고가용성 네트워킹을 위한 Cisco IOS 관리: 모범 사례 백서

목차

소개
Cisco IOS 모범 사례 개요
소프트웨어 수명 주기 관리 프로세스 개요
계획 - Cisco IOS 관리 프레임워크 구축
Cisco IOS 계획을 위한 전략 및 툴
소프트웨어 버전 추적 정의
업그레이드 주기 및 정의
인증 프로세스
설계 - Cisco IOS 버전 선택 및 검증
Cisco IOS 선택 및 검증을 위한 전략 및 툴
후보자 관리
테스트 및 검증
구현 - 신속하고 성공적인 Cisco IOS 구축
Cisco IOS 구축을 위한 전략 및 툴
파일럿 프로세스
구현
운영 - 고가용성 관리 Cisco IOS 구현
Cisco IOS 운영을 위한 전략 및 툴
소프트웨어 버전 제어
사전 대응적 Syslog 관리
문제 관리
구성 표준화
가용성 관리
부록 A - Cisco IOS 릴리스 개요
릴리스 라이프 사이클 마일스톤
Cisco IOS 버전 명명 규칙
부록 B - Cisco IOS 신뢰성
Cisco IOS 품질 프로그램
Cisco IOS 릴리스 테스트
소프트웨어 MTBF
소프트웨어 신뢰성 가정
관련 정보

소개

신뢰할 수 있는 Cisco IOS® 소프트웨어를 구축하고 유지하는 것은 오늘날 비즈니스 크리티컬 네트

워크 환경에서 우선 순위이며, 이를 위해서는 Cisco와 고객이 무중단 가용성을 보장해야 합니다. Cisco는 소프트웨어 품질에 대한 노력에 주력해야 하지만, 네트워크 설계 및 지원 그룹은 Cisco IOS 소프트웨어 관리를 위한 모범 사례에 중점을 두어야 합니다. 그 목표는 더 높은 가용성과 소프트웨어 관리 효율성입니다. 이 방법은 소프트웨어 관리 모범 사례를 공유, 학습 및 구현하는 결합된 파트너십입니다.

이 문서에서는 엔터프라이즈 및 서비스 공급자 고객 모두에게 Cisco IOS 관리 방법의 효과적인 운영 프레임워크를 제공하여 소프트웨어 신뢰성 향상, 네트워크 복잡성 감소, 네트워크 가용성 향상을 지원합니다. 또한 이 프레임워크는 Cisco 릴리스 운영과 Cisco 고객 기반 간의 소프트웨어 관리 테스트 및 검증에서 책임과 중복되는 부분을 파악하여 소프트웨어 관리 효율성을 개선하는 데 도움이 됩니다.

Cisco IOS 모범 사례 개요

다음 표에서는 Cisco IOS 모범 사례에 대한 개요를 제공합니다. 이 표는 정의된 모범 사례에 대한 관리 개요, 격차 분석 체크리스트를 통해 현재 Cisco IOS 관리 사례를 검토하거나, Cisco IOS 관리 관련 프로세스를 생성하는 프레임워크로 사용할 수 있습니다.

이 표에서는 Cisco IOS 관리의 4가지 라이프사이클 구성 요소를 정의합니다. 각 테이블은 식별된 라이프사이클 영역에 대한 전략 및 툴 요약으로 시작합니다. 전략 및 툴 요약에는 정의된 라이프사이클 영역에만 적용되는 구체적인 모범 사례가 나와 있습니다.

Planning - Cisco IOS Management Framework 구축—Planning은 조직이 소프트웨어 업그레이드 시기, 업그레이드 위치, 잠재적 이미지를 테스트 및 검증하는 데 사용할 프로세스를 결정하는 데 필요한 Cisco IOS 관리의 초기 단계입니다.

모범 사례	세부 정보
Cisco IOS 계획을 위한 전략 및 툴	Cisco IOS 관리 계획을 시작하는 것은 현재 관행에 대한 솔직한 평가, 달성 가능한 목표 개발, 프로젝트 계획 등입니다.
소프트웨어 버전 추적 정의	소프트웨어 일관성을 유지할 수 있는 위치를 식별합니다. 소프트웨어 트랙은 고유한 지역, 플랫폼, 모듈 또는 기능 요구 사항에 따라 다른 영역과 차별화된 고유한 소프트웨어 버전 그룹으로 정의할 수 있습니다.
업그레이드 주기 및 정의	업그레이드 주기 정의는 소프트웨어 및 변경 관리의 기본 품질 단계로 정의할 수 있으며, 소프트웨어 업그레이드 주기가 시작되는 시기를 결정하는 데 사용됩니다.
인증 프로세스	인증 프로세스 단계에는 추적 식별, 업그레이드 주기 정의, 후보 관리, 테스트/검증 및 최소 일부 파일럿 생산 사용이 포함되어야 합니다.

설계 - IOS 버전 선택 및 검증—Cisco IOS 버전 선택 및 검증 프로세스를 잘 정의하면 업그레이드 시도 실패 및 예기치 않은 소프트웨어 결함으로 인한 예상치 못한 다운타임을 줄일 수 있습니다.

모범 사례	세부 정보
Cisco IOS 선택 및 검증을 위한 전략 및 툴	새 Cisco IOS 버전을 선택, 테스트 및 검증하기 위한 프로세스를 정의합니다. 여기에는 프로덕션 네트워크를 에뮬레이트하는 네트워크 테스트 랩이 포함됩니다.
후보자 관리	후보 관리는 특정 하드웨어 및 지원 기능 세트에 대한 소프트웨어 버전 요구 사항 및 잠재적 위험을 식별하는 것입니다.
테스트 및 검증	테스트 및 검증은 소프트웨어 관리 및 고가용성 네트워킹의 중요한 부분입니다. 적절한 랩 테스트를 통해 운영 다운타임을 크게 줄이고 네트워크 지원 인력을 교육하며 네트워크 구현 프로세스를 간소화할 수 있습니다.

구현 - Swift and Successful Cisco IOS Deployment - 잘 정의된 구현 프로세스를 통해 조직은 새 Cisco IOS 버전을 빠르고 성공적으로 구축할 수 있습니다.

모범 사례	세부 정보
Cisco IOS 구축을 위한 전략 및 툴	Cisco IOS 구축의 기본 전략은 파일럿 프로세스를 통해 최종 인증을 수행하고 업그레이드 툴과 잘 정의된 구현 프로세스를 사용하여 신속하게 구축하는 것입니다.
파일럿 프로세스	잠재적인 노출을 최소화하고 나머지 생산 문제를 더 안전하게 포착하려면 소프트웨어 파일럿을 사용하는 것이 좋습니다. 개별 파일럿 계획은 파일럿 선택, 파일럿 기간 및 측정을 고려해야 합니다.
구현	파일럿 단계를 완료한 후 Cisco IOS 구현 단계를 시작해야 합니다. 구현 단계에는 느린 시작, 최종 인증, 업그레이드 준비, 업그레이드 자동화, 최종 검증 등 소프트웨어 업그레이드 성공 및 효율성을 보장하기 위한 몇 가지 단계가 포함될 수 있습니다.

운영 - 고가용성 관리 Cisco IOS 구현—Cisco IOS 운영 모범 사례에는 소프트웨어 버전 제어, Cisco IOS Syslog 관리, 문제 관리, 구성 표준화, 가용성 관리 등이 포함됩니다.

모범 사례	세부 정보
Cisco IOS 운영을 위한 전략 및 툴	Cisco IOS 운영의 첫 번째 전략은 컨피그레이션 및 Cisco IOS 버전의 변동을 피하면서 환경을 최대한 간단하게 유지하는 것입니다. 두 번째 전략은 네트워크 결함을 식별하고 신속하게 해결할 수 있는 능력입니다.

소프트웨어 버전 제어	소프트웨어 버전 제어는 표준화된 소프트웨어 버전만 구현하고 비버전 규정 준수로 인해 소프트웨어를 검증하거나 변경할 수 있도록 네트워크를 모니터링하는 프로세스입니다.
사전 대응적 Syslog 관리	Syslog 수집, 모니터링 및 분석은 다른 방법으로 식별하기 어렵거나 불가능한 더 많은 Cisco IOS 특정 네트워크 문제를 해결하기 위해 권장되는 장애 관리 프로세스입니다.
문제 관리	문제 식별, 정보 수집 및 잘 분석된 솔루션 경로를 정의하는 상세한 문제 관리 프로세스 이 데이터를 사용하여 근본 원인을 파악할 수 있습니다.
구성 표준화	컨피그레이션 표준은 장치 및 서비스와 같은 모든 장치에서 표준 글로벌 컨피그레이션 매개변수를 생성하고 유지 관리하여 전사적 차원의 컨피그레이션 일관성을 유지하는 방식을 나타냅니다.
가용성 관리	가용성 관리는 네트워크 가용성을 품질 개선 메트릭으로 사용하여 품질을 개선하는 프로세스입니다.

소프트웨어 수명 주기 관리 프로세스 개요

Cisco IOS 소프트웨어 라이프사이클 관리는 안정적인 소프트웨어 구현 및 고가용성 네트워킹에 권장되는 계획, 설계, 구현 및 운영 프로세스 집합으로 정의됩니다. 여기에는 네트워크에서 Cisco IOS 버전을 선택, 검증 및 유지 관리하는 프로세스가 포함됩니다.

