、および Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）のルータの生成がサポートされます。ツールは、PacketStorm Communications および Spirent Communications から入手可能です。
- **セッション エミュレータ：スライディング ウィンドウによるマルチプロトコル トラフィック ストリームを生成し、受信デバイスに向けて、テスト ネットワーク全体にマルチプロトコル トラフィック ストリームを送信できます。**受信デバイスは、送信元に向けて、パケットをエコーバックします。送信されたパケット、受信されたパケット、シーケンス違反のパケット、およびエラー パケットの数が、送信元デバイスで検証されます。このツールでは Transmission Control Protocol（TCP; 伝送制御プロトコル）のウィンドウ パラメータを定義する柔軟性も提供されるため、クライアント/サーバ間のトラフィック セッションが、ラボ ネットワーク内で厳密に模倣できます。ツールは、Empirix から入手可能です。
- **大規模ネットワーク エミュレータ：大規模環境におけるスケーラビリティのテストに役立ちます。**これらのツールでは、より厳密に実稼働環境を模倣するために、管理タイプのトラフィックを作成し、ラボ トポロジに容易に取り込むことができます。ルート インジェクタ、プロトコル ネイバー、およびレイヤ 2 プロトコル ネイバーなどの機能が含まれます。ツールは Agilent および Spirent Communications から入手できます。
- **WAN シミュレータ：帯域幅および遅延が問題となることのある企業のアプリケーション ト**

ラフィックのテストに適しています。これらのツールでは、遅延および帯域幅の予想されるアプリケーションをローカルでテストし、WAN 上でアプリケーションがどう機能するかを確認できます。多くの場合、これらのツールは、企業組織内でのアプリケーション開発およびアプリケーション プロファイルのテストタイプとして使用されます。Spirent CommunicationsとShunraの部門であるAdtechは 、WANシミュレーションツールを提供しています。

候補管理

候補管理は、特定のハードウェアおよび有効な機能セットに対する、ソフトウェア バージョン要件および潜在的なリスクを特定するプロセスです。ソフトウェア要件、リリース ノート、ソフトウェアの不具合、および潜在的なリスクを適切に調査するために、リリースの試験導入前に 4 ~ 8 時間を費やすことをお勧めします。候補管理の基本的な概要は、次のとおりです。

- Cisco Connection Online（CCO）ツールによる、候補ソフトウェアの特定。
- ソフトウェアの完成度についてのリスク分析、新機能、またはコード サポート。
- ライフサイクル全体における既知のソフトウェアの不具合、問題、および要件の特定と追跡。
- 選択したイメージのデフォルトの設定動作の確認。
- maintain キャンセルされ、候補の繰上げ候補が変わります。
- バグ スクラブ。
- シスコ アドバンスト サービスによるサポート。

候補ソフトウェアの特定は、Cisco 製品およびソフトウェア トレインの数の増加とともに、より複雑になりました。現在、CCO では、Cisco IOS Upgrade Planner、ソフトウェア検索ツール、ソフトウェアとハードウェアの互換性マトリックス、および製品アップグレード ツールなど、リリース候補の特定に役立ついくつかのツールが用意されています。これらのツールは、http://www.cisco.com/public/sw-center/ で入手できます。

次に、候補ソフトウェアのリスクを分析します。これは、ソフトウェアが現在、完成度曲線のどの部分に位置しているのかを理解し、リリース候補の持つ潜在的なリスクと配備の要件を評価するプロセスです。たとえば、Early Deployment（ED; 初期導入）ソフトウェアをクリティカルなハイ アベイラビリティ環境に配備する場合、認証を成功させるためには、関連リスクおよびリソース要件を検討する必要があります。リスクのより高い状況で成功するためには、少なくともソフトウェア管理リソースを追加する必要があります。逆に、組織のニーズを満たす General Deployment（GD; 一般導入）バージョンが利用できる場合には、必要なソフトウェア管理リソースは少なくなります。

リリース候補と潜在的なリスクを特定した後は、認証を妨げる可能性のある致命的な不具合が存在しているかどうかを判断するために、バグスクラブを実行します。シスコの Bug Watcher、Bug Navigator、および Bug Watcher Agents を使用すると、潜在的な問題を特定するために役立ちます。潜在的なセキュリティまたは不具合の問題を特定するには、ソフトウェア ライフサイクル全体で、これらのツールを使用する必要があります。

新しいソフトウェアの候補は、潜在的なデフォルト設定処理に確認する必要があります。デフォルトの設定動作の確認は、新しいソフトウェア イメージのリリース ノートを確認し、指定したプラットフォーム上にロードされたイメージとの設定の違いを確認することによって行えます。候補管理では、選択したバージョンがプロセス内のどこかのポイントで認証基準を満たさない場合、その前後のバージョンが検討されることもあります。指定したトラックの機能に関連するバグを監視することによって、認証の候補となるソフトウェアを管理できます。

シスコ アドバンスト サービスもまた、候補管理のために非常に役立ちます。このグループを通し

て、さまざまな垂直市場環境における、数多くの業界エキスパート達による開発プロセスやコラボレーションをより深く理解することができます。通常、Cisco サポートでは、優れたバグスクラブや候補管理の方法を提供しています。これは、専門家のレベルが高く、さまざまな組織で稼動しているソフトウェア バージョンに対する知識が豊富であるためです。

テストと検証

テストと検証は、管理のベスト プラクティスおよびハイ アベイラビリティ ネットワーキングの全体における重要な側面です。適切なラボ テストを実行すると、生産のダウンタイムが大幅に削減され、ネットワーク サポート担当者のトレーニングに役立ち、ネットワーク実装プロセスの簡素化にも役立ちます。ただし、効果的にテストを行うためには、適切なラボ環境を構築して維持し、正しいテストの実行に必要なリソースを確保し、推奨されるテスト方法（測定データの収集も含む）を使用するために、それぞれに必要なリソースを割り当てる必要があります。これらのリソースが割り当てられていない場合、テストと検証のプロセスは、期待通りの結果にならない可能性があります。

ほとんどの企業に推奨されるテスト ラボ環境がありません。そのために、多くの組織がソリューションを正しく配備できず、ネットワークの変更に失敗し、ラボ環境があれば発見できたはずのソフトウェア関連の問題に悩まされています。この状態が許容される環境もあります。それは、高度なラボ環境を準備するためのコストが、ダウンタイムのコストを超えてしまう場合です。しかし、多くの組織ではダウンタイムは容認されません。そのような場合には、実稼動ネットワークの品質を改善するために、推奨されるテスト ラボ、テスト タイプ、およびテスト方法論の開発を強くお勧めします。

テスト ラボとその環境

ラボは、机、作業台、テスト装置、および機器キャビネットや棚を置くための十分な空間を持つ、隔離された場所である必要があります。大規模な組織で実稼動環境を模擬する場合、通常では、機器の棚が 4 ~ 10 個必要になります。テストを行っている環境を維持するために、物理的なセキュリティを設置することをお勧めします。物理的なセキュリティを設置すると、ハードウェアの借り入れ、トレーニング、またはリハーサルの実施など、そのラボにおける他の優先事項によってラボ テストが中断されることが回避できます。論理セキュリティでは、不正なルートがラボを終了してから実稼働ネットワークまたは望ましくないトラフィックを入力しないように推奨されます。論理的なセキュリティは、ルーティング フィルタ、およびラボ ゲートウェイ ルータ上の拡張アクセス リストによって実現できます。実稼働ネットワークへの接続は、実稼働環境のラボのソフトウェアおよびアクセスに役立ちます。

すべてのテスト計画において、ラボ トポロジが実稼動環境を模倣できる必要があります。ハードウェア、ネットワーク トポロジ、および機能設定を再現することをお勧めします。もちろん、実際のトポロジを再現することはほとんど不可能です。ただし、ネットワーク階層および実稼働デバイス間のインタラクションを再現することは可能です。特に、複数デバイス間におけるプロトコルや機能のインタラクションは重要です。テスト トポロジによっては、ソフトウェア テスト要件によって異なるものもあります。たとえば、WAN エッジで使用する Cisco IOS のテストでは、LAN タイプのデバイスやテストは必要なく、WAN エッジ ルータと WAN ディストリビューション ルータだけが必要になります。実稼動環境を複製はせずに、ソフトウェアの機能を模倣するのが重要な点です。場合によっては、プロトコル ネイバーの数やルーティング テーブルなどの大規模な動作も、ツールを使用して模擬することが可能です。

いくつかのテスト タイプでは、実稼働環境の模倣やテスト データの収集能力を改善するためにもツールが使用されます。実稼動の模倣に役立つツールには、トラフィック コレクタ、トラフィック ジェネレータ、および WAN シミュレーション デバイスなどがあります。SmartBits は、ネットワーク トラフィックを収集してリプレイすることが可能なデバイス、または大規模なトラフィ

ックを生成することが可能なデバイスの好例です。また、プロトコル アナライザなどの、データの収集に使用するデバイスが役立つ場合もあります。

ラボにはまた、ある程度の管理も必要です。多くの大規模な組織にラボ ネットワークを管理する役割を持つフルタイム lab マネージャがあります。他の組織では、既存のアーキテクチャ チームおよび技術チームが、ラボ検証を担当します。ラボの管理責任は、オーダーの試験装置と資産追跡のケーブル接続、物理的なスペース管理のほか、ラボのルールと方向を、ラボのスケジューリング、ラボのドキュメント、セットアップのトポロジの例、ラボ テストを実行し、で示された問題を管理する書き込み test plan 定義します。

テスト タイプ

実行可能なテストには、多くの異なるタイプがあります。多くの設定ですべてのテストを行える完全なテスト ラボを構築し、テスト計画を作成する前に、さまざまなタイプのテスト方法や、テストを行う目的を理解しておく必要があります。また、それらのテストの中に、シスコの技術、テクニカル マーケティング、またはカスタマー サポートの各部門が責任を負う必要がある、あるいは責任を負うことが可能なテストがあるのかどうかについても理解しておく必要があります。顧客のテスト計画は一般的により提供されたテスト タイプについて説明します。次の表は、さまざまなテスト タイプ、テストを実行する時期、および各テストに対する責任の所在を理解するために役立ちます。

通常、次に示すテストでは、組織特有の機能セット、トポロジ、およびアプリケーション構成を適切にテストすることが最も大切です。シスコは、完全な機能テストおよび回帰テストを行いますが、お客様の組織のアプリケーション プロファイルをテストすることはできません。各組織のアプリケーション プロファイルは、トポロジ、ハードウェア、および設定機能の固有の組み合わせで構成されているためです。実際のところ、機能、ハードウェア、モジュール、およびトポロジ置換をすべてテストすることは不可能です。また、シスコはサードパーティ機器との相互運用性のテストも行えません。お客様の環境内で使用されているハードウェア、モジュール、機能、およびトポロジの正確な組み合わせを、お客様ご自身でテストされることをお勧めします。このテストは、パフォーマンス、相互運用性、停止、および焼き込みなどのサポート テストタイプとともに、組織の実稼動環境を再現するコラプスト トポロジを備えたラボ内で実行する必要があります。

テスト	テストの概要	テストの責任者
機能および機能性	基本的な Cisco IOS の機能および Cisco ハードウェア モジュールが、公表されているとおりに機能するかどうかを判断します。機能またはモジュールの機能、および機能の設定は任意でテストする必要があります。設定の削除および追加はテストする必要があります。基本的な停止のテストおよびバーンイン テストが含まれます。	シスコ デバイス テスト チーム
リグレ	機能またはモジュールが他のモジュールおよび機能との組み合わせで動作するか	シスコ リグレッション テスト チーム