Cisco IOS 소프트웨어 수명 주기 관리의 목표는 생산 식별된 소프트웨어 결함 또는 소프트웨어 관련 변경/업그레이드 실패 가능성을 줄여 네트워크 가용성을 개선하는 것입니다. 이 문서에 정의된 모범 사례는 많은 Cisco 고객 및 Cisco Advanced Services 팀의 실제 경험을 바탕으로 그러한 결함을 줄이고 오류를 변경하는 것입니다. 소프트웨어 수명 주기 관리는 초기에 비용을 증가시킬 수 있지만, 운영 중단 감소 및 보다 간소화된 구축 및 지원 메커니즘으로 전반적인 소유 비용을 절감할 수 있습니다.

계획 - Cisco IOS 관리 프레임워크 구축

Planning은 조직이 소프트웨어 업그레이드 시기, 업그레이드 위치, 잠재적 이미지를 테스트하고 검증하는 데 사용할 프로세스를 결정하는 데 필요한 Cisco IOS 관리의 초기 단계입니다.

모범 사례에는 소프트웨어 버전 트랙 정의, 업그레이드 주기 및 정의, 내부 소프트웨어 인증 프로세스의 생성 등이 포함됩니다.

Cisco IOS 계획을 위한 전략 및 툴

Cisco IOS 관리 계획을 현재 사례, 달성 가능한 목표 개발, 프로젝트 계획 등의 솔직한 평가를 통해 시작합니다. 자체 평가는 이 문서의 모범 사례를 조직 내 프로세스와 비교하여 수행해야 합니다. 기본적인 질문은 다음과 같습니다.

- 조직에서 소프트웨어 테스트/검증을 포함하는 소프트웨어 인증 프로세스를 보유하고 있습니까?
- 조직에서 제한된 양의 Cisco IOS 버전을 네트워크에서 실행하는 Cisco IOS 소프트웨어 표준을

보유하고 있습니까?
- 조직에서 Cisco IOS 소프트웨어 업그레이드 시기를 결정하는 데 어려움이 있습니까?
- 조직에서 새로운 Cisco IOS 소프트웨어를 효율적이고 효과적으로 구축하는 데 어려움을 겪고 있습니까?
- 구축 후 다운타임 비용에 심각한 영향을 미치는 Cisco IOS 안정성 문제가 조직에 있습니까?

평가를 마친 후 Cisco IOS 소프트웨어 관리에 대한 목표 정의를 시작해야 합니다. 먼저 아키텍처 계획 그룹, 엔지니어링, 구현 및 운영 팀의 부서간 관리자 및/또는 리드를 결합하여 Cisco IOS 목표 및 프로세스 개선 프로젝트를 정의하는 데 도움이 됩니다. 초기 회의의 목표는 전체 목표, 역할 및 책임을 결정하고 작업 항목을 할당하고 초기 프로젝트 일정을 정의하는 것입니다. 또한 소프트웨어 관리 혜택을 결정하는 데 중요한 성공 요소와 메트릭을 정의합니다. 잠재적 메트릭은 다음과 같습니다.

- 가용성(소프트웨어 문제로 인해)
- 소프트웨어 업그레이드 비용
- 업그레이드에 필요한 시간
- 프로덕션에서 실행되는 소프트웨어 버전 수
- 소프트웨어 업그레이드 변경 성공/실패율

일부 조직은 전반적인 Cisco IOS 관리 프레임워크 계획 외에도 월별 또는 분기별로 진행 중인 소프트웨어 계획 회의를 정의합니다. 이 회의의 목표는 현재 소프트웨어 구축을 검토하고 새로운 소프트웨어 요구 사항을 계획하는 것입니다. Planning에는 현재 소프트웨어 관리 프로세스를 수정 또는 수정하거나 다른 소프트웨어 관리 단계에 대한 역할과 책임을 정의하는 작업이 포함될 수 있습니다.

계획 단계의 툴은 소프트웨어 인벤토리 관리 툴로만 구성됩니다. CiscoWorks 2000 RME(Resource Manager Essentials) Inventory Manager는 이 영역에서 사용되는 기본 툴입니다. CiscoWorks2000 RME Inventory Manager는 소프트웨어 버전, 디바이스 플랫폼, 메모리 크기 및 디바이스 이름을 기반으로 Cisco IOS 장치를 보고하고 정렬하는 웹 기반 보고 툴을 통해 Cisco 라우터 및 스위치의 버전 관리를 크게 간소화합니다.

소프트웨어 버전 추적 정의

첫 번째 Cisco IOS 소프트웨어 관리 계획 모범 사례는 소프트웨어 일관성을 유지할 수 있는 위치를 식별합니다. 소프트웨어 트랙은 고유한 지역, 플랫폼, 모듈 또는 기능 요구 사항에 따라 다른 영역과 차별화된 고유한 소프트웨어 버전 그룹화로 정의됩니다. 최적의 경우, 네트워크는 하나의 소프트웨어 버전만 실행해야 합니다. 따라서 소프트웨어 관리 관련 비용이 크게 절감되고 일관되고 쉽게 관리할 수 있는 환경이 제공됩니다. 그러나 대부분의 조직은 특정 영역 내의 기능, 플랫폼, 마이그레이션 및 가용성 문제로 인해 네트워크에서 여러 버전을 실행해야 하는 것이 현실입니다. 대부분의 경우 이기종 플랫폼에서 동일한 버전이 작동하지 않습니다. 다른 경우 조직은 한 버전이 모든 요구 사항을 지원할 때까지 기다릴 수 없습니다. 목표는 테스트/검증, 인증 및 업그레이드 요구 사항을 고려하여 네트워크에 대한 최소 소프트웨어 트랙을 식별하는 것입니다. 대부분의 경우 테스트/검증, 인증 및 업그레이드 비용을 전반적으로 낮추기 위한 트랙이 약간 더 많을 수 있습니다.

첫 번째 차별화 요소는 플랫폼 지원입니다. 일반적으로 LAN 스위치, WAN 스위치, 코어 라우터 및 에지 라우터는 각각 별도의 소프트웨어 트랙을 가지고 있습니다. DLSw(Data-link Switching), QoS(Quality of Service), IP 텔레포니 등 특정 기능 또는 서비스에 대한 다른 소프트웨어 트랙이 필요할 수 있습니다. 특히 이러한 요구 사항을 네트워크 내에서 현지화할 수 있는 경우 더욱 그렇습니다.

또 다른 기준은 신뢰성입니다. 많은 조직에서는 네트워크 코어 및 데이터 센터에 가장 안정적인 소프트웨어를 실행하면서 엣지를 향해 최신 고급 기능 또는 하드웨어 지원을 제공합니다. 반면 코어 또는 데이터 센터 환경에서는 확장성 또는 대역폭 기능이 가장 많이 필요합니다. 다른 WAN 라우터

플랫폼을 사용하는 대규모 배포 사이트 등 특정 플랫폼에 다른 트랙이 필요할 수 있습니다. 다음 표는 대기업 조직의 소프트웨어 트랙 정의 예입니다.

트랙	영역	하드웨어 플랫폼	기능	Cisco IOS 버전	인증 상태
1	LAN 코어 스위칭	6500	QoS	12.1E(A8)	테스트 중
2	LAN 액세스 스위치	2924 XL 2948 XL	UDLD(Unidirectional Link Detection Protocol), STP(Spanning Tree Protocol)	12.0(5.2)XU	인증 3/1/01
3	LAN 배포/액세스	5500 6509	수퍼바이저 3	5.4(4)	인증 7/1/01
4	RSM(Distribution Switch Route Switch Module)	RSM	OSPF(Open Shortest Path First) 라우팅	12.0(11)	인증 3/4/02
5	WAN 헤드엔드 배포	7505 7507 7204 7206	OSPF 프레임 릴레이	12.0(11)	인증 11/1/01
6	WAN 액세스	2600	OSPF 프레임 릴레이	12.1(8)	인증 6/1/01
7	IBM 연결	3600	SDLC(Synchronous Data Link Control) 헤드엔드	11.3(8)T1	인증 11/1/00

추적 지정도 시간이 지남에 따라 변경될 수 있습니다. 대부분의 경우 기능 또는 하드웨어 지원이 더 많은 메인라인 소프트웨어 버전으로 통합될 수 있으므로 여러 트랙이 결국 함께 마이그레이션됩니다. 트랙 정의가 정의되면 조직은 다른 정의된 프로세스를 사용하여 새 버전의 일관성 및 검증을 위해 마이그레이션할 수 있습니다. 트랙 정의는 또한 지속적인 노력입니다. 새로운 기능, 서비스, 하드웨어 또는 모듈 요구 사항이 확인될 때마다 새로운 트랙을 고려해야 합니다.

추적 프로세스를 시작하려는 조직은 새로 정의된 트랙 요구 사항으로 시작해야 합니다. 또는 경우에 따라 기존 네트워크의 안정화 프로젝트를 시작해야 합니다. 조직은 현재 트랙 정의를 가능하게 하는 기존 소프트웨어 버전과 몇 가지 식별 가능한 공통성을 가질 수도 있습니다. 대부분의 경우 네트워크 안정성이 충분할 경우 식별된 버전으로 신속하게 마이그레이션할 필요가 없습니다. 일반적으로 네트워크 아키텍처 또는 엔지니어링 그룹은 트랙 정의 프로세스를 소유합니다. 경우에 따라 한 개인이 트랙 정의를 담당해야 할 수도 있습니다. 다른 경우에는 프로젝트 리드가 개별 프로젝트

를 기반으로 소프트웨어 요구 사항과 새로운 트랙 정의를 개발하는 작업을 담당합니다. 또한 분기별로 트랙 정의를 검토하여 새 트랙이 필요한지 또는 기존 트랙의 통합 또는 업그레이드가 필요한지 여부를 확인하는 것도 좋습니다.

엄격한 버전 제어로 소프트웨어 트랙을 식별하고 유지 관리하는 조직은 운영 네트워크에서 소프트웨어 버전 수가 감소하면서 가장 성공적으로 성공을 거둔 것으로 나타났습니다. 따라서 일반적으로 소프트웨어 안정성과 전반적인 네트워크 신뢰성이 향상됩니다.

업그레이드 주기 및 정의

업그레이드 주기 정의는 소프트웨어 업그레이드 주기 시작 시기를 결정하는 데 사용되는 소프트웨어 및 변경 관리의 기본 품질 단계로 정의됩니다. 업그레이드 주기 정의를 통해 조직은 소프트웨어 업그레이드 주기를 적절하게 계획하고 필요한 리소스를 할당할 수 있습니다. 업그레이드 주기를 정의하지 않으면 일반적으로 현재 안정적인 버전의 기능 요구 사항으로 인해 소프트웨어 안정성 문제가 증가합니다. 프로덕션 사용이 필요하기 전에 새 버전을 올바르게 테스트하고 검증할 기회를 갖지 못한 조직도 위험에 노출될 수 있습니다.

이 방법의 중요한 측면은 소프트웨어 계획 프로세스를 언제 어느 정도 시작해야 하는지를 파악하는 것입니다. 이는 소프트웨어 문제의 주요 원인이 기업 실사 없이 생산 과정에서 기능, 서비스 또는 하드웨어 기능을 설정하거나 소프트웨어 관리 고려 사항 없이 새로운 Cisco IOS 버전으로 업그레이드하기 때문입니다. 다른 문제는 업그레이드하지 않습니다. 일반적인 소프트웨어 주기와 요구 사항을 무시함으로써 많은 고객이 다양한 주요 릴리스를 통해 소프트웨어를 업그레이드하는 어려운 과제에 직면해 있습니다. 이미지 크기, 기본 동작 변경, CLI(Command Level Interpreter) 변경, 프로토콜 변경 등으로 인해 어려움이 있습니다.

Cisco는 본 백서에서 정의한 모범 사례를 기반으로 새로운 주요 기능, 서비스 또는 하드웨어 지원이 필요할 때마다 이를 시작하는 것이 좋습니다. 정확한 테스트/검증 요구 사항을 결정하기 위해 인증 및 테스트/검증 수준을 분석(위험 기준)해야 합니다. 위험 분석은 지리적 위치, 논리적 위치(코어, 배포 또는 액세스 레이어) 또는 영향을 받는 예상 인원/고객 수를 기준으로 수행할 수 있습니다. 주요 기능 또는 하드웨어 기능이 현재 릴리스에 포함되어 있는 경우, 일부 간소화된 업그레이드 주기 프로세스도 시작해야 합니다. 이 기능이 상대적으로 작은 경우 위험을 고려한 다음 어떤 프로세스를 시작해야 할지 결정합니다. 또한 2년 이내에 소프트웨어를 업그레이드하여 조직이 비교적 최신 상태를 유지하고 업그레이드 프로세스가 그리 번거롭지 않도록 해야 합니다.

또한 EOL(End Of Life) 상태를 통과한 소프트웨어 열차는 버그 수정이 수행되지 않는다는 사실을 고려해야 합니다. 테스트/검증 프로세스가 거의 또는 전혀 없고 이로 인한 다운타임이 발생할 경우 더 많은 기능을 추가할 수 있으므로 비즈니스 요구 사항도 고려해야 합니다. 고객은 테스트 요구 사항을 고려할 때 Cisco 릴리스 작업에 수집된 최신 데이터를 고려해야 합니다. 버그 및 근본 원인을 분석한 결과, 버그 루트 원인 중 대부분이 영향을 받는 소프트웨어 영역 내에서 개발자가 코딩한 결과라는 것이 밝혀졌습니다. 즉, 조직이 기존 릴리스에서 네트워크에 특정 기능 또는 모듈을 추가하는 경우 해당 기능 또는 모듈과 관련된 버그가 발생할 가능성이 있지만 새로운 기능, 하드웨어 또는 모듈이 다른 영역에 영향을 미칠 가능성은 훨씬 적습니다. 이 데이터를 통해 조직은 새 서비스 또는 기능만 활성화된 다른 서비스와 함께 테스트함으로써 기존 릴리스에서 지원되는 새로운 기능 또는 모듈을 추가할 때 테스트 요구 사항을 줄일 수 있어야 합니다. 네트워크에서 발견된 몇 가지 중요한 버그를 기반으로 소프트웨어를 업그레이드할 때도 데이터를 고려해야 합니다.

다음 표는 주요 고가용성 엔터프라이즈 조직에 대한 권장 업그레이드 요구 사항을 보여줍니다.

소프트웨어 관리 트리거	**소프트웨어 수명 주기 요구 사항**
새로운 네트워크 서비스. 예를 들어, 새 ATM 백본	새로운 기능 테스트(다른 활성화된 서비스와 함께), 축소된 토

또는 새 VPN 서비스.	폴로지 테스트, what-if 성능 분석, 애플리케이션 프로필 테스트 등 완벽한 소프트웨어 라이프사이클 검증
현재 소프트웨어 릴리스에서는 새 네트워크 기능이 지원되지 않습니다. 예를 들면 QoS 및 MPLS(Multiprotocol Label Switching)가 있습니다.	새로운 기능 테스트, 축소된 토폴로지 테스트, what-if 성능 분석, 애플리케이션 프로필 테스트 등을 포함한 완벽한 소프트웨어 수명 주기 검증
현재 릴리스에 있는 새로운 주요 기능 또는 하드웨어 모듈. 예를 들어, 새 GigE 모듈, 멀티캐스트 지원 또는 DLSW를 추가합니다.	후보자 관리 프로세스. 릴리스 요구 사항에 따라 전체 검증이 가능합니다. 후보 관리가 현재 릴리스를 잠재적으로 허용 가능한 것으로 식별하는 경우 제한된 테스트/검증 가능
부 기능 추가. 예를 들어 액세스 제어를 위한 TACACS 디바이스입니다.	기능의 위험을 기준으로 후보 관리를 고려하십시오. 위험을 기반으로 새 기능을 테스트하거나 파일럿 수행하는 것을 고려하십시오.
2년 또는 분기별로 소프트웨어 리뷰 진행 중	현재 지원 가능한 릴리스를 식별하기 위해 전체 라이프사이클 관리와 관련된 후보 관리 및 비즈니스 결정

긴급 업그레이드

경우에 따라 조직은 치명적인 버그로 인해 소프트웨어를 업그레이드해야 하는 상황에 직면하기도 합니다. 따라서 조직에 긴급 업그레이드 방법이 없는 경우 문제가 발생할 수 있습니다. 소프트웨어 관련 문제는 소프트웨어 수명 주기 관리 없이 소프트웨어를 업그레이드하는 비관리 소프트웨어 업그레이드와 네트워크 장치가 지속적으로 다운되는 상황까지, 다음 후보 릴리스에 대한 인증/테스트가 완료되지 않았으므로 조직은 업그레이드하지 않습니다. Cisco는 제한된 테스트 및 파일럿이 네트워크의 비즈니스 크리티컬 영역이 아닌 영역에서 수행되는 이러한 상황에 대해 긴급 업그레이드 프로세스를 권장합니다.

명확한 해결 방법 없이 치명적인 오류가 발생하며 소프트웨어 결함과 관련된 문제가 있을 경우 Cisco는 Cisco 지원에 전적으로 참여하여 결함을 격리하고 수정 가능 여부를 확인하는 것이 좋습니다. 문제가 해결되면 Cisco는 긴급 업그레이드 주기를 통해 제한적인 다운타임으로 문제를 해결할 수 있는지 신속하게 판단할 것을 권장합니다. 대부분의 경우 조직에서 지원되는 버전의 코드를 실행하고 있으며, 문제 수정 사항은 기존의 최신 버전의 소프트웨어의 임시 버전에서 사용할 수 있습니다.

조직은 잠재적인 긴급 업그레이드에 대비할 수도 있습니다. 준비 과정에는 지원되는 Cisco IOS 릴리스로의 마이그레이션 및 인증 버전과 동일한 Cisco IOS 열차 내에서 후보 교체 버전을 식별/개발하는 작업이 포함됩니다. 지원되는 소프트웨어는 Cisco 개발 팀에서 확인된 소프트웨어 교육에 버그 픽스를 추가한다는 의미이므로 중요합니다. 네트워크에서 지원되는 소프트웨어를 유지함으로써 조직은 더 익숙하고 안정적인 코드 기반으로 인해 검증 시간을 단축합니다. 일반적으로 후보 대체는 동일한 Cisco IOS 열차 내에서 기능 또는 하드웨어 지원 추가 없이 새로운 임시 소프트웨어 이미지입니다. 특정 소프트웨어 교육의 조기 도입 단계에 있는 조직의 경우 후보 교체 전략이 특히 중요합니다.