ッション	どうかを判断します。また、Cisco IOS の対象バージョンが、定義した機能に関して、他の Cisco IOS バージョンとの組み合わせで動作するかどうかを判断します。バーンインおよび停止テストが含まれます。	
デバイスの基本的なパフォーマンス	Cisco IOS の機能またはハードウェア モジュールが、負荷を受けたときに最低要件を満たすかどうかを判断するために、機能またはモジュールの基本的なパフォーマンスを測定します。	シスコ デバイス テスト チーム
トポロジ、機能、ハードウェアの組み合わせ	特定のトポロジとモジュール、機能、ハードウェアを組み合わせた場合に、機能およびモジュールが予想どおりに機能するかどうかを判断します。このテストには、プロトコルの検証、機能の検証、show コマンドの検証、焼き込みテスト、および停止テストを含める必要があります。	シスコは、Enterprise Solutions Engineering (ESE; エンタープライズ ソリューション エンジニアリング) および Networked Solutions Integration Test Engineering (NSITE; ネットワーク ソリューション統合テスト エンジニアリング) などのラボにおいて、公開されている標準のトポロジをテストします。 ハイアベイラビリティ環境のお客様は、必要に応じて、機能、モジュール、トポロジの組み合わせをテストしてください。特に、初期採用ソフトウェアおよび非標準トポロジの場合は、テストが必要です。
停止 (W	特定の機能、モジュール、トポロジによって構成される環境で発生する可能性がある一般的な停止タイプや	シスコは、基本的な停止のテストを行います。お客様の環境の拡張性に関する停止のパフ

hat-if）	停止動作、および、その結果生じる可能性のある機能性への影響を判断します。停止テストには、カードスワッピング、リンク フラップ、デバイス障害、リンク障害、およびカード障害が含まれます。	ォーマンスの問題に最終的な責任があります。停止テストは、可能な限り、お客様のラボ環境内で実行する必要があります。
ネットワークパフォーマンス（What-if）	特定の機能、ハードウェア、トポロジの組み合わせに関連したデバイス負荷を調査します。対象となるのは、設定されているトラフィック タイプとプロトコル、隣接デバイス、ルート数、およびその他の機能におけるリソース要件に関連した、CPU、メモリ、バッファ使用率、およびリンク使用率などの、デバイス容量およびパフォーマンスです。このテストは、大規模な環境におけるスケーラビリティを確認するのに役立ちます。	顧客はデバイスのロードと拡張性に優れた責任があります。負荷とスケーラビリティの考慮事項は、シスコの営業または Advanced Services で発生し、顧客の概念実証 Lab（CPOC）などのシスコ ラボでテストされます。
不具合修正	発見された不具合が、不具合修正によって修正されることを確認します。	シスコには、バグが修正するようにバグ修正をテストします。発生した不具合が修正されたことと、不具合が発生したモジュールまたは機能において、その不具合による他の側面への影響がないことを確認するために、お客様もテストを行う必要があります。メンテナンス リリースは回帰テスト済みですが、暫定リリースは通常、回帰テスト済みではありません。
ネットワーク管理	Simple Network Management Protocol（SNMP; 簡易ネットワーク管理プロトコル）管理機能、SNMP の MIB 変数の精度、トラップのサポート、および Syslog のサポートを調査	シスコは、基本的な SNMP 機能および MIB 変数の精度をテストする必要があります。お客様は、ネットワーク管理の結果を検証する必要があります。また、新しいテクノロジー

	します。	の展開における管理戦略および方法論に関しては、お客様に最終的な責任があります。
大規模ネットワークエミュレーション	大規模ネットワーク エミュレーションでは、大規模な環境をシミュレートするために、Agilent のルータ シミュレータや Spirent のテスト ツール スイートなどのツールが使用されます。エミュレーションには、プロトコルの隣接ルータ、フレームリレーによる Permanent Virtual Circuit (PVC; 相手先固定接続) の数、ルーティング テーブルのサイズ、キャッシュ エントリなどが含まれます。また、デフォルトではラボ内に存在しないが、実稼動環境では通常必要とされる、その他のリソースが含まれる場合もあります。	お客様は一般にネットワークシミュレーションのテストの側面を管理する製品におけるルーティング プロトコル ネイバーと隣接と関連するルーティングテーブルサイズなどのリソースを含む可能性があるネットワーク環境を再生します。
相互運用性	サードパーティ ネットワーク機器への接続性に関するすべての側面をテストします。特に、プロトコルまたはシグナリング相互運用性が必要な場合、このテストを行います。	お客様は通常、相互運用性テストのすべてを担当します。
通電	ルータ リソースを経時的に調査します。バーンイン テストは通常、デバイスのメモリ、CPU、およびバッファを含むリソース使用率に調査の口ードで、時間とともにあります。	シスコは、基本的なバーンイン テストを実行します。トポロジ、デバイス、および機能の固有の組み合わせに関しては、お客様によるテストが推奨されます。

テストの方法論

テストの内容が理解できたら、次に、テスト プロセスのための方法論に進む必要があります。ベスト プラクティスのテスト方法論の目的は、合意されたテストが包括的であり、適切に文書化され、容易に再現可能であり、実稼動において発生する可能性のある潜在的な問題の検出に有用であることの確認です。ラボ シナリオの文書化と再作成は、新しいバージョンをテストする場合や、ラボ環境で発見された不具合の修正をテストする場合に、特に重要です。テスト方法論の手順は、次のとおりです。いくつかのテスト手順は、同時に実行することも可能です。

1. テスト対象の実稼働環境をシミュレートするテスト トポロジを作成します。WAN エッジのテスト環境は LAN テストは推奨環境を表すことができるより多くのデバイスが含まれるこ

とがありますが、コア ルータだけと 1 のエッジ ルータを含めることができます。
2. 実稼働環境をシミュレートする機能を設定します。ラボ デバイスの設定は、想定する実稼動デバイスのハードウェアおよびソフトウェアの設定と厳密に一致する必要があります。
3. テスト計画を記述して、テストおよび目標を定義し、トポロジを文書化し、機能テストを定義します。テストには、基本的なプロトコルの検証、show コマンドの検証、停止テスト、および焼き込みテストを含める必要があります。あるテスト計画に含まれる特定のテストの例を、次の表に示します。
4. ルーティング機能およびプロトコル機能を検証します。ドキュメントまたはベースラインに必要な**show コマンドの結果。**プロトコルには、ATM、フレームリレー、Cisco Discovery Protocol（CDP）、イーサネット、スパニング ツリーなどのレイヤ 2 プロトコルと、IP、IPX、およびマルチキャストなどのレイヤ 3 プロトコルの両方を含める必要があります。
5. 機能の機能性を検証します。ドキュメントまたはベースラインに必要な**show コマンドの結果。**機能には、グローバル設定コマンドや、authentication, authorization, and accounting（AAA; 認証、認可、会計）などの任意の重要な機能が含まれる場合があります。
6. 実稼動環境内で予想される負荷をシミュレートします。負荷をシミュレートはトラフィックのデータ収集ツールやジェネレータを使用できます。すべてのパケット損失を調査し、CPU、メモリ、バッファ使用率、およびインターフェイス統計情報などの、予想されるネットワーク デバイス使用率（可変）を検証します。ドキュメントまたはベースラインに必要な**show コマンドの結果。**
7. 負荷を受けているデバイスおよびソフトウェアの、処理または回避が予想される状況での停止テストを実行します。たとえば、カードの取り外し、リンク フラッピング、ルータ フラッピング、およびブロードキャスト ストームなどがあります。正しい SNMP トラップがネットワークで使用される機能に基づいて生成されていることを確認します。
8. テストは再現可能である必要があるため、テスト結果およびデバイスの測定データを文書化します。

テスト名	**ホットスタンバイ ルータ プロトコル（HSRP）フェールオーバー**
テストの設定要件	プライマリ ゲートウェイのインターフェイスに負荷をかけます。ユーザ ステーション側からゲートウェイに向かうトラフィックを約 20 % にして、ユーザ ステーション側に向かう着信トラフィックを約 60 % にする必要があります。また、トラフィックを増加して、より大きな負荷をかけます。
テスト手順	show コマンドで STP、**HSRP をモニタします。**プライマリ ゲートウェイのインターフェイス接続を失敗させて、情報を収集してから接続を復旧します。
予想される測定データ	フェールオーバー中の CPU。フェールオーバーの前、途中、および後の、プライマリ ゲートウェイおよびセカンダリ ゲートウェイのインターフェイス。フェールオーバーの前、途中、および後の HSRP。
期待される結果	プライマリ ゲートウェイは、2 秒以内に他のルータ ゲートウェイにフェールオーバーします。**show コマンドの結果に、**

	変更が正しく反映されます。接続が復帰すると、プライマリ ゲートウェイへのフェールオーバーが発生します。
実際の結果	
Pass または Fail	
パスを処理するために必要な変更	

デバイス測定

デバイスが適切に動作していることを確認するために、テスト フェーズ中に次の測定を実行して文書化します。

- Memory usage
- CPU load
- バッファ使用率
- インターフェイスの統計情報
- ルート テーブル
- 特定のデバッグ

測定する情報は、実施しているテストによってそれぞれ異なります。そのテストで取り組んでいる問題によっては、これ以外にも測定の必要な情報が存在する場合もあります。

テスト対象の各アプリケーションで、そのアプリケーションのパフォーマンスに対する悪影響がないことを確認するために、パラメータを測定します。この測定は、パフォーマンスのベースラインを使用して、配備前と配備後のパフォーマンスを比較して行います。アプリケーション測定テストには、たとえば次のものが含まれます。

- ネットワークにログインするまでの平均時間。
- ファイルのグループを Network File System（NFS; ネットワーク ファイル システム）にコピーするために必要な平均時間。
- アプリケーションを起動し、最初の画面でプロンプトを表示するまでの平均時間。
- その他のアプリケーション固有のパラメータ。

実装 - 迅速で正常な Cisco IOS の導入

明確な実装プロセスを使用して、Cisco IOS の新しいバージョンを効果的に配備できます。

実装フェーズには、試験プロセスおよび実装プロセスが含まれます。試験プロセスでは、そのバージョンの Cisco IOS が実装先の環境で正常に動作することを確認し、実装プロセスでは、大規模な Cisco IOS の配備を迅速に、失敗なく行えます。

Cisco IOS の配備における戦略とツール

Cisco IOS の配備における戦略は、アップグレード ツールと明確な実装プロセスを使用して、試験プロセスおよび迅速な配備を通して、最終的な認証を行うことです。

ネットワークの試験プロセスを開始する前に、多くの組織では、一般的な試験ガイドラインを作

成します。試験ガイドラインには、成功の基準、試験場所の条件、試験の文書化、試験所有者の条件、ユーザ通知の要件、および予想される試験期間など、すべての試験で想定される内容が含まれている必要があります。通常は、技術、実装、および運用の各グループから集められた職能上の枠を越えて活躍するチームが、全般的な試験ガイドラインおよび試験プロセスの構築に携わります。試験プロセスが作成されると、個々の実装グループが、指定されたベスト プラクティス方式を使用して正しく試験を実施できるようになります。

新しいソフトウェア バージョンの配備および最終認証が承認されると、次に、Cisco IOS のアップグレード計画の開始が必要になります。アップグレードの計画は、プラットフォーム、メモリ、フラッシュ、および設定などの、新しいソフトウェア イメージの要件を特定することから開始します。通常、新しいソフトウェア イメージの要件は、Cisco IOS の管理ライフサイクルの候補管理フェーズに、アーキテクチャ グループと技術グループによって定義されます。必要な要件が特定されたら、実装グループが各デバイスを検証し、必要に応じてそれらのデバイスをアップグレードします。CiscoWorks2000 Software Image Manager（SWIM）モジュールを使用して、デバイス インベントリに対する Cisco IOS の要件を検証し、検証の手順を実行することもできます。すべてのデバイスが検証された場合か、ソフトウェア イメージが適切に新しい標準にアップグレードされた場合（あるいは、その両方が実行された場合）に、実装グループは、CiscoWorks2000 SWIM モジュールをソフトウェア配備のツールとして使用して、スロースタートの実装プロセスを開始できます。

新しいイメージを何度も正しく配備した後は、CiscoWorks SWIM を使用して、迅速に配備を行えるようになります。

Cisco IOS のインベントリ管理

CiscoWorks2000 Resource Manager Essentials（RME） のインベントリ マネージャを使用すると、Web ベースのレポート ツールによって、Cisco ルータおよびスイッチのバージョン管理が大幅に簡素化されます。この Web ベースのレポート ツールでは、ソフトウェア バージョン、デバイスのプラットフォーム、およびデバイス名に基づいて、Cisco IOS のデバイスが報告およびソートされます。

Cisco IOS SWIM

CiscoWorks2000 SWIM は、アップグレード プロセスが複雑なために エラーが発生しやすい場合に、プロセスの複雑さを軽減するために役立ちます。CCO へのリンクが埋め込まれているために、ネットワーク内に配備されている Cisco IOS および Catalyst ソフトウェアに対して、ソフトウェア パッチに関するシスコのオンライン情報が関連付けられ、関連するテクニカル ノートが強調表示されます。目的のソフトウェア イメージ アップデートをサポートするためにハードウェアのアップグレード（Boot ROM、Flash RAM）が必要な場合、新しい計画ツールによってシステム要件が検出され、通知が送信されます。

更新が開始される前に、新しいイメージの前提条件は、ターゲット スイッチまたはルータのインベントリ データに対して正常なアップグレードを確実に検証されます。複数のデバイスがアップデートされる場合、SWIM によってダウンロード タスクの同期が取られ、ユーザは作業の進行を監視できます。予定された作業は承認プロセスを通して管理され、各アップグレード タスクの開始前にマネージャによって技術者の作業が承認されます。RME 3.3 には Cisco IGX、BPX、および MGX プラットフォームのソフトウェア アップグレードを分析する機能が含まれているため、ソフトウェア アップグレードの影響を判断するために必要な時間が大幅に簡素化され、短縮されます。

試験プロセス

障害の発生する危険を最小限に抑え、実稼動における未解決の問題をより確実に把握するために、ソフトウェアの試験的導入を行うことをお勧めします。新しいテクノロジーを配備する際は一般的に試験が重要ですが、配備するソフトウェアが新しいサービス、機能、またはハードウェアにリンクされる場合には、よりクリティカルな試験が必要とされます。個々の試験計画ごとに、試験の選択、試験の期間、および測定方法を検討する必要があります。試験の選択とは、試験を実行する時期と場所を特定するプロセスです。試験の測定とは、試験の成功および失敗、または潜在的な問題を特定するために、必要なデータを収集するプロセスです。

試験を完了する時期と場所は、試験の選択によって特定されます。試験は、まず大きな影響の出ない領域で 1 つのデバイスを使用して開始し、その後、より大きな影響の発生する領域で、複数のデバイスに拡張する必要があります。影響を軽減することが可能な試験を選択するための、いくつかの検討事項を次に示します。

- 冗長性に起因する単一デバイスへの影響に対して、回復力のあるネットワーク領域内であるか。
- 選択されたデバイスの背後に存在するユーザ数が最小であり、実稼動で影響が発生した場合にはそれらのユーザが対応することのできるネットワーク領域内であるか。
- アーキテクチャ ラインに沿った試験の分離の検討。たとえば、アクセス、ディストリビューション、またはネットワークのコア レイヤのいずれか（あるいそれらすべて）で試験を行います。

この試験の期間は、デバイスの機能を十分にテストおよび評価するために必要な時間に基づいて決定する必要があります。試験の期間には、焼き込み試験と、通常のトラフィック負荷におけるネットワークの試験の両方の時間を含む必要があります。また、試験の期間は、コードのアップグレード手順と、その Cisco IOS が稼動しているネットワーク領域によっても異なります。新しいメジャー リリースの Cisco IOS を試験する場合には、より長い試験期間が必要です。逆に、そのアップグレードが新機能を少ししか含まないメンテナンス リリースである場合には、短い試験期間で対応できます。

試験フェーズ中は、初期テストと同様の方法で、結果を監視し、文書化する必要があります。この中には、ユーザ サーベイ、試験データの収集、問題の収集、および失敗と成功の基準を含めることができます。それぞれの担当者は、試験経過の追跡および監視に対して直接的な責任を負い、すべての問題が特定され、試験に関ったユーザおよびサービスが試験結果に満足していることを確認する必要があります。ほとんどの組織では、リリースが試験または実稼動環境で正常に動作した場合に、そのリリースを認証します。この手順は、一部の環境では重大な障害となります。測定方法や成功の基準が特定または文書化されていない状態で、成功したと認識される場合があるためです。

実装

実稼動ネットワーク内における試験フェーズが完了したら、次に、Cisco IOS の実装フェーズを開始します。実装フェーズには、実装のスロースタート、最終認証、アップグレードの準備、アップグレードの自動化、および最終検証など、ソフトウェア アップグレードの成功と実装の効率性を確認するためのいくつかの手順が含まれます。

実装のスロースタートは、新しくテストされたリリースを時間をかけて実装するプロセスです。このプロセスでは、最終認証を行って全面的な転換をする前に、ソフトウェア イメージが実稼動環境で完全に実現されていることを確認します。1 日めは 1 台のデバイスだけをアップグレードし、翌日に 2 台のデバイスをアップグレード、その翌日にはおそらくもう数台へと増やしてゆく組織もあります。実稼動環境に約 10 台のデバイスが配置されている組織では、特定の Cisco IOS バージョンが最終的に承認されるまでに 1 ～ 2 週間かかる場合もあります。最終認証を行う段階では、そのバージョンは、より高い信頼性を持ち、より迅速に配備できるようになります。

スロースタート プロセスの後、要件が満たされていることを確認するために、デバイス インベントリと、ブートストラップ、DRAM、およびフラッシュに対する Cisco IOS の最小標準のマトリックスを使用して、アップグレード対象であるすべてのデバイスを確認および検証する必要があります。このデータは、社内ツール、サードパーティ SNMP ツール、または CiscoWorks2000 RME を使用して取得することができます。CiscoWorks2000 SWIM では、実装の前にこれらの変数が確認または調査されます。ただし、想定されるデータを実装の試行中に認識することは、常に推奨されます。

類似した 100 台以上のデバイスのアップグレードを予定している場合、自動化された方法を使用することを強くお勧めします。1000 台のデバイスを社内アップグレードした例では、SWIM を使用するかどうかに関係なく、自動化によって大規模な配備におけるアップグレードの効率が上がり、デバイス アップグレードの成功の割合も改善されています。シスコは、アップグレード中に実行される検証の程度を根拠に、大規模な配備における CiscoWorks 2000 SWIM の使用を推奨します。問題が検出された場合には、SWIM を使用して Cisco IOS を前のバージョンに戻すこともできます。SWIM はアップグレード作業を作成およびスケジューリングすることによって機能し、それぞれの作業はデバイス、望ましいアップグレード イメージ、および作業のランタイムによって設定されます。各作業に含まれるデバイス アップグレードは 12 個までで、最大で 12 の作業が同時に実行できます。SWIM では、アップグレードの後に、Cisco IOS のアップグレードされたバージョンが正常に稼動しているかどうかも検証されます。各デバイスのアップグレードには、（検証も含めて）約 20 分間を割り当てることをお勧めします。 この計算に従えば、組織は 1 時間当たり 36 台のデバイスをアップグレードできます。潜在的な問題の発生を抑えるために、1 日にアップグレードするデバイス数は最大で 100 台までにすることをお勧めします。

自動化されたアップグレードを行った後には、アップグレードの成功を検証する必要があります。CiscoWorks2000 SWIM ツールでは、成功をより確実に検証するために、アップグレードの後にカスタマイズされたスクリプトを実行できます。ここでの検証には、ルータに適切な数のルートが含まれていることの検証、論理インターフェイスと物理インターフェイスがアップ状態でアクティブであることの確認、またはデバイスがアクセス可能であることの検証などがあります。次のサンプル チェックリストを使用すると、Cisco IOS の配備の成功を完全に検証できます。

- デバイスが正常にリロードしましたか。
- デバイスに ping が通りますか。また、Network Management System（NMS; ネットワーク管理システム）を介して、デバイスに到達できますか。
- デバイス上の予想されていたインターフェイスはアップ状態でアクティブですか。
- デバイスには、正確なルーティング プロトコル隣接関係がありますか。
- ルーティング テーブルは読み込まれていますか。
- デバイスで、トラフィックが正しく受け渡されていますか。

運用 - ハイ アベイラビリティ Cisco IOS の実装の管理

Cisco IOS 環境でハイ アベイラビリティのベスト プラクティスが実現されると、ネットワークの複雑さが軽減し、問題解決にかかる時間が短縮され、ネットワーク アベイラビリティが改善されます。Cisco IOS 管理の運用のセクションには、Cisco IOS を管理するための戦略、ツール、およびベスト プラクティスの方法論が含まれます。

Cisco IOS の運用のベスト プラクティスには、ソフトウェア バージョン管理、Cisco IOS Syslog 管理、問題管理、設定の標準化、およびアベイラビリティ管理が含まれます。ソフトウェア バージョン管理とは、特定されたソフトウェア トラック内におけるソフトウェアの一貫性を追跡、検証、および改善するプロセスです。Cisco IOS Syslog 管理とは、Cisco IOS によって生成されたより高い優先順位の Syslog メッセージに基づいて、予防的な監視や動作を行うプロセスです。問題管理とは、ソフトウェアに関連する問題が将来的に発生することを防ぐために、そのような問

題に関する重要な情報を迅速かつ効果的に収集する方法です。設定の標準化とは、テストされていないコードが実稼動環境で実行される可能性を軽減し、ネットワーク プロトコルと機能の動作を標準化するための、設定標準化のプロセスです。アベイラビリティ管理とは、メトリック、改善の目標、および改善プロジェクトに基づいて、アベイラビリティを改善するプロセスです。

Cisco IOS の運用における戦略とツール

多くの品質の戦略とツールは Cisco IOS 環境を管理する。Cisco IOS の運用における最初の主要な戦略は、環境をできるだけ簡素化し、設定と Cisco IOS のバージョンのばらつきをできるだけ減らすことです。Cisco IOS 証明書がすでに設定の一貫性がもう一つの重要な要素であるが、記載されています。設定の標準の作成に対しては、アーキテクチャ グループまたは技術グループが担当します。次に、Cisco IOS のバージョン管理および設定の標準と管理を通して、実装と運用のグループが、これらの標準を設定および保守します。

Cisco IOS の運用における 2 番目の戦略は、ネットワークの不具合を識別し、迅速に解決する能力です。通常、ネットワークの問題は、ユーザから通報を受ける前に運用グループによって発見されている必要があります。また発見された問題は、ネットワーク環境に対してそれ以上の影響や変更が発生する前に、できるだけ迅速に解決する必要があります。この領域での主要なベストプラクティスには、問題管理と Cisco IOS Syslog 管理があります。Cisco IOS ソフトウェアのクラッシュを迅速に診断するには、Cisco Output Interpreter が役立ちます。

3 番目の戦略は、一貫性のある改善です。この戦略のプライマリ プロセスは、品質ベースのアベイラビリティ改善プログラムを強化することです。Cisco IOS 関連の問題も含めたすべての問題に対して根本原因分析を実行することにより、テストのカバー範囲を広げ、問題解決にかかる時間を短縮し、停止による影響を排除または軽減するプロセスを改善できます。また、組織内で共通する問題を調査し、それらの問題をより迅速に解決するためのプロセスを構築することも可能です。

Cisco IOS の運用には、ソフトウェア バージョン管理のインベントリ管理（CiscoWorks2000 RME）、Syslog メッセージを管理する Syslog 管理、およびデバイス設定の一貫性を管理するデバイス設定マネージャが含まれます。

Syslog 管理

Syslog メッセージは、デバイスから収集サーバに対して送信されるメッセージです。これらのメッセージは、エラー（たとえば、リンクが使用不可能など）である場合と、情報を知らせるメッセージ（たとえば、デバイス上の端末を設定するために口グインしているユーザがいるなど）である場合があります。