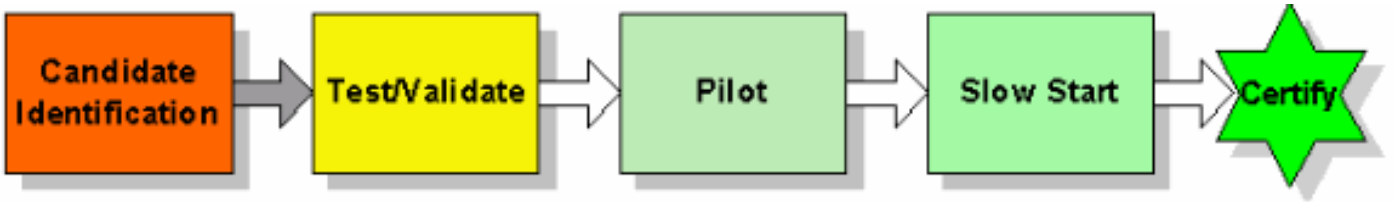
인증 프로세스

인증 프로세스를 통해 검증된 소프트웨어가 조직의 프로덕션 환경에 지속적으로 구축되도록 할 수 있습니다. 인증 프로세스 단계에는 추적 식별, 업그레이드 주기 정의, 후보 관리, 테스트/검증 및 일부 파일럿 생산 사용이 포함되어야 합니다. 그러나 단순한 인증 프로세스는 식별된 트랙 내에 일관된 소프트웨어 버전이 구축되도록 하는 데 도움이 됩니다.

인증 프로세스를 작성 및 관리할 아키텍처, 엔지니어링/구축 및 운영 담당자를 식별하여 인증 프로세스를 시작합니다. 먼저 비즈니스 목표와 리소스 기능을 검토하여 인증 프로세스가 계속 성공하도록 해야 합니다. 다음으로, 추적 관리, 라이프사이클 업그레이드 정의, 테스트/검증, 파일럿 등 인증 프로세스의 주요 단계에 대한 전체 책임을 개인이나 그룹에 할당합니다. 이러한 각 영역은 조직 내에서 정의, 승인 및 공식적으로 전달되어야 합니다.

인증 프로세스의 각 단계에서 품질 또는 승인에 대한 지침을 포함합니다. 프로세스가 다음 단계로 이동하기 전에 특정 품질 기준을 충족해야 하기 때문에 이를 품질 게이트 프로세스라고도 합니다. 이를 통해 인증 프로세스가 유효하고 할당된 리소스의 가치가 있는지 확인할 수 있습니다. 일반적으로 한 영역에서 품질 문제가 발견되면 이 프로세스는 한 단계로 역행합니다.

소프트웨어 품질 또는 예기치 않은 동작으로 인해 소프트웨어 후보가 정의된 인증 기준을 충족하지 못할 수 있습니다. 환경에 영향을 미치는 문제가 발견되면 조직은 이후 임시 릴리스를 인증하기 위해 더욱 간소화된 프로세스를 갖추어야 합니다. 이는 리소스 요구 사항을 줄이는 데 도움이 되며, 조직이 변경 사항 및 해결된 결함을 이해할 수 있는 경우 일반적으로 효과적입니다. 조직에서 최초 후보자와 관련된 문제를 경험하고 이후 임시 Cisco IOS 릴리스를 인증하는 것은 드문 일이 아닙니다. 또한 조직은 제한된 인증을 수행하거나 문제가 있을 경우 주의 사항을 제공할 수 있으며, 새로운 임시 릴리스가 검증되면 나중에 완전히 인증된 릴리스로 업그레이드할 수 있습니다. 아래 흐름도는 기본 인증 프로세스이며 품질 게이트를 포함합니다(각 블록에 대한 검토).



설계 - Cisco IOS 버전 선택 및 검증

Cisco IOS 버전을 선택 및 검증하기 위한 잘 정의된 방법론을 보유함으로써 업그레이드 시도 실패 및 예기치 않은 소프트웨어 결함으로 인한 예상치 못한 다운타임을 줄일 수 있습니다.

설계 단계에는 후보 관리 및 테스트/검증이 포함됩니다. 후보 관리는 정의된 소프트웨어 트랙의 특정 버전을 식별하는 데 사용되는 프로세스입니다. 테스트/검증은 인증 프로세스의 일부이며, 식별된 소프트웨어 버전이 필요한 트랙 내에서 성공하는지 확인합니다. 테스트/검증은 프로덕션 환경과 거의 비슷한 축소된 토폴로지 및 컨피그레이션을 포함하는 랩 환경에서 수행해야 합니다.

Cisco IOS 선택 및 검증을 위한 전략 및 툴

모든 조직에는 Cisco IOS 버전 선택 프로세스로 시작하여 네트워크에 대한 표준 Cisco IOS 버전을 선택하고 검증하는 프로세스가 있어야 합니다. 아키텍처, 엔지니어링 및 운영 부서의 부서간 팀이 후보 관리 프로세스를 정의하고 문서화해야 합니다. 승인이 완료되면 해당 배송 그룹으로 프로세스를 인계해야 합니다. 또한 식별된 후보자 정보로 업데이트할 수 있는 표준 후보 관리 템플릿을 생성하는 것이 좋습니다.

모든 조직이 프로덕션 환경을 쉽게 모방할 수 있는 정교한 랩 환경을 가지고 있는 것은 아닙니다. 일부 조직은 비용 및 주요 비즈니스 영향 없이 네트워크에서 새 버전을 파일럿 처리할 수 있는 능력 때문에 랩 테스트를 건너뜁니다. 그러나 고가용성 조직은 프로덕션 네트워크를 모방하는 랩을 구축하고 테스트/검증 프로세스를 개발하여 새로운 Cisco IOS 버전에 대한 높은 테스트 범위를 보장하는 것이 좋습니다. 한 조직은 연구실을 구축하는 데 약 6개월이 소요될 수 있습니다. 이 기간 동안 조직은 특정 테스트 계획 및 프로세스를 작성하여 랩이 모든 이점을 누릴 수 있도록 해야 합니다. Cisco IOS의 경우, 이는 각 필수 소프트웨어 트랙에 대한 특정 Cisco IOS 테스트 계획을 생성하는 것을 의미합니다. 이러한 프로세스는 많은 랩이 신제품 및 소프트웨어 소개에 사용되지 않기 때문에 대규모 조직에서 매우 중요합니다.

다음 섹션에서는 Cisco IOS 선택 및 검증에 사용할 후보 관리 및 테스트/검증 툴에 대해 간략하게 설명합니다.

후보자 관리 도구

참고: 아래 제공된 대부분의 도구를 사용하려면 등록된 사용자여야 하며 로그인해야 합니다.

- 릴리스 정보 - 릴리스의 하드웨어, 모듈 및 기능 지원에 대한 정보를 제공합니다. 후보 관리 중에 릴리스 정보를 검토하여 필요한 모든 하드웨어 및 소프트웨어 지원이 잠재적인 릴리스에 존재하는지 확인하고, 다른 기본 동작 또는 업그레이드 요구 사항을 포함한 마이그레이션 문제를 파악해야 합니다.

테스트 및 검증 툴

테스트 및 검증 툴은 새로운 하드웨어, 소프트웨어 및 애플리케이션을 비롯한 네트워크 솔루션을 테스트하고 검증하는 데 사용됩니다.

- **Traffic Generator** - 특정 프로토콜을 사용하여 특정 링크의 속도를 모델링하는 데 사용되는 멀티 프로토콜 트래픽 스트림 및 원시 패킷 속도를 생성합니다. 사용자는 소스, 대상 MAC 및 소켓 번호를 지정할 수 있습니다. 이러한 값은 지정된 단계에서 증가하거나 정적/고정 또는 임의 증분으로 구성할 수 있습니다. 트래픽 생성기는 다음 프로토콜에 대한 패킷을 생성할 수 있습니다 .IPIPX(Internetwork Packet Exchange)12월애플Xerox 네트워크 시스템(XNS)ICMP(Internet Control Message Protocol)IGMP(Internet Group Management Protocol)연결 없는 네트워크 서비스(CLNS)UDP(사용자 데이터그램 프로토콜)가상 통합 네트워크 서비스(VINES)데이터 링크 패킷툴은 Agent 및 Spient Communications에서 사용할 수 있습니다 .
- **Packet Counter/Capture/Decoder(Sniffer)(패킷 카운터/캡처/디코더(스니퍼)** - 고객이 모든 패킷 및 데이터 링크 레이어에서 선택적으로 패킷을 캡처하고 디코딩할 수 있도록 합니다. 이 툴에는 사용자가 필터를 지정할 수 있는 기능이 있으며, 이 기능을 통해 지정된 프로토콜 데이터만 캡처할 수 있습니다. 필터를 사용하면 특정 IP 주소, 포트 번호 또는 MAC 주소와 일치하는 패킷을 캡처하도록 지정할 수 있습니다. 툴은 스니퍼 기술에서 사용할 수 있습니다 .
- **Network Simulator/Emulator** - 고객이 프로덕션 네트워크 요구 사항에 따라 특정 라우터의 라우팅 테이블을 채울 수 있습니다. IP RIP(Routing Information Protocol), OSPF, IS-IS(Intermediate System-to-Intermediate System), IGRP(Interior Gateway Routing Protocol), EIGRP(Enhanced IGRP) 및 BGP(Border Gateway Protocol) 라우터의 생성을 지원합니다. 툴은 PacketStorm Communications 및 Spient Communications에서 사용할 수 있습니다.
- **Session Emulators** - 슬라이딩 윈도우 멀티 프로토콜 트래픽 스트림을 생성하며, 테스트 네트워크를 통해 수신 디바이스로 멀티 프로토콜 트래픽 스트림을 전송할 수 있습니다. 수신 디바이스는 패킷을 소스 쪽으로 다시 에코합니다. 소스 디바이스는 전송, 수신, 시퀀스 외 패킷 및 오류 패킷의 수를 확인합니다. 또한 이 툴은 TCP(Transmission Control Protocol)에서 창 매개변수를 정의할 수 있는 유연성을 제공하여 랩 네트워크의 클라이언트/서버 트래픽 세션을 자세히 모방합니다. 이 툴은 Empirix에서 사용할 수 있습니다 .

- **대규모 네트워크 에뮬레이터**—대규모 환경의 확장성을 테스트하는 데 도움이 됩니다. 이러한 툴은 프로덕션 환경을 더 자세히 모방하기 위해 제어 유형 트래픽을 만들고 랩 토폴로지에 쉽게 삽입할 수 있습니다. 이러한 기능에는 경로 삽입, 프로토콜 네이버, 레이어 2 프로토콜 네이버가 포함됩니다. 툴은 Agent 및 Spient Communications에서 사용할 수 있습니다 .
- **WAN 시뮬레이터**—대역폭과 지연이 잠재적으로 문제가 되는 엔터프라이즈 애플리케이션 트래픽을 테스트하는 데 이상적입니다. 이러한 툴을 통해 조직은 애플리케이션이 WAN을 통해 작동하는 방식을 확인하기 위해 예상 지연 및 대역폭으로 애플리케이션을 로컬로 테스트할 수 있습니다. 이러한 툴은 애플리케이션 개발 및 엔터프라이즈 조직 내의 애플리케이션 프로파일링 테스트 유형을 위해 자주 사용됩니다. Spirit Communications 및 Shunra 의 한 부서인 Adtech는 WAN 시뮬레이션 툴을 제공합니다.

후보자 관리

후보 관리는 특정 하드웨어 및 지원 기능 세트에 대한 소프트웨어 버전 요구 사항 및 잠재적 위험을 식별하는 프로세스입니다. 소프트웨어 요구 사항, 릴리스 정보, 소프트웨어 결함 및 잠재적 위험을 올바르게 조사하려면 릴리스 파일럿 전에 4~8시간을 사용하는 것이 좋습니다. 다음은 후보자 관리의 기본을 간략하게 설명합니다.

- CCO(Cisco Connection Online) 툴을 통해 소프트웨어 후보를 식별합니다.
- 위험 분석 소프트웨어 성숙도, 새로운 기능 또는 코드 지원
- 라이프사이클 전반에 걸쳐 알려진 소프트웨어 버그, 문제 및 요구 사항을 파악하고 추적합니다.
- 선택한 이미지의 기본 구성 동작을 식별합니다.
- 잠재적 후보자 변경에 대한 재평가 및 롤 포워드를 유지합니다.
- 버그 스크럽.
- Cisco Advanced Services 지원.

Cisco 프로덕션 및 소프트웨어 열차들의 수가 증가함에 따라 소프트웨어 후보를 식별하는 것이 더욱 복잡해졌습니다. 이제 CCO는 Cisco IOS 업그레이드 플래너, 소프트웨어 검색 툴, 소프트웨어-하드웨어 호환성 매트릭스, 조직이 잠재적인 릴리스 후보를 식별하는 데 도움이 되는 제품 업그레이드 툴을 비롯한 여러 가지 툴을 갖추고 있습니다. 이러한 툴은 http://www.cisco.com/cisco/software/navigator.html에서 확인할 수 있습니다.

다음으로 잠재적 후보 소프트웨어의 위험을 분석합니다. 이는 소프트웨어가 현재 성숙도 곡선에 있는 위치를 파악한 다음, 배포 요구 사항에 따라 릴리스 후보자의 잠재적 위험을 평가하는 프로세스입니다. 예를 들어, 조직이 ED(Early Deployment) 소프트웨어를 중요한 고가용성 환경에 배치하려는 경우, 성공적인 인증을 위한 관련 위험 및 리소스 요구 사항을 고려해야 합니다. 조직은 성공을 보장하기 위해 더 높은 위험 상황에 대비해 최소한 소프트웨어 관리 리소스를 추가해야 합니다. 반면 조직의 요구 사항을 충족하는 GD(General Deployment) 버전을 사용할 수 있는 경우 필요한 소프트웨어 관리 리소스가 줄어듭니다.

잠재적인 릴리스 및 위험이 확인되면 버그 스크럽을 수행하여 인증을 방지할 수 있는 치명적인 버그가 있는지 확인합니다. Cisco의 Bug Watcher, Bug Navigator 및 Bug Watcher Agent는 잠재적 문제를 식별하는 데 도움이 될 수 있으며, 소프트웨어 수명 주기 전반에 걸쳐 잠재적 보안 또는 결함 문제를 식별하는 데 사용해야 합니다.

새로운 소프트웨어 후보 역시 잠재적인 기본 구성 동작을 검토해야 합니다. 이 작업은 새 소프트웨어 이미지에 대한 릴리스 정보를 검토하고 지정된 플랫폼에 로드되는 잠재적 이미지와 구성 차이를 검토하여 수행할 수 있습니다. 후보 관리에는 선택한 버전이 프로세스의 특정 시점에 인증 기준을 충족하지 않을 경우 백아웃 버전 또는 go-to 버전의 식별도 포함될 수 있습니다. 조직은 지정된 트랙의 기능과 관련된 버그를 관찰하여 잠재적 인증 후보를 유지할 수 있습니다.

Cisco Advanced Services는 후보자 관리를 위한 훌륭한 툴이기도 합니다. 이 그룹은 다양한 수직 시장 환경에서 수많은 업계 전문가 간에 개발 프로세스 및 협업에 대한 심층적인 정보를 제공할 수 있습니다. 일반적으로 다른 조직에서 실행되는 운영 소프트웨어 버전에 대한 전문성 및 가시성 수준 때문에 Cisco 지원 내에 최고의 버그 스크럽 또는 후보 관리 기능이 있습니다.

테스트 및 검증

테스트 및 검증은 관리 모범 사례 및 고가용성 네트워킹의 중요한 측면입니다. 적절한 랩 테스트를 통해 운영 다운타임을 크게 줄이고 네트워크 지원 인력을 교육하며 네트워크 구현 프로세스를 간소화할 수 있습니다. 그러나 효과를 거두기 위해서는 적절한 랩 환경을 구축 및 유지 관리하는 데 필요한 리소스를 할당하고, 올바른 테스트를 수행하는 데 필요한 리소스를 적용하고, 측정 수집이 포함된 권장 테스트 방법을 사용해야 합니다. 이러한 영역이 없으면 테스트 및 검증 프로세스가 조직의 기대에 미치지 못할 수 있습니다.

대부분의 기업 조직에는 권장되는 테스트 랩 환경이 없습니다. 이러한 이유로 많은 조직이 솔루션을 잘못 구축했거나 네트워크 변경 장애를 경험했거나 랩 환경에서 격리될 수 있는 소프트웨어 문제를 경험했습니다. 일부 환경에서는 다운타임으로 인한 비용이 정교한 랩 환경의 비용을 상쇄하지 않으므로 이 문제가 해결됩니다. 그러나 많은 조직에서는 다운타임을 허용할 수 없습니다. 이러한 조직은 생산 네트워크 품질을 개선하기 위해 권장되는 테스트 랩, 테스트 유형 및 테스트 방법을 개발해야 합니다.

테스트 랩 및 환경

이 실습은 책상, 워크벤치, 테스트 장비, 장비 캐비닛 또는 랙에 충분한 공간이 있는 격리된 공간이어야 합니다. 대부분의 대규모 조직에서는 생산 환경을 모방하기 위해 4~10개의 장비 랙이 필요합니다. 테스트가 진행되는 동안 테스트 환경을 유지하기 위해 일부 물리적 보안을 사용하는 것이 좋습니다. 이를 통해 하드웨어 차입, 교육 또는 구현 리허설 등 다른 랩 우선 순위로 인해 랩 테스트가 중단되는 것을 방지할 수 있습니다. 또한 논리적 보안은 위조된 경로가 프로덕션 네트워크에 진입하거나 원치 않는 트래픽이 Lab에서 나가는 것을 방지하기 위해 권장됩니다. 랩 게이트웨이 라우터의 라우팅 필터 및 확장 액세스 목록을 사용하여 이 작업을 수행할 수 있습니다. 프로덕션 네트워크에 연결하면 소프트웨어 다운로드 및 프로덕션 환경에서 랩 네트워크에 액세스하는 데 유용합니다.

랩 토폴로지는 특정 테스트 계획에 대한 프로덕션 환경을 모방할 수 있어야 합니다. 하드웨어, 네트워크 토폴로지 및 기능 구성을 재생성하는 것이 좋습니다. 물론 실제 토폴로지를 재현하는 것은 거의 불가능하지만, 가능한 일은 네트워크 계층 구조와 프로덕션 디바이스 간의 상호 작용을 재현하는 것입니다. 이는 여러 디바이스 간의 프로토콜 또는 기능 상호 작용에 중요합니다. 일부 테스트 토폴로지는 소프트웨어 테스트 요구 사항에 따라 다릅니다. 예를 들어 WAN 에지 Cisco IOS 테스트에서는 LAN 유형 디바이스 또는 테스트가 필요하지 않으며 WAN 에지 라우터 및 WAN 배포 라우터만 필요할 수 있습니다. 핵심은 생산을 복제하지 않고 소프트웨어 기능을 모방하는 것입니다. 프로토콜 네이버 수 및 라우팅 테이블과 같은 대규모 동작을 모방하는 데 툴을 사용할 수도 있습니다.