Syslog 管理のツールでは、ルータやスイッチが受信した Syslog メッセージの記録および追跡が行われます。一部のツールには、重要なメッセージを見つけにくくする不要なメッセージを削除するためのフィルタが用意されています。Syslog ツールでは、受信メッセージに関するレポーティングも可能です。レポートは、期間、デバイス、メッセージ タイプ、またはメッセージ優先順位ごとに表示できます。

Cisco IOS 管理の最も一般的な syslog ツールは、CiscoWorks2000 RME Syslog Manager です。SL4NTやNetalのシェアウェアプログラム、OpenSystemsのPrivate Iなどのツールも 利用できます。

CiscoWorks Interface Configuration Manager

CiscoWorks2000 Device Configuration Manager では、アクティブなアーカイブが維持されてお

り、複数の Cisco ルータおよびスイッチ全体における設定変更を簡単に更新する方法も提供されています。Device Configuration Manager は、設定変更の行われるネットワークを監視し、変更が検出された場合にアーカイブを更新し、変更についての情報を Change Audit Service に記録します。Web ベースのユーザ インターフェイスを使用することで、特定の設定属性のアーカイブを検索し、2 つの設定ファイルの内容を比較して違いを簡単に確認することができます。

Cisco Output Interpreter

Cisco Output Interpreter は、ソフトウェアによる強制クラッシュを診断するために使用するツールです。このツールを使用すると、Cisco Technical Assistance Center (TAC) に連絡することなく、ソフトウェアの不具合を特定できます。または、ソフトウェア強制クラッシュが発生した場合に、TAC へ伝えるプライマリ情報を収集することもできます。このツールでは、少なくとも、必要な情報が収集できるため、問題を効率的に解決するために役立ちます。

ソフトウェア バージョン管理

ソフトウェア バージョン管理は、標準化されたソフトウェア バージョンだけを実装して、ネットワークを監視するプロセスです。このプロセスでは、ソフトウェアを検証し、バージョンに準拠しないソフトウェアを変更する場合もあります。一般に、ソフトウェアのバージョン管理は認証プロセスと標準制御を使用して行われます。多くの組織では、Web サーバのバージョンの標準をパブリッシュします。また、実装担当者は、どのバージョンが稼動しているかを確認して、標準に準拠していないバージョンをアップデートするための教育を受けます。一部の組織では品質ゲート プロセスを用意して、監査による二次的な検証を行い、実装期間中に標準が準拠されていることを確認しています。

運用が開始されてから、ネットワーク内に標準以外のバージョンが見つかることは珍しくありません。特に、ネットワークや運用の担当者が大勢いる場合によく起こります。原因としては、教育を受けていない新しい担当者がいる、boot コマンドが誤って設定されている、または確認されていない実装がある、などが考えられます。CiscoWorks 2000 RME では、すべてのデバイスを Cisco IOS のバージョンでソートすることができます。このようなツールを使用してソフトウェア バージョン標準を定期的に検証することをお勧めします。非標準のバージョンが確認されると即座にフラグが付けられ、そのバージョンを標準のバージョンに移行するために、「Trouble Ticket」または「Change Ticket」が開始されます。

予防的な Syslog 管理

Syslog の収集、監視、および分析は、その他の手段では識別が困難または不可能な、Cisco IOS に特有のより多くのネットワーク問題を解決するために推奨される、障害管理プロセスです。Syslog を収集、監視および分析することで、さらに深刻な問題がネットワークで発生したりユーザによって報告される前に、多くの障害を前もって特定、解決し、問題解決にかかる時間を短縮することができます。SNMP で多数の MIB 変数をポーリングする方法と比較して、Syslog では、さまざまな種類の問題をより効果的に収集することもできます。Syslog の収集、監視、および分析は、Cisco IOS を適切に設定し、CiscoWorks2000 RME などの Syslog 相関ツールまたは Syslog イベント管理のいずれか (あるいはその両方) を使用して行います。Syslog イベント管理は、収集された、重要なメッセージに対応する Syslog データを解析し、次にリアルタイムの通知および解決のためのアラートまたはトラップをイベント マネージャに転送することによって実行されます。

Syslog を監視するには、Syslog データを解析および報告するために、NMS ツールまたはスクリプトが必要です。これには、日付または期間、デバイス、Syslog メッセージ タイプ、またはメッセージ頻度によって Syslog メッセージをソートする機能が含まれます。大規模なネットワークでは、syslog データを解析し、イベント管理システムにアラートまたは通知またはアクションおよ

び技術者の送信、ツールやスクリプトが実行される introverted thatさまざまな syslog データのアラートを使用しない場合は、各日高プライオリティの syslog データをまだ確認し、潜在的な問題のトラブル チケットを作成する必要があります。事前に正常な監視、定期的な監査によって認識されない可能性がある disappear されない可能性がある状況を検出するために、履歴 syslog データの分析を実行する必要があるネットワークの問題を検出すると、サービスに影響を与えることになる前に問題を示すを提供する you start。

問題管理

多くの顧客は、問題管理プロセスの不足によって追加ダウンタイムが発生します。追加のダウンタイムが発生して、ネットワーク管理者がすぐに発生した時間を、問題の特定、情報収集、およびフル分析ソリューションに）、もむしろサービスに影響を与えるコマンドまたは設定変更の組み合わせを使用して問題を解決する必要があります。この領域で観察される行動には、デバイスをリロードしたり、問題とその根本原因を調査する前に IP ルーティング テーブルを消去することなどがあります。場合によっては、これは、最初のレベルのサポートでの問題解決の目標に起因しています。ソフトウェアに関連する問題が発生した場合は、どの問題においても、接続またはサービスの復旧前に根本原因分析に必要な情報を迅速に収集することを目標とする必要があります。

大規模な環境では、問題管理プロセスを適用することをお勧めします。このプロセスには、デフォルトの問題に関するある程度の説明、および tier 2 へ問題を報告する前に、show コマンドを使用して適切な情報を収集することが含まれます。tier 1 のサポートでは、ルータの消去やデバイスのリロードは行いません。好ましいのは、tier 1 のサポートで情報をすばやく収集し、tier 2 へ報告することです。最初の段階で、問題の特定および問題の説明に数分間を費やすことによって、根本的な原因を発見できる可能性が高くなるため、回避策、ラボでの確認、および不具合報告が可能になります。2 番めのレベルのサポートは、問題の診断または不具合レポートをまとめるためにシスコが必要とする情報の種類をよく理解している必要があります。必要な情報には、メモリ ダンプ、ルーティング情報の出力、およびデバイスの show コマンド出力などがあります。

設定の標準化

グローバルなデバイス設定の標準とは、複数のデバイスやサービスに共通な標準グローバル設定パラメータを維持する方法を意味します。その結果、企業全体でグローバルな設定の一貫性が実現されます。グローバル設定コマンドは、全デバイスに対して適用されるコマンドであり、個々のポート、プロトコル、またはインターフェイスに適用されるものではありません。グローバル設定コマンドは、一般的に、デバイスのアクセス、デバイスの全般的な動作、およびデバイスのセキュリティに影響を与えます。Cisco IOS のグローバル設定コマンドには、サービス コマンド、IP コマンド、vty コマンド、コンソール ポート コマンド、ロギング コマンド、AAA/TACACS+ コマンド、SNMP コマンド、およびバナー コマンドなどがあります。また、グローバル デバイスの設定標準に含まれる重要な項目に、デバイスの命名規則があります。この規則によって、管理者は Domain Name System（DNS; ドメイン ネーム システム）名に基づいて、デバイス、デバイス タイプ、およびデバイスの場所を確認できます。グローバル設定の整合性、複雑さを軽減し、ネットワークの supportability 向上に役立つため、ネットワーク環境全体の supportability と信頼性が重要です。多くの場合、サポートに関する問題は、設定が標準化されておらず、デバイスの動作、SNMP アクセス、および一般的なデバイス セキュリティが不適切、または一貫していないことが原因です。

グローバルなデバイス規格の維持は、同様のネットワーク デバイスのグローバル設定パラメータを作成および維持するための運用グループまたは内部エンジニアリング通常によって実現されます。また、メッセージがすべてのにプロビジョニングされたデバイスにダウンロードするために TFTP ディレクトリのグローバル コンフィギュレーション ファイルのコピーを提供することを推奨します。また、Web アクセスできるファイルは各設定パラメータの説明を標準コンフィギュレ

ーション ファイルを提供します。また、デバイスのグローバル設定を定期的に行って設定の一貫性を確認したり、デバイスが適切なグローバル設定標準を満たしていることを定期的に確認したりしている組織もあります。プロトコルおよびインターフェイスの設定の標準とは、インターフェイスおよびプロトコルの設定の標準を保守する方法を意味します。

プロトコルおよびインターフェイス設定の一貫性が実現されると、ネットワークの複雑さが軽減し、デバイスとプロトコルが予想通りに動作するようになり、ネットワークのサポート性が強化されるため、ネットワーク アベイラビリティが改善されます。プロトコルまたはインターフェイス設定の一貫性が実現されていない場合、予想外のデバイス動作、トラフィック ルーティングの問題、接続性に関する問題の増加、およびサポートの応答時間の遅延などが発生する可能性があります。インターフェイス設定の標準には、CDP インターフェイス記述子、キャッシング設定、およびその他のプロトコル固有の標準が含まれている必要があります。プロトコル固有の設定標準には、次のようなものがあります。

- IP ルーティングの設定
- DLSW の設定
- アクセス リスト コンフィギュレーション
- ATM の設定
- フレームリレー設定
- スパニングツリーの設定
- VLAN の割り当ておよび設定
- 仮想トランキング プロトコル（VTP）
- HSRP

注：ネットワーク内で何が設定されているかによって、他のプロトコル固有の設定標準を設定できます。

IP の標準には、次のようなものが含まれます。

- サブネットのサイズ
- 使用されている IP アドレス空間
- 使用されているルーティング プロトコル
- ルーティング プロトコルの設定

プロトコル設定と標準の維持は、通常、ネットワーク エンジニアと実装のグループがあります。技術グループは、標準の特定、テスト、検証、および文書化を担当します。次に、実装グループが、技術文書や設定テンプレートを使用して新しいサービスを提供します。技術グループは、一貫性を確認するために、必要とされる標準のすべての側面に関して文書を作成する必要があります。設定テンプレートは、コンフィギュレーション標準を強化するために作成する必要があります。運用グループも、標準に関する教育を受けて、非標準の設定を識別できるようになる必要があります。設定の一貫性はテスト、検証、および証明書のフェーズにも役立ちます。実際のところ、標準化された設定テンプレートが用意されていない場合、比較的大規模なネットワークで CIsco IOS のバージョンを適切にテスト、検証、または認証することはほぼ不可能です。

アベイラビリティ管理

アベイラビリティ管理は品質向上メトリックとしてネットワークの可用性を使用して品質向上するプロセスです。多くの組織は現在、可用性と停止のタイプを測定します。停止のタイプには、ハードウェア、ソフトウェア、リンクまたはキャリア、電源または環境、設計、あるいはユーザ エラーまたはプロセスなどがあります。障害を特定し、リカバリの根本原因の分析を実行することにより、企業は、アベイラビリティを向上させる方法を指定できます。ハイ アベイラビリティを達成したほとんどすべてのネットワークでは、いくつかの品質改善プロセスが実行されています。