또한 프로덕션 환경을 모방하고 테스트 데이터를 수집할 수 있는 기능을 개선하여 일부 테스트 유형을 지원하는 툴도 필요합니다. 프로덕션을 모방하는 툴로는 트래픽 컬렉터, 트래픽 생성기, WAN 시뮬레이션 디바이스가 있습니다. Smartbits는 네트워크 트래픽을 수집 및 재생하거나 대량의 트래픽을 생성할 수 있는 디바이스의 좋은 예입니다. 조직에서는 프로토콜 분석기와 같은 데이터를 수집하는 데 도움이 되는 장치도 활용할 수 있습니다.

또한 Lab을 관리하려면 일부 관리가 필요합니다. 많은 대규모 조직에는 랩 네트워크 관리 업무를 담당하는 정규 랩 관리자가 있습니다. 다른 조직에서는 랩 검증을 위해 기존 아키텍처 및 엔지니어링 팀을 활용합니다. 랩 관리 업무에는 랩 장비 및 자산 추적, 케이블 연결, 물리적 공간 관리, 랩 규칙 및 방향 정의, 랩 스케줄링, 랩 설명서, 랩 토폴로지 설정, 테스트 계획 작성, 랩 테스트 수행, 확인된 잠재적 문제 관리 등이 포함됩니다.

테스트 유형

전반적으로 여러 가지 유형의 테스트를 수행할 수 있습니다. 다양한 구성으로 모든 것을 테스트할 수 있는 완벽한 테스트 랩 및 테스트 계획을 구축하기 전에, 조직은 다양한 테스트 유형, 테스트 목적, Cisco 엔지니어링, 기술 마케팅 또는 고객 지원 팀이 다양한 테스트의 일부를 담당해야 하는지 또는 담당해야 하는지 여부를 이해해야 합니다. 고객 테스트 계획에는 일반적으로 노출되는 테스트 유형이 포함됩니다. 다음 표에서는 다양한 테스트 유형, 테스트 수행 시기 및 책임 당사자를 이해하는 데 도움이 됩니다.

아래 테스트 중 조직의 특정 기능 집합, 토폴로지 및 애플리케이션 혼합을 올바르게 테스트하는 것이 일반적으로 가장 중요합니다. Cisco는 완전한 기능 및 회귀 테스트를 수행하지만 Cisco는 토폴로지, 하드웨어 및 구성된 기능의 특정 조합으로 조직의 애플리케이션 프로필을 테스트할 수 없습니다. 실제로, 모든 기능, 하드웨어, 모듈 및 토폴로지 순열을 테스트하는 것은 불가능합니다. 또한 Cisco는 타사 장비와의 상호운용성을 테스트할 수 없습니다. Cisco는 조직에서 해당 환경에 있는 하드웨어, 모듈, 기능 및 토폴로지의 정확한 조합을 테스트할 것을 권장합니다. 이 테스트는 랩에서 수행되어야 합니다. 축소된 토폴로지는 조직의 운영 환경을 나타내며 성능, 상호 운용성, 중단, 번인 등의 다른 지원 테스트 유형을 나타냅니다.

테스트	테스트 개요	테스트 책임
기능	기본 Cisco IOS 기능 및 Cisco 하드웨어 모듈이 광고된 대로 작동하는지 확인합니다. 기능 또는 모듈 기능과 기능 구성 옵션을 테스트해야 합니다. 구성 제거 및 추가를 테스트해야 합니다. 기본 중단 테스트 및 번인 테스트가 포함되어 있습니다.	Cisco 장치 테스트
회귀	기능 또는 모듈이 다른 모듈 및 기능과 함께 작동하는지, 그리고 Cisco IOS 버전이 정의된 기능과 관련하여 다른 Cisco IOS 버전과 함께 작동하는지 확인합니다. 일부 번인 및 중단 테스트를 포함합니다.	Cisco 회귀 테스트

기본 디바이스 성능	Cisco IOS 기능 또는 하드웨어 모듈이 로드 중인 최소 요구 사항을 충족하는지 확인하기 위해 기능 또는 모듈의 기본 성능을 결정합니다.	Cisco 디바이스 테스트
토폴로지/기능/하드웨어 조합	기능과 모듈이 특정 토폴로지 및 모듈/기능/하드웨어 조합에서 예상대로 작동하는지 확인합니다. 이 테스트에는 프로토콜 확인, 기능 확인, **show** 명령 확인, 번인 테스트 및 중단 테스트가 포함되어야 합니다.	Cisco는 ESE(Enterprise Solutions Engineering) 및 NSITE(Networked Solutions Integration Test Engineering)와 같은 랩에서 표준 보급 토폴로지를 테스트합니다. 고가용성 고객은 필요에 따라 기능/모듈/토폴로지 조합을 테스트해야 합니다. 특히 조기 도입 소프트웨어 및 비표준 토폴로지를 사용해야 합니다.
중단(What-if)	특정 기능/모듈/토폴로지 환경에서 발생할 수 있는 일반적인 중단 유형 또는 동작 및 잠재적인 기능 영향을 포함합니다. 중단 테스트에는 카드 스와핑, 링크 플랩, 디바이스 장애, 링크 장애, 카드 장애 등이 포함됩니다.	Cisco는 기본적인 운영 중단 테스트를 담당합니다. 고객은 궁극적으로 개별 환경의 확장성과 관련된 중단 성능 문제를 책임집니다. 가능한 경우 고객 랩 환경에서 중단 테스트를 수행해야 합니다.
NetworkPerformance(What-if)	특정 기능/하드웨어/토폴로지 조합과 관련하여 디바이스 로드를 조사합니다. 프로토콜, 네이버, 경로 수 및 기타 기능에 대한 설정된 트래픽 유형 및 리소스 요구 사항과 관련하	고객은 궁극적으로 디바이스 로드 및 확장성을 책임집니다. 로드 및 확장성 문제는 Cisco 세일즈 또는 Advanced Services에서 종종 제기되며 CPOC(Customer Proof-of-Concept Labs)와 같은 Cisco 랩에서 테스트되는 경우가 많습니다.

	여 CPU, 메모리, 버퍼 사용률, 링크 사용률 등의 디바이스 용량 및 성능에 중점을 둡니다. 이 테스트는 대규모 환경에서 확장성을 보장하는 데 도움이 됩니다.	
버그 수정	버그가 확인된 결함을 복구하도록 합니다.	Cisco는 버그가 고정되었는지 확인하기 위해 버그 수정을 테스트합니다. 또한 고객은 자신이 경험한 버그가 고정되어 있고 버그가 모듈이나 기능의 다른 측면을 방해하지 않는지 테스트해야 합니다. 유지 보수 릴리스는 회귀 테스트를 수행하지만 중간 릴리스는 대개 그렇지 않습니다.
네트워크 관리	SNMP(Simple Network Management Protocol) 관리 기능, SNMP MIB 변수 정확성, 트랩 지원 및 Syslog 지원을 조사합니다.	Cisco는 기본 SNMP 기능 및 MIB 변수 정확성 테스트를 담당합니다. 고객은 네트워크 관리 결과를 검증해야 하며, 궁극적으로 새로운 기술 구축을 위한 관리 전략 및 방법론을 담당해야 합니다.
대규모 네트워크 에뮬레이션	대규모 네트워크 에뮬레이션은 Agilent의 라우터 시뮬레이터 및 Spirent의 테스트 툴 세트와 같은 툴을 사용하여 대규모 환경을 시뮬레이션합니다. 여기에는 프로토콜 네이버, PVC(Frame-Relay Permanent Virtual Circuit) 수, 라우팅 테이블 크기, 캐시 항목 및 기	Cisco 고객은 일반적으로 네트워크 환경을 재생성하는 네트워크 시뮬레이션 테스트의 측면을 담당하며, 여기에는 라우팅 프로토콜 네이버/인접성, 관련 라우팅 테이블 크기 및 프로덕션 중인 기타 리소스가 포함될 수 있습니다.

	본적으로 Lab에 없는 프로덕션 환경에 일반적으로 필요한 기타 리소스가 포함될 수 있습니다.	
상호 운용성	특히 프로토콜 또는 신호 상호 운용성이 필요한 경우 서드파티 네트워크 장비에 대한 연결과 관련된 모든 요소를 테스트합니다.	Cisco 고객은 일반적으로 상호 운용성 테스트의 모든 측면을 책임집니다.
번인	시간이 지남에 따라 라우터 리소스를 조사합니다. 번인 테스트에서는 일반적으로 시간이 지남에 따라 메모리, CPU 및 버퍼를 비롯한 리소스 사용률을 조사하여 디바이스가 일부 로드 상태에 있어야 합니다.	Cisco는 기본적인 번인 테스트를 수행합니다. 고유한 토폴로지, 디바이스 및 기능 조합과 관련하여 고객 테스트가 권장됩니다.

테스트 방법론

어떤 조직이 어떤 테스트를 수행하는지 알게 되면 테스트 프로세스를 위한 방법을 개발해야 합니다. 모범 사례 테스트 방법론의 목적은 합의된 테스트가 잠재적인 생산 문제를 발견하는 데 있어서 종합적이고, 문서화되고, 쉽게 재현할 수 있으며, 가치가 있는지 확인하는 것입니다. 실습 시나리오의 문서화 및 재생성은 실습 환경에서 발견된 버그 수정 사항을 테스트하거나 최신 버전을 테스트하는 데 특히 중요합니다. 테스트 방법론의 단계는 아래와 같습니다. 일부 테스트 단계는 동시에 수행할 수도 있습니다.