付録 A - Cisco IOS リリースの概要

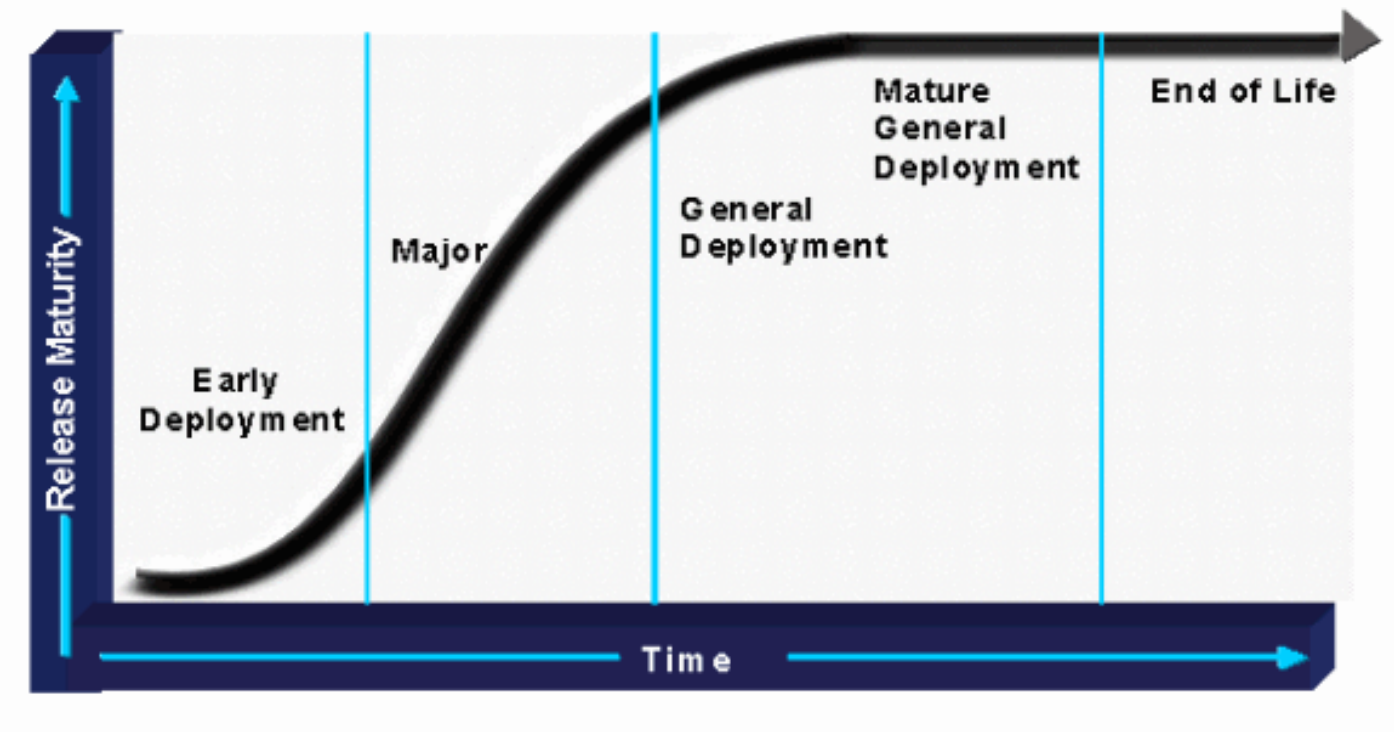
Cisco IOS ソフトウェア リリースの戦略は、正常なソフトウェア開発、品質保証、および短期間での市場投入という、お客様のネットワークの成功に欠かせない原則に基づいて構築されています。

リリースのプロセスは、次に説明する 4 つのリリース カテゴリを中心として定義されます。

- 早期導入リリース (ED)
- メジャー リリース
- 限定導入リリース (LD)
- 一般導入リリース (GD)

シスコでは、個々のリリース、対象となる市場、移行パス、新機能の説明などに関する情報が含まれた、IOS ロードマップを作成し、管理しています。

次の図は、Cisco IOS ソフトウェア リリースのライフサイクルを示しています。



ED リリース

Cisco IOS の ED リリースは、市場に新しい開発を公開する媒体となります。ED リリースの各メンテナンス リビジョンには、不具合修正だけではなく、新機能、新しいプラットフォームのサポート、およびプロトコルと Cisco IOS インフラストラクチャに対する一般的な拡張も含まれています。ED リリースの機能およびプラットフォームは、1 ~ 2 年ごとに Cisco IOS の次の主要リリースに移植されます。

ED リリースには 4 つのタイプがあり、それぞれリリース モデルとライフ サイクル マイルストーンがわずかに異なります。ED リリースは、次のように分類できます。

- **Consolidated Technology Early Deployment (CTED; 統合技術初期配備) リリース : 新しい Cisco IOS リリース モデルでは、「T」トレインとも呼ばれる統合 ED リリース トレインを使用して、新機能、新しいハードウェア プラットフォーム、および Cisco IOS へのその他の**

拡張を導入します。これらは、社内の Business Unit（BU）および Line Of Business（LOB）の定義の枠を超えているため、統合テクノロジーと呼ばれます。統合テクノロジー リリースの例には、Cisco IOS 11.3T、12.0T、および 12.1T があります。

- **Specific Technology Early Deployment（STED; 特定技術初期配備）リリース：STED リリースには、特定のテクノロジーまたはマーケットを対象としている点を除き、CTED リリースと同様の機能を持つ特徴があります。**STED は常に特定のプラットフォーム向けにリリースされ、完全に Cisco BU の監督下にあります。STED リリースは、メジャー リリース バージョンに付加された 2 文字によって識別されます。STED リリースの例には、Cisco IOS 11.3NA、11.3MA、11.3WA、および 12.0DA があります。
- **Specific Market Early Deployment（SMED; 特定市場初期配備）リリース：Cisco IOS の SMED は、特定の垂直マーケット（ISP、企業、金融機関、通信会社など）を対象としている点で、STED と区別されます。**SMED には、目的の垂直市場で使用される特定プラットフォームだけを対象した、特定のテクノロジー機能要件が含まれます。SMED は、目的の垂直市場に関連する特定のプラットフォーム専用に構築されますが、CTED は、より広範なテクノロジー要件に基づいて、複数のプラットフォーム用に構築されます。この点で、SMED は CTED と異なります。Cisco IOS SMED リリースは、（CTED と同様に）メジャー リリース バージョンに付加された、1 文字のアルファベットによって識別されます。 SMED の例には、Cisco IOS 12.0S や 12.1E があります。
- **X リリース（XED）、または短期初期配備リリース：Cisco IOS の XED リリースは、新しいハードウェアとテクノロジーを市場に公開します。**ソフトウェア メンテナンス リビジョンや定期的なソフトウェア暫定リビジョンは提供されません。CTED と統合される前に XED に不具合が発見された場合は、ソフトウェア リビルドが開始され、名前に数が付加されます。たとえば、Cisco IOS リリース 12.0(2)XB1 および 12.0(2)XB2 は、12.0(2)XB リビルドの例です。

メジャーリリース

メジャー リリースは、Cisco IOS ソフトウェア製品の配備における主要な媒体です。メジャー リリースは Cisco IOS 技術部門によって管理されており、機能、プラットフォーム、機能性、テクノロジー、および以前の ED リリース以降に増設されたホストが統合されています。Cisco IOS メジャー リリースでは、安定性と品質の強化を目指しています。そのため、メジャー リリースで機能またはプラットフォームが追加されることはありません。各メンテナンス リビジョンでは、不具合修正だけが行われます。たとえば、Cisco IOS ソフトウェア リリース 12.1 および 12.2 は、メジャー リリースです。

メジャー リリースには、メンテナンス リリースと呼ばれる定期的なメンテナンス アップデートがあります。メンテナンス リリースは、完全に回帰テスト済みで、最新の不具合修正が組み込まれていますが、新しいプラットフォームや機能はサポートしません。メジャーリリースとそのメンテナンス レベルは、メジャー リリースのリリース番号によって識別されます。Cisco IOS ソフトウェア リリース 12.0(7) の場合、12.0 がメジャー リリース番号で、7 がメンテナンス レベルです。完全なリリース番号は 12.0(7) です。 Cisco IOS ソフトウェア リリース 12.1 (3) の場合も同様に、12.1 がメジャー リリースで、12.1(3) は 3 番目のメンテナンス リリースです。

限定導入（LD）リリース

LD は、主要リリースの FCS と一般導入の中間に位置する、Cisco IOS の完成度を表すフェーズです。Cisco IOS の ED リリースは GD 認証を獲得することがないため、限定導入フェーズを過ぎた後は使用されません。

一般導入（GD）リリース

リリース ライフ サイクルのある時点で、シスコはメジャー リリースが GD 認証のために準備されたことを宣言します。GD の状態になれるのは、メジャー リリースだけです。あるリリースについて、シスコが次の条件を満たしていると判断したときに、メジャー リリースが GD 認証のマイルストーンに到達します。

- さまざまなネットワークで、市場に幅広く公開されていること。
- 安定性傾向および不具合傾向のメトリックによって認定されていること。
- お客様の満足度調査によって認定されていること。
- 直前の 4 つのメンテナンス リリースにおいて、お客様の発見する不具合が平均的に減少していること。

TAC エンジニア、Advanced Engineering Services（AES）エンジニア、システム テスト エンジニアリング、および Cisco IOS エンジニアリングによって職能上の枠を越えたカスタマー サポートの GD 認証チームが結成され、対象リリースにおけるすべての未解決の不具合を評価します。GD 認証の最終的な承認は、このチームによって与えられます。リリースが GD ステータスを獲得すると、そのリリースに続くリビジョンもすべて GD になります。そのため、あるリリースが GD であると宣言された場合、そのリリースは自動的に限定的なメンテナンス フェーズに入ることになります。このフェーズでは、大幅なコードの手直しをともなう不具合修正などのコードの技術的な修正は、プログラム マネージャによって厳密に制限され、管理されます。そうすることで、Cisco IOS ソフトウェアの GD 認証されたバージョンに、不具合が入り込むことを防ぎます。GD になるのは、特定のメンテナンス バージョンです。そのリリースの後続のメンテナンス アップデートも GD リリースになります。たとえば、Cisco IOS ソフトウェア リリース 12.0 は、12.0(8) での GD 認証を取得しました。 したがって、Cisco IOS ソフトウェア リリース 12.0(9)、12.0(10) なども、GD リリースです。

実験イメージまたは診断イメージ

実験イメージまたは診断イメージは、エンジニアリング スペシャルとも呼ばれ、ソフトウェアに関連する重大な問題が発見されたときにのみ作成されます。これらのイメージは、通常のリリース プロセスには含まれません。この種類のイメージは、お客様固有のビルドであり、問題の診断、不具合修正のテスト、または迅速な修正の提供を目的としています。次回の暫定リリースまたはメンテナンス リリースまで待てないような場合には、すぐに修正が提供される場合もあります。実験イメージまたは診断イメージは、任意のリリース タイプのメンテナンス バージョンや暫定バージョンなどの、サポートされている任意のソフトウェアをベースに構築される場合があります。公式な命名規則はありませんが、多くの場合は、ベースのイメージ名にイニシャル（実験イメージであれば exp）または数字を追加します。これらのイメージは、シスコの開発との関連の中で一時的にサポートされるだけです。Cisco TAC および Cisco IOS のリリース工程では、シンボル テーブルやベース イメージ履歴などの、サポート文書が管理されないためです。これらのイメージには、シスコの社内テストが適用されません。

リリースのライフサイクル マイルストーン

GD リリースは、ある時点において、最新のネットワーキング テクノロジーを備えたより新しいリリースによって置き換えられます。したがって、次の 3 つの主要なマイルストーンを使用して、リリースを停止するプロセスが確立されています。

- **End of Sales(EOS)**：メジャーリリースの場合、EOS日はFirst Commercial Shipment(FCS)日から3年後です。これによって、新しいシステムのためにリリースの購入が可能な最終の日付が設定されます。EOS リリースのメンテナンス アップグレードは、引き続き Cisco Connection Online（CCO）からダウンロードして入手可能です。
- **End of Engineering(EOE)**:EOEリリースはGDリリースの最後のメンテナンスリリースで、通常はEOSリリースから約3ヵ月後にリリースされます。お客様は、CCO から EOE リリース