1. 테스트 중인 프로덕션 환경을 시뮬레이션하는 테스트 토폴로지를 만듭니다. WAN 에지 테스트 환경에는 몇 개의 코어 라우터와 하나의 에지 라우터만 포함될 수 있으며, LAN 테스트에는 환경을 가장 잘 나타낼 수 있는 더 많은 장치가 포함될 수 있습니다.
2. 프로덕션 환경을 시뮬레이션하는 기능을 구성합니다. 랩 디바이스의 컨피그레이션은 예상 프로덕션 디바이스 하드웨어 및 소프트웨어 컨피그레이션과 밀접하게 일치해야 합니다.
3. 테스트 계획 작성, 테스트 및 목표 정의, 토폴로지 문서화, 기능 테스트 정의 테스트에는 기본 프로토콜 검증, show 명령 검증, 중단 테스트 및 번인 테스트가 포함됩니다. 테스트 계획 내의 특정 테스트의 예는 다음 표에 나와 있습니다.
4. 라우팅 및 프로토콜 기능을 검증합니다. 문서 또는 베이스라인에 **show** 명령 결과가 필요합니다. 프로토콜에는 ATM, Frame-Relay, Cisco Discovery Protocol(CDP), Ethernet 및 Spanning-Tree와 같은 레이어 2 프로토콜과 IP, IPX 및 멀티캐스트와 같은 레이어 3 프로토콜이 모두 포

함되어야 합니다.

5. 기능 기능을 검증합니다. 문서 또는 베이스라인에 **show** 명령 결과가 필요합니다. 기능에는 전역 컨피그레이션 명령 및 AAA(Authentication, Authorization, and Accounting)와 같은 중요한 기능이 포함될 수 있습니다.
6. 프로덕션 환경에서 예상되는 로드 시뮬레이트 로드 시뮬레이션은 트래픽 컬렉터/생성기로 수행할 수 있습니다. CPU, 메모리, 버퍼 사용률, 인터페이스 통계 등 예상 네트워크 디바이스 사용률 변수를 패킷 손실에 대한 조사를 통해 검증합니다. 문서 또는 베이스라인에 **show** 명령 결과가 필요합니다.
7. 장치 및 소프트웨어가 부하를 처리하거나 방지할 것으로 예상되는 곳에서 중단 테스트를 수행합니다. 예를 들어 카드 제거, 링크 플랩, 경로 플랩, 브로드캐스트 스톰 등이 있습니다. 네트워크 내에서 사용 중인 기능에 따라 올바른 SNMP 트랩이 생성되는지 확인합니다.
8. 테스트 결과 및 디바이스 측정은 반복 가능해야 하므로 문서화합니다.

테스트 이름	**HSRP(Hot Standby Router Protocol) 장애 조치**
테스트 구성 요구 사항	기본 게이트웨이 인터페이스에 로드를 적용합니다. 사용자 스테이션의 관점에서 트래픽은 게이트웨이의 약 20%, 사용자 스테이션의 관점에서 트래픽의 60%가 수신되어야 합니다. 또한 트래픽을 더 높은 로드로 증가시킵니다.
테스트 단계	**show** 명령을 통해 STP 및 HSRP를 모니터링합니다. 기본 게이트웨이 인터페이스 연결에 실패한 다음 정보를 수집한 후 연결을 복구합니다.
예상 측정	장애 조치 중 CPU. 기본 및 보조 게이트웨이의 이전, 중, 후 인터페이스를 표시합니다. HSRP를 이전, 중, 후 전에 표시합니다.
예상 결과	기본 게이트웨이가 2초 내에 다른 라우터 게이트웨이로 장애 조치됩니다. **show 명령**은 변경 사항을 제대로 반영합니다. 연결이 복원될 때 기본 게이트웨이에 대한 장애 조치가 발생합니다.
실제 결과	
통과 또는 실패	
통과에 필요한 수정 사항	

디바이스 측정

테스트 단계에서 다음 측정치를 수행 및 문서화하여 디바이스가 올바르게 작동하는지 확인합니다.

- 메모리 사용량
- CPU 로드
- 버퍼 사용량

- 인터페이스 통계
- 경로 테이블
- 특정 디버깅

측정 정보는 구현 중인 특정 테스트에 따라 달라집니다. 해결 중인 특정 문제에 따라 측정을 위한 추가 정보가 있을 수도 있습니다.

테스트 중인 각 응용 프로그램에 대해 매개 변수를 측정하여 지정된 응용 프로그램에 부정적인 성능이 영향을 미치지 않도록 합니다. 이 작업은 구축 전 및 후 성능을 비교하는 데 사용할 수 있는 성능 기준을 활용하여 완료됩니다. 애플리케이션 측정 테스트의 예는 다음과 같습니다.

- 네트워크에 로그온하는 데 걸리는 평균 시간입니다.
- NFS(Network File System)에서 파일 그룹을 복사하는 데 걸리는 평균 시간입니다.
- 응용 프로그램을 시작하고 첫 번째 화면으로 메시지가 표시되는 데 걸리는 평균 시간입니다.
- 기타 애플리케이션별 매개변수

구현 - 신속하고 성공적인 Cisco IOS 구축

잘 정의된 구현 프로세스를 통해 조직은 새로운 Cisco IOS 버전을 효율적으로 구축할 수 있습니다.

구현 단계에는 파일럿 프로세스 및 구현 프로세스가 포함됩니다. 파일럿 프로세스를 통해 Cisco IOS 버전이 환경에서 성공하고 구현 프로세스를 통해 Cisco IOS 구축 규모가 빠르고 성공적으로 확대됩니다.

Cisco IOS 구축을 위한 전략 및 툴

Cisco IOS 구축을 위한 전략은 파일럿 프로세스를 통해 최종 인증을 수행하고 업그레이드 툴과 잘 정의된 구현 프로세스를 사용하여 신속하게 구축하는 것입니다.

네트워크 파일럿 프로세스를 시작하기 전에 많은 조직이 일반적인 파일럿 지침을 작성합니다. 파일럿 지침에는 성공 기준, 허용되는 파일럿 위치, 파일럿 문서, 파일럿 소유자 기대치, 사용자 알림 요구 사항, 예상 파일럿 기간과 같은 모든 파일럿에 대한 기대치가 포함되어야 합니다. 엔지니어링, 구현 및 운영 팀의 부서간 팀은 일반적으로 전체 파일럿 지침 및 파일럿 프로세스를 구축하는 데 관여합니다. 파일럿 프로세스가 생성되면 개별 구현 그룹은 일반적으로 식별된 모범 사례 방법을 사용하여 성공적인 파일럿을 수행할 수 있습니다.

새로운 소프트웨어 버전이 배포 및 최종 인증을 승인되면 조직은 Cisco IOS 업그레이드 계획을 시작해야 합니다. Planning은 플랫폼, 메모리, 플래시, 컨피그레이션을 비롯한 새로운 이미지 요구 사항을 파악하는 것으로 시작합니다. 아키텍처 및 엔지니어링 그룹은 일반적으로 Cisco IOS 관리 라이프사이클의 후보 관리 단계에서 새로운 소프트웨어 이미지 요구 사항을 정의합니다. 요구 사항이 확인되면 각 디바이스를 구현 그룹에 의해 검증 및 업그레이드해야 합니다. CiscoWorks2000 SWIM(Software Image Manager) 모듈은 디바이스 인벤토리에 대한 Cisco IOS 요구 사항을 검증하여 검증 단계를 수행할 수도 있습니다. 모든 디바이스가 올바른 새 이미지 표준으로 검증되거나 업그레이드되면 구현 그룹은 CiscoWorks2000 SWIM 모듈을 소프트웨어 구축 툴로 사용하여 느린 시작 구현 프로세스를 시작할 수 있습니다.

새 이미지가 여러 번 성공적으로 구축되면 조직은 CiscoWorks SWIM을 사용하여 빠른 구축을 시작할 수 있습니다.

Cisco IOS 인벤토리 관리

CiscoWorks2000 RME(Resource Manager Essentials) Inventory Manager는 소프트웨어 버전, 장치 플랫폼 및 장치 이름을 기반으로 Cisco IOS 장치를 보고하고 정렬하는 웹 기반 보고 도구를 통해 Cisco 라우터 및 스위치의 버전 관리를 크게 간소화합니다.

Cisco IOS SWIM

CiscoWorks2000 SWIM은 업그레이드 프로세스의 오류 발생 가능성이 높은 복잡성을 줄이는 데 도움이 됩니다. CCO에 대한 기본 제공 링크에서는 소프트웨어 패치에 대한 Cisco 온라인 정보를 네트워크에 구축된 Cisco IOS 및 Catalyst 소프트웨어와 상호 연결하여 관련 기술 정보를 강조합니다. 새로운 계획 툴은 제안된 소프트웨어 이미지 업데이트를 지원하기 위해 하드웨어 업그레이드(Boot ROM, Flash RAM)가 필요할 때 시스템 요구 사항을 찾고 알림을 전송합니다.