をダウンロードできるだけではなく、Cisco TAC からのテクニカル サポートも引き続き受けることができます。EOS リリース予定日の 1 年前に、EOS および EOE がリリースされることと、それぞれのリリース日を公表する製品速報が発表されます。この時点で、最新のネットワーキング テクノロジーを利用するために、Cisco IOS ソフトウェアのアップグレードに関する調査を開始する必要があります。

- **End of Life (EOL; 廃止) ：リリース ライフ サイクルの終了時には、Cisco IOS ソフトウェア リリースに対するすべてのサポートが終了します。EOL の日付以降は、リリースをダウンロードできなくなります。**一般的に、EOL の日付は、EOE の日付の 5 年後になります。実際の EOL の日付の約 1 年前に、EOL の製品速報が発表されます。

Cisco IOS バージョンの命名規則

Cisco IOS イメージの命名規則によって、リリースされているすべてのイメージの完全なプロファイルが提供されます。名前には、常にメジャー リリースの ID とメンテナンス リリースの ID が含まれます。その他に、トレインを示す文字、リビルドを示す文字 (メンテナンス リリースの場合) 、BU 固有の機能番号、および BU 固有の機能番号のリビルド ID も名前に含まれる場合があります。書式の説明を、次に示します。

命名規則のセクション	説明
x.y	メジャー リリース値を表す、「.」で分けられた 2 つの数字 (1 桁または 2 桁) の ID の組み合わせ。この値は、Cisco IOS のマーケティングによって決定されます。例：12.1
z	x.y のメンテナンス リリースを示す、1 ～ 3 個の数字。これは 8 週間ごとに発生します。ベータ版は 0、FCS は 1、および最初のメンテナンス リリースは 2 です。例：12.1(2)
p	x.y (z) のリビルドを示す 1 文字のアルファベット。 最初のリビルドが小文字の「a」、次が「b」、その後も同様です。例：12.1 (2a)
A	1 ～ 3 文字のアルファベットは、リリース トレインを示し、CTED、STED、および X の各リリースでは必須です。この文字によって、製品ファミリやプラットフォームも示されます。CTED リリースおよび STED リリースでは、2 文字が使用されます。最初の文字は技術を示し、2 番目の文字は差別化のために使用されます。以下に、いくつかの例を示します。 A = Access Server/Dial technology (example:11.3AA) B = Broadband (example:12.2B) D = xDSL technology (example:12.2DA) E = Enterprise feature set (example:12.1E) H = SDH/SONET technology (example:11.3HA) N = Voice, Multimedia, Conference (example:11.3NA)

	M = Mobile (example:12.2MB) S = Service Provider (example:12.0S) T = Consolidated Technology (example:12.0T) W = ATM/LAN Switching/Layer 3 (example:12.0W5) リリース名の最初に「X」がある場合は、CTED の「T」トレインに基づく 1 回だけのリリースを意味します。たとえば、XA、XB、XC などとなります。リリース名の 2 番目の部分に「X」または「Y」がある場合は、STED リリースをベースにしているか関連している、短期 ED リリースを意味します。たとえば、11.3NX (11.3NA ベース)、11.3WX (11.3WA ベース) などとなります。
o	特定のリリース値のリビルドを示す、オプションの 1 桁または 2 桁の数字。リビルドではない場合、空欄にしておきます。1 から開始し、次に 2、その後も同様です。例：12.1(2)T1、12.1(2)XE2
u	BU 固有のリリースの機能を示す、1 桁または 2 桁の数字。この値は、BU のマーケティング チームによって決定されます。例：11.3(6)WA4、12.0(1)W5
v	BU 固有のコードのメンテナンス リリースを示す、1 ～ 2 桁の数字。ベータ版は 0、FCS は 1、および最初のメンテナンス リリースは 2 です。例：11.3(6)WA4(9), 12.0(1)W5(6)
p	特定の技術リリースのリビルドを示す 1 文字のアルファベット。最初のリビルドが小文字の「a」、次が「b」、その後も同様です。例：11.3(6)WA4(9a) は 11.3(6)WA4(9) のリビルドとなります。

次のグラフでは、Cisco IOS 命名規則のそれぞれのセクションにラベルを付けてあります。

付属 B - Cisco IOS の信頼性

Cisco IOS の信頼性は、シスコが継続して改善に取り組んでいる領域です。お客様志向のベストプラクティスを説明する前に、シスコ社内における IOS の品質および信頼性への取り組みに関す

る理解が必要です。これらのセクションは、Cisco IOS ソフトウェアの品質に対するシスコの最近の取り組みについての概要と、ソフトウェアの信頼性に関連してお客様に必要な前提を説明することを主な目的としています。

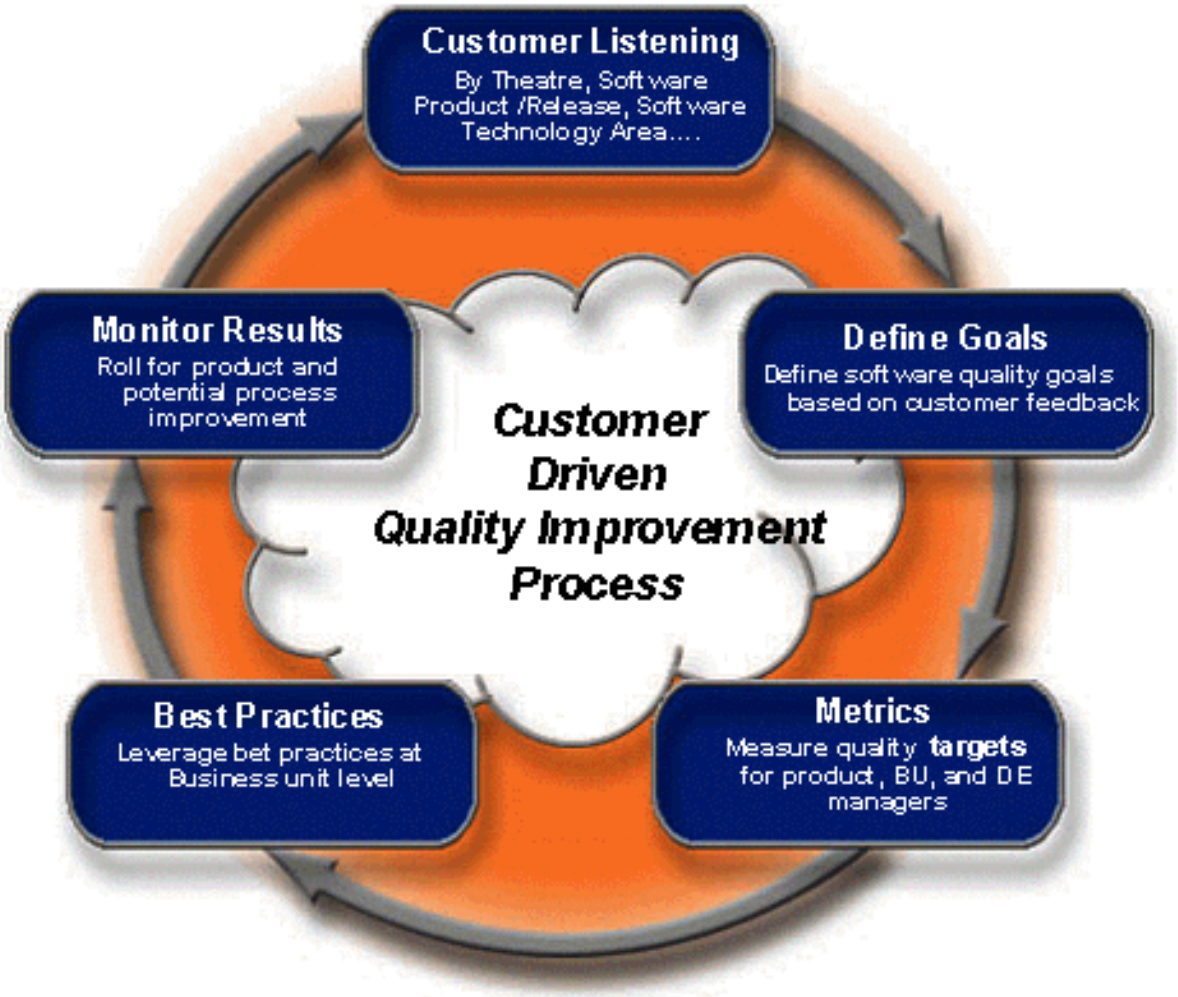
Cisco IOS 品質プログラム

シスコには、Great Engineering Methodology（GEM）と呼ばれる、明確な IOS 開発プロセスが用意されています。 このプロセスのライフサイクルは、次の 3 つのフェーズで構成されます。

- 戦略と計画
- 実行
- 導入

ライフサイクル内の一般的な領域には、機能導入におけるプライオリティ設定、開発、テスト プロセス、ソフトウェア導入フェーズ、First Customer Shipment（FCS; お客様向け出荷開始）、GD、および維持エンジニアリングなどが含まれます。シスコは、International Standards Organization（ISO; 国際標準化機構）、Telcordia（旧 Bellcore）、IEEE、および Carnegie Mellon Software Engineering Institute などの組織による、多数のソフトウェア品質ベスト プラクティス ガイドラインに準拠しています。これらのガイドラインは、シスコの GEM プロセスに取り入れられています。シスコのソフトウェア開発プロセスは、ISO 9001（1994 年版）による認証を受けています。

Cisco IOS ソフトウェア品質改善の主要なプロセスは、お客様主導のプロセスです。このプロセスによって、シスコはお客様のご意見を伺い、目標とメトリックを定義し、ベスト プラクティスを実装し、結果を監視します。ソフトウェア品質を改善するために結成された、組織の枠を越えたチームが、このプロセスの中核となっています。Cisco IOS 品質改善プロセスのダイアグラムを次に示します。



品質改善のプロセスには、2002 年会計年度以降、明確で重要な目標が用意されています。これらの目標の主要な焦点は、ソフトウェア問題をテスト サイクルの初期段階において特定することによって不具合を削減し、不具合の未処理件数を減少させ、機能の一貫性およびソフトウェア リリースの透明性を強化し、一貫した予想可能なリリース スケジュールおよび高いソフトウェア品質を提供することにあります。これらの領域に取り組む第一歩としては、脆弱なテスト カバレッジを特定する新しいテスト カバレッジ ツール、テストを修正するためのアクション プロセスの改善、および Cisco IOS システムの回帰テストの拡張などがあります。これらの問題に取り組むために新たにリソースが投入され、すべての主要な Cisco IOS ソフトウェア リリースのために、実行力のある、職能上の枠を越えて結成されたチームが全力を尽くしています。

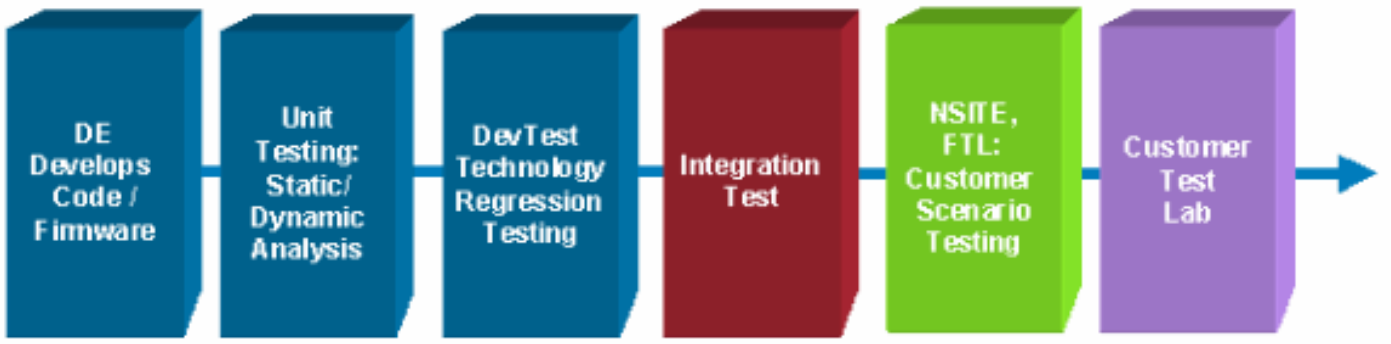
Cisco IOS リリースのテスト

シスコ社内でのソフトウェアの信頼性に関する品質への取り組みにおいて、テスト品質、テスト目的、テスト カバレッジは欠くことのできない領域です。シスコの掲げる IOS の品質目標は次のとおりです。

- シスコ社内で検出されているリグレッション障害を削減する。これには、開発のより高い品質、およびスタティック/ダイナミック分析によるさらに多くの問題の特定が含まれます。
- お客様によって発見される不具合を削減する。
- 未解決の不具合の総数を削減する。
- ソフトウェア リリースの透明性および機能の一貫性を強化する。
- 主要リリースおよびメンテナンス リリースにおいて、リリース スケジュールと高い品質を提供する。

シスコの社内テストでは、テストの各段階でそれぞれ異なる不具合が特定されます。全体的な目

標は、適切なラボ内において、適切な種類の不具合を発見することです。これは、いくつかの理由により重要です。まず、最も重要な理由は、同じ範囲をカバーするテストが、その後のテスト段階には存在していない可能性があることです。また、初期段階のテストは自動化できますが、後になるにしたがいテストの複雑さや必要な技術が増加するため、段階から段階へとテスト費用も劇的に増大します。次の図は、Cisco IOS のテスト スペクトラムを示しています。



最初の段階は、ソフトウェアの開発です。シスコは、ソフトウェアの初期品質の改善に役立つ、いくつかの取り組みをこの領域で行っています。開発グループでは、コードの確認または複合的なコードの確認を行っており、ソフトウェアの変更や新機能のコードに対して、他の開発者によって承認が与えられます。

次の段階は、ユニット テストです。ユニット テストは、ラボを利用せずにソフトウェアのインタラクションを調べるツールを使用して行います。DevTest は、機能/機能性テストおよび回帰テストを含むラボ テストです。機能/機能性テストの目的は、任意の機能の機能性を検査することです。このテストでは、設定、設定解除、および機能の仕様書で定義されている、すべての機能の組み合わせがテストされます。リグレッション テストの目的は、機能の機能性と動作を継続的に検証することであり、自動化されたテスト機能を使用して実行されます。このテストの主要な焦点は、多くの異なるネットワーク トポロジにおけるルーティング、スイッチング、および機能の機能性を、ping および限定的なトラフィック生成によってテストすることです。リグレッション テストを行う際に考えられる機能、プラットフォーム、ソフトウェア バージョン、およびトポロジの組み合わせはあまりにも多いため、シスコでは、これらのうちの限られた組み合わせに対してだけリグレッション テストを実施しています。それでも、現在利用されているリグレッション テスト スクリプトは 4000 以上にのぼります。統合テストでは、包括的な製品スイートおよび相互運用性をテストするための、拡張されたラボ テスト機能が提供されます。統合テストでは、テストの種類が拡張されて、相互互換性テスト、ストレス/パフォーマンス テスト、システム テスト、およびネガティブ テスト（予想外イベントのテスト）なども実行されるため、テストにおけるコード カバレッジも広がります。

次のラボ フェーズでは、一般的な顧客環境を対象としたエンドツーエンドのテストが提供されます。これらのテストは、顧客シナリオにおけるテストである Financial Test Lab（FTL）および NSITE として、上記の図に示されています。FTL は、ミッション クリティカルな金融業界の組織向けのテストを行うために構築されました。NSITE は、Cisco IOS のさまざまなテクノロジーをより詳細にテストするグループです。NSITE および FTL の各ラボでは、スケーラビリティとパフォーマンスのテスト、アップグレード性、アベイラビリティと回復力、相互互換性、およびサービサビリティなどの領域に焦点をあてています。サービサビリティにおける重点項目は、バルク プロビジョニングの問題、イベントの管理/相関、および負荷時のトラブルシューティングです。シスコ社内には、この他にも、さまざまな垂直市場においてこれらの領域のテストを行うためのラボが置かれています。

上記の図で最後に示されているラボは、お客様のラボです。お客様によるテストは品質に対する取り組みの延長上にあり、ハイ アベイラビリティ環境において、機能、設定、プラットフォーム、モジュール、およびトポロジの実際の組み合わせを完全にテストするために、導入されること

をお勧めします。テスト カバレッジには、特定のトポロジにおけるネットワーク スケーラビリティとパフォーマンス、特定のアプリケーション テスト、特定の設定におけるネガティブ テスト、シスコ製以外のデバイスとの相互運用性テスト、および焼き込みテストなどを含む必要があります。

ソフトウェアの MTBF

全体的な信頼性における最も一般的なメトリックの 1 つは、Mean Time Between Failure（MTBF; 平均故障間隔）です。ハードウェアの信頼性に対して開発された MTBF による分析機能が利用できるため、MTBF はソフトウェアの信頼性においても役立ちます。ハードウェアの信頼性は、既存のいくつかの標準を使用することで、より正確に判定することが可能です。シスコでは、Telcordia Technologies の標準的な MTBF データに基づくパーツカウント方式を使用しています。ただし、ソフトウェアの MTBF には対応する分析方法論が存在しないため、現場での測定結果に依存して MTBF 分析を行う必要があります。

過去 3 年間、シスコは社内 IT ネットワークにおいてソフトウェアの信頼性に関する現場測定を実行してきました。その内容はシスコ内で文書化されています。調査は Cisco IOS デバイスのソフトウェア強制クラッシュに基づいており、このデータはネットワーク管理における SNMP トラップ情報および稼働時間情報を使用して測定できます。この調査では、特定のソフトウェア リリースに対する統計的な対数正規分布モデルを使用して、ソフトウェアの信頼性を分析しています。ソフトウェア不具合の Mean Time To Repair（MTTR; 平均修復時間）は、ルータの再起動および回復にかかる平均時間に基づいています。企業環境では 6 分の回復時間が費やされ、より規模の大きいインターネット サービス プロバイダー（ISP）では 15 分が費やされています。 この進行中の調査では、ソフトウェアは通常、リリース時、またはいくつかのメンテナンス バージョンの後には、ファイブ ナインズ（99.999 %）のアベイラビリティを満たしているという結果が出ています。また、ダウンタイムの原因としてソフトウェア強制クラッシュだけを使用した調査では、長期にわたってこれよりも高い結果が出ています。この調査で確認されている潜在的な MTBF 値は、初期導入ソフトウェアにおける 5,000 時間から、一般導入ソフトウェアにおける 50,000 時間までの範囲となっています。

この調査に対する最も一般的な反論は、ソフトウェアの信頼性に関連して発生した停止時間のすべてが、ソフトウェア強制クラッシュに含まれているわけではないというものです。このメトリックを品質向上の取り組みに対して使用した場合、ソフトウェア強制クラッシュの発生率の軽減には役立つ可能性がありますが、ソフトウェアの信頼性における他の重要な領域が無視される可能性もあります。統計的な方法によってソフトウェアの信頼性を正確に予測することが困難であるために、この反論に対する答えはほとんど出ていません。シスコのソフトウェア品質の統計担当者は、より幅広い停止タイプを使用してソフトウェアの MTBF を確実に予測するためには、より大規模な正確なデータのサンプル セットが必要であると結論付けています。さらに、ネットワークの複雑さ、ソフトウェア関連の問題を解決する担当者の知識、ネットワークの設計、使用可能な機能、およびソフトウェアの管理プロセスなどが一定でないこともまた、理論的な統計学による分析を困難にしている要因です。

この種のデリケートなデータの誤差のない収集が困難であるために、現時点では、現場測定によってソフトウェアの信頼性をより正確に予測するという業界の取り組みは完了していません。また、ほとんどの顧客は、可用性データの独自性により、ネットワークから直接可用性の情報を収集することを望んでいません。ただし、一部の組織では、ソフトウェアの信頼性に関するデータを収集しています。シスコは、ソフトウェアの停止に起因するアベイラビリティのメトリックの収集と、それらの停止に対する根本原因分析を推奨しています。ソフトウェアの高い信頼性を実現している組織では、このような予防的な立場をとって、管理可能な多くの方法を通してソフトウェアの信頼性を向上しています。

ソフトウェアの信頼性における前提

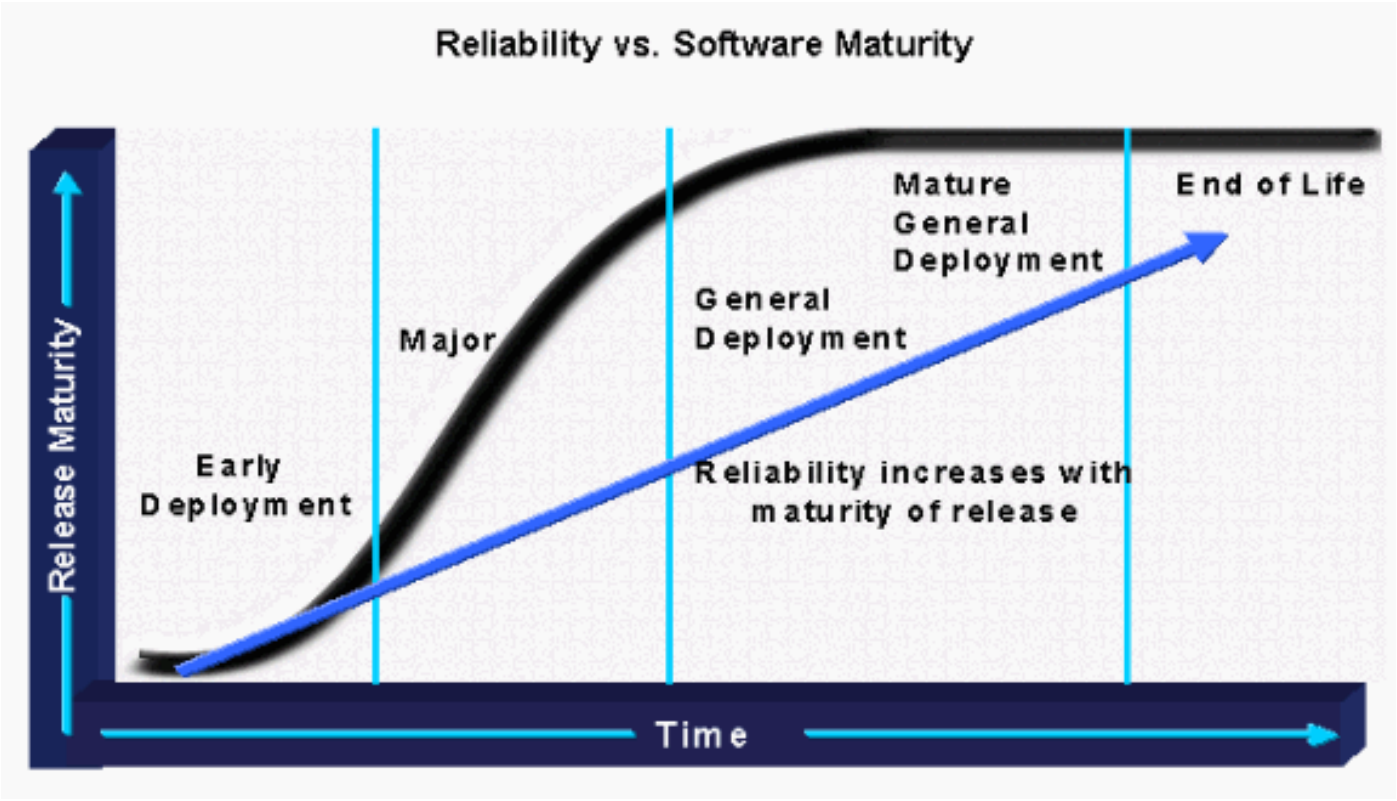
お客様によるフィードバック、Cisco IOS Technologies グループによる予防的な調査、および Cisco アドバンスト サービス チームによる根本原因分析の結果として、ソフトウェアの信頼性の改善に役立つ、新しい前提およびベスト プラクティスが形成されています。これらの前提の中核となるのは、テストの責任、ソフトウェアの完成度または開発されてからの期間、使用可能な機能、および導入されているソフトウェア バージョンの数です。

テストの責任

最初の新しい前提は、テストの責任に関するものです。シスコは常に、新しい機能および機能性のテストと検証を行い、これらが新製品において動作することを確認する責任を負っています。シスコはまた、新しいソフトウェア バージョンに対して回帰テストを行い、これらのバージョンの下位互換性を確認する責任も負っています。ただし、お客様の環境に存在する可能性のある、注意の必要なあらゆる問題（設計の特異性、負荷、およびトラフィック プロファイル）に対して、すべての機能、トポロジ、およびプラットフォームを検証することは、シスコにはできません。 お客様に用意されるハイ アベイラビリティのベスト プラクティスには、お客様定義の機能、設計、サービス、およびアプリケーション トラフィックを使用して実稼動ネットワークを模擬した、コラプスト ラボ トポロジ内におけるテストが含まれます。

信頼性とソフトウェアの完成度

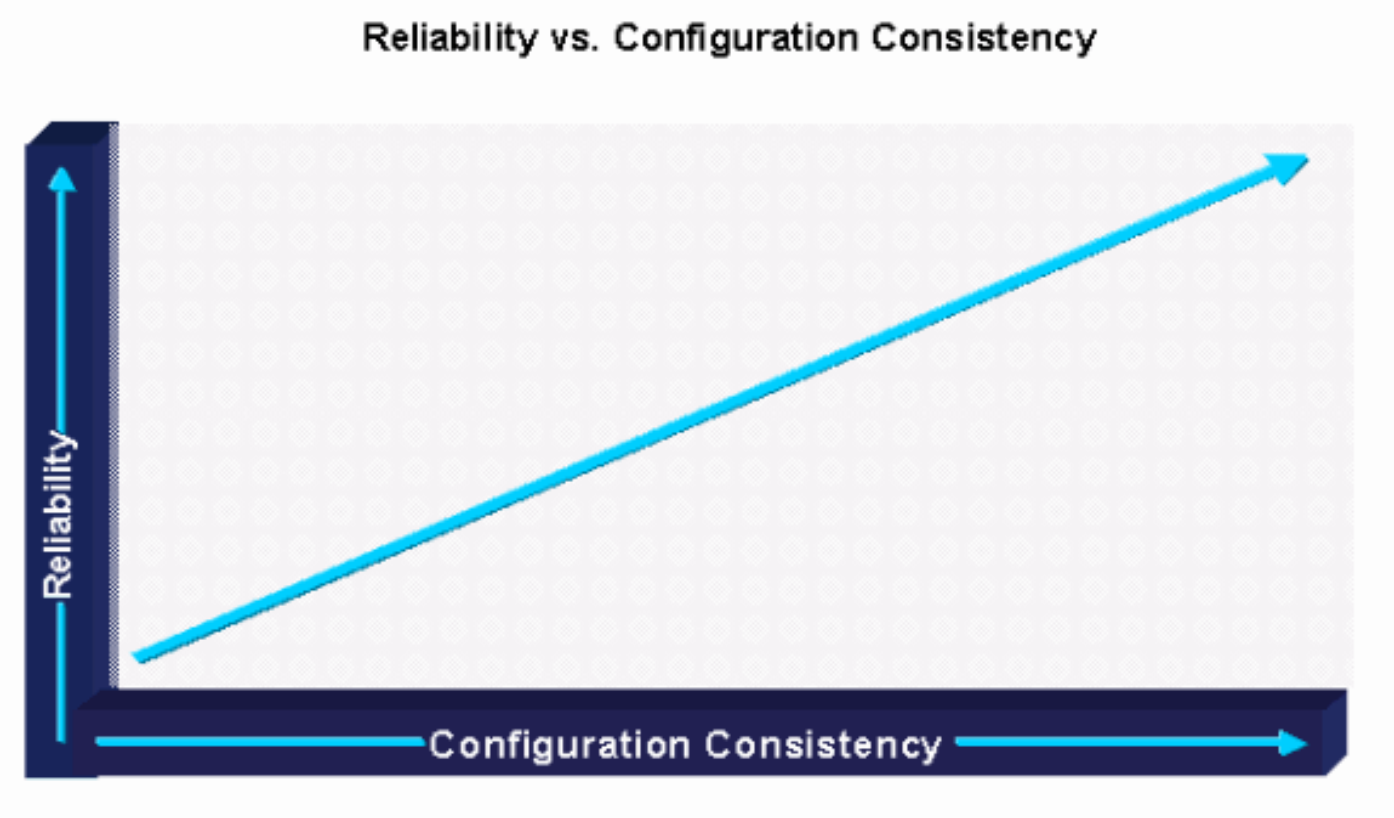
ソフトウェアの信頼性は、ソフトウェアの完成度の主要な要因です。公開されて使用され、確認された不具合が修正されることにより、ソフトウェアの完成度は高くなります。新機能を追加せずにソフトウェア完成度を確認するために、シスコのリリース工程ではトレイン リリースのアーキテクチャを採用しています。ハイ アベイラビリティを必要とするお客様は、現在必要な機能を含む、より完成度の高いソフトウェアを求めています。ソフトウェアの完成度、アベイラビリティの要件、および新機能または機能性用のビジネス ドライバは、互いに排他的な関係にあります。多くの組織には、許容できる完成度を示す標準またはガイドラインが用意されています。特定のトレインの 5 番目の暫定リリースだけを受け入れる組織もあります。あるいは、9 番目または GD 認証だけを受け入れる組織もあります。どの組織も最終的には、ソフトウェア完成度に対して、許容できるリスクの水準を決定する必要があります。



信頼性と機能および標準の量

ソフトウェアの信頼性とは、実稼働環境でどれだけのコードがテストされ、実行されるかでもあります。異なるハードウェア プラットフォームおよびモジュールの量が増加するにつれ、実行されるコードの量も増加します。そうすると、一般的に、ソフトウェアの不具合に対する脆弱性も増加します。設定されているプロトコルの量、設定の種類、およびトポロジまたは実装されている設計の種類についても同様です。設計、設定、プロトコル、およびハードウェア モジュールといった要因は、実行されるコード量の増加につながり、ソフトウェアの不具合に対するリスクまたは脆弱性の増大への一因となる可能性があります。

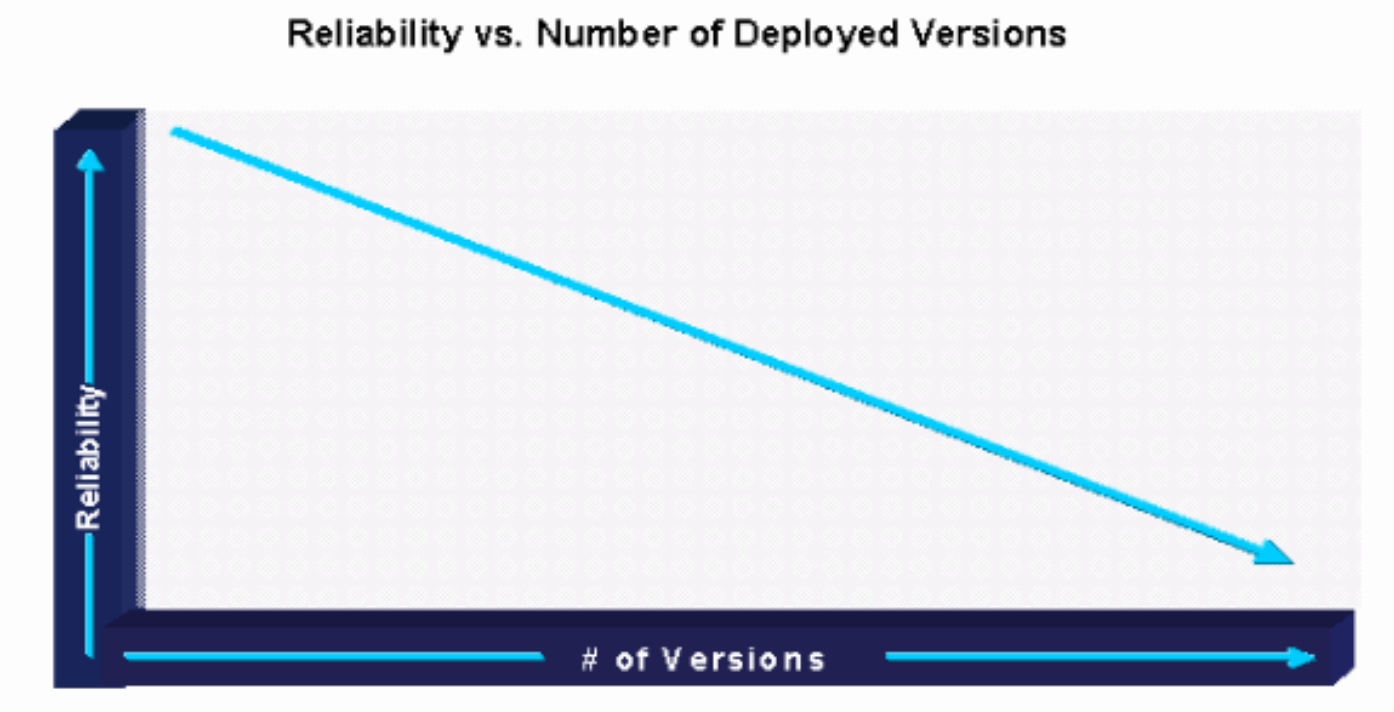
現在、ソフトウェアのリリース工程では、ある特定の領域における利用可能なコードを全般的に制限する、特別な目的を持つソフトウェアが用意されています。BU では、シスコ社内でより徹底的にテストされ、お客様によってより広範囲に使用されている設計および設定を推奨しています。テストされていないコードの利用を減らし、ソフトウェアの全体的な信頼性を強化するために、標準化されたモジュラ トポロジおよび標準設定のためのベスト プラクティスの採用も開始されています。ハイ アベイラビリティ ネットワークの中には、テストされていないコードの利用を削減するために、厳密な標準設定のガイドライン、モジュラ トポロジの標準、およびソフトウェアのバージョン管理が用意されているものもあります。



信頼性と配備されているバージョンの数

ソフトウェアの信頼性に関するもう 1 つの問題は、バージョン間の相互運用性と、複数のバージョンで実行されるコードの膨大な量です。ソフトウェア バージョンの数が増加するにつれ、実行されるコードの量も増加します。つまり、ソフトウェアの不具合に対する脆弱性も増加することになります。複数のバージョンの追加のコードが実行されると、信頼性に対するリスクは急激に増加します。現在では、特定の機能を使用したり、特定のプラットフォーム要件を満たすためには、ネットワーク内で実行するバージョン数をなるべく少なくする必要があることが認識されています。それにもかかわらず、ほぼ均一のネットワーク環境において 50 以上のバージョンを実行しているような場合は、一般的にソフトウェアの問題が存在しています。バージョンの数が多すぎて、適切な分析や検証ができないためです。

ソフトウェアの信頼性を強化するために、シスコの開発は、ソフトウェア リグレッション テストも行って、異なるソフトウェア バージョン間の互換性を確認しています。また、ソフトウェアのコードはよりモジュール化されており、時間の経過とともにバージョンが変更されても、コア モジュールが大幅に変更される可能性は高くありません。シスコのリリース工程では、お客様が使用できるソフトウェアの量も変更されています。既知の不具合または相互運用性の問題を持つバージョンは、不具合が発見されると迅速に CCO から削除されるためです。



関連情報

- Ciscoインターネットワーキング・オペレーティング・システム(IOS)
- テクニカル サポートとドキュメント – Cisco Systems