업데이트가 시작되기 전에, 성공적인 업그레이드를 위해 타겟 스위치 또는 라우터의 인벤토리 데이터에 대해 새 이미지의 사전 요구 사항이 검증됩니다. 여러 디바이스가 업데이트되는 경우 SWIM은 다운로드 작업을 동기화하고 사용자가 작업의 진행 상황을 모니터링할 수 있도록 합니다. 예약된 작업은 사인오프 프로세스를 통해 제어되므로 관리자는 각 업그레이드 작업을 시작하기 전에 기술자 활동을 승인할 수 있습니다. RME 3.3에는 Cisco IGX, BPX 및 MGX 플랫폼의 소프트웨어 업그레이드를 분석할 수 있는 기능이 포함되어 있어 소프트웨어 업그레이드의 영향을 결정하는 데 필요한 시간을 대폭 간소화하고 줄일 수 있습니다.

파일럿 프로세스

잠재적인 노출을 최소화하고 나머지 생산 문제를 더 안전하게 포착하려면 소프트웨어 파일럿을 사용하는 것이 좋습니다. 파일럿은 일반적으로 새로운 기술 구축에서 더 중요하지만, 파일럿이 더 중요한 새로운 서비스, 기능 또는 하드웨어와 연결되는 새로운 소프트웨어 구축이 많습니다. 개별 파일럿 계획은 파일럿 선택, 파일럿 기간 및 측정을 고려해야 합니다. 파일럿 선택은 파일럿이 언제 어디서 수행되어야 하는지를 확인하는 프로세스입니다. 파일럿 측정은 성공 및 실패 또는 잠재적 문제를 식별하기 위해 필요한 데이터를 수집하는 프로세스입니다.

파일럿 선택은 파일럿이 완료되는 위치와 방법을 식별합니다. 파일럿은 영향이 적은 영역에서 하나의 디바이스로 시작하여 영향이 큰 영역에서 여러 디바이스로 확장할 수 있습니다. 영향을 줄일 수 있는 파일럿 선택의 몇 가지 고려 사항은 다음과 같습니다.

- 이중화로 인한 단일 장치에 영향을 줄 수 있는 네트워크 영역에 설치됩니다.
- 선택한 디바이스 뒤에 최소 수의 사용자가 있는 네트워크 영역에서 프로덕션 영향을 처리할 수 있습니다.
- 아키텍처 라인을 따라 파일럿 분리를 고려하십시오. 예를 들어, 네트워크의 액세스, 배포 및/또는 코어 레이어에서 파일럿 작업을 수행합니다.

이 파일럿의 기간은 모든 장치 기능을 충분히 테스트하고 평가하는 데 걸리는 시간을 기준으로 해야 합니다. 여기에는 정상적인 트래픽 로드 시 번인 및 네트워크가 모두 포함되어야 합니다. 기간은 코드 업그레이드의 단계 및 Cisco IOS가 실행 중인 네트워크 영역에 따라 달라집니다. Cisco IOS가 새로운 주요 릴리스인 경우 파일럿 기간이 길어지는 것이 좋습니다. 업그레이드가 새로운 기능을 최소화한 유지 보수 릴리스인 경우 파일럿 기간을 단축하면 충분합니다.

파일럿 단계에서는 초기 테스트와 유사한 방식으로 결과를 모니터링하고 문서화하는 것이 중요합니다. 여기에는 사용자 설문조사, 파일럿 데이터 수집, 문제 수집, 성공/실패 기준이 포함될 수 있습니다. 개인은 모든 문제가 식별되고 파일럿과 관련된 사용자 및 서비스가 파일럿 결과에 만족하는지 확인하기 위해 파일럿 진행 상황을 추적하고 모니터링하는 업무를 직접 담당해야 합니다. 파일럿 또는 생산 환경에서 성공적으로 릴리스될 경우 대부분의 조직에서 해당 릴리스를 인증합니다. 이 단계는 측정 또는 성공 기준을 식별하거나 문서화하지 않은 경우 성공을 인지하는 일부 환경에서 매우 중요한 실패입니다.

구현

프로덕션 네트워크 내에서 파일럿 단계가 완료되면 Cisco IOS 구현 단계를 시작합니다. 구현 단계에는 구현 느린 시작, 최종 인증, 업그레이드 준비, 업그레이드 자동화, 최종 검증 등 소프트웨어 업그레이드 성공 및 구현 효율성을 보장하기 위한 몇 가지 단계가 포함되어 있습니다.

구현 느린 시작은 최종 인증 및 전체 규모 변환 전에 이미지가 생산 환경에 완전히 노출되도록 새로 테스트된 릴리스를 천천히 구현하는 프로세스입니다. 일부 조직은 하루 1개 장치에서 하루 노출되는 것으로 시작하여 다음 날 2개의 장치로 업그레이드하고 그 다음 날 몇 개 더 많은 장치로 이동할 수 있습니다. 약 10개의 디바이스가 프로덕션 환경에 배치되면 조직은 특정 Cisco IOS 버전의 최종 인증을 받기 전까지 최대 1주에서 2주 정도 기다릴 수 있습니다. 최종 인증 후, 조직은 훨씬 높은 신뢰도로 식별된 버전을 더 신속하게 배포할 수 있습니다.

느린 시작 프로세스 후, 디바이스 인벤토리와 부트스트랩, DRAM 및 플래시에 대한 최소 Cisco IOS 표준 매트릭스를 사용하여 업그레이드를 위해 식별된 모든 디바이스를 검토 및 검증하여 요구 사항을 충족해야 합니다. 사내 툴, 타사 SNMP 툴 또는 CiscoWorks2000 RME를 사용하여 데이터를 얻을 수 있습니다. CiscoWorks2000 SWIM은 구현 전에 이러한 변수를 검토하거나 검사합니다. 그러나 구현 시도 중에 무엇을 기대할지 아는 것은 항상 좋은 생각입니다.

100개가 넘는 유사 장치를 업그레이드 예약된 경우에는 자동화된 방법을 사용하는 것이 좋습니다. SWIM을 사용하거나 사용하지 않는 1,000개의 장치를 내부 업그레이드한 것을 기반으로, 자동화는 업그레이드 효율성을 개선하고 대규모 구축 시 장치 업그레이드 성공 비율을 개선하는 것으로 나타났습니다. 업그레이드 중에 수행되는 검증 정도에 따라 대규모 구축에 CiscoWorks 2000 SWIM을 사용하는 것이 좋습니다. 문제가 탐지되면 SWIM은 Cisco IOS 버전에서도 다시 실행됩니다. SWIM은 업그레이드 작업을 생성하고 예약함으로써 작동합니다. 여기서 작업은 디바이스로 구성되고, 원하는 업그레이드 이미지 및 작업 실행 시간이 설정됩니다. 각 작업에는 12개 이하의 장치 업그레이드가 포함되어야 하며, 최대 12개의 작업이 동시에 실행될 수 있습니다. 또한 SWIM은 예약된 Cisco IOS 업그레이드 버전이 업그레이드 후 성공적으로 실행 중인지 확인합니다. 각 디바이스 업그레이드(확인 포함)에 대해 약 20분을 허용하는 것이 좋습니다. 이 공식을 사용하여 조직은 시간당 36개의 장치를 업그레이드할 수 있습니다. 또한 Cisco는 잠재적인 문제 노출을 줄이기 위해 저녁 1회 최대 100대의 장치를 업그레이드할 것을 권장합니다.

자동 업그레이드 후에는 성공을 보장하기 위해 일부 검증을 수행해야 합니다. CiscoWorks2000 SWIM 툴은 업그레이드 후 맞춤형 스크립트를 실행하여 추가 성공 확인을 수행할 수 있습니다. 검증에는 라우터에 적절한 수의 경로가 있는지 검증하고, 논리적/물리적 인터페이스가 작동 및 활성 상태인지 확인하거나, 디바이스에 액세스할 수 있는지 확인하는 작업이 포함됩니다. 다음 샘플 체크리스트는 Cisco IOS 구축의 성공을 완벽하게 검증할 수 있습니다.

- 장치가 올바르게 다시 로드되었습니까?
- 디바이스가 NMS(Network Management System) 플랫폼을 통해 ping할 수 있으며 연결할 수 있습니까?
- 디바이스의 예상 인터페이스가 작동 및 활성 상태입니까?
- 디바이스에 올바른 라우팅 프로토콜 인접성이 있습니까?
- 라우팅 테이블이 채워져 있습니까?
- 디바이스가 트래픽을 올바르게 전달합니까?

운영 - 고가용성 관리 Cisco IOS 구현

Cisco IOS 환경의 고가용성 모범 사례 운영을 통해 네트워크 복잡성을 줄이고 문제 해결 시간을 단축하며 네트워크 가용성을 높일 수 있습니다. Cisco IOS 관리의 운영 섹션에는 Cisco IOS 관리에 권장되는 전략, 툴 및 모범 사례 방법론이 포함되어 있습니다.

Cisco IOS 운영 모범 사례에는 소프트웨어 버전 제어, Cisco IOS Syslog 관리, 문제 관리, 구성 표준화, 가용성 관리 등이 있습니다. 소프트웨어 버전 제어는 식별된 소프트웨어 트랙 내에서 소프트웨어 일관성을 추적, 검증 및 개선하는 프로세스입니다. Cisco IOS Syslog 관리는 Cisco IOS에서 생성한 우선순위가 높은 Syslog 메시지를 사전 대응적으로 모니터링하고 이에 따라 작동하는 프로세스입니다. 문제 관리는 소프트웨어 관련 문제에 대한 중요한 문제 정보를 신속하고 효율적으로 수집하여 향후 발생하는 문제를 방지하는 것입니다. 구성 표준화는 구성을 표준화하여 테스트되지 않은 코드가 프로덕션 환경에서 실행될 가능성을 줄이고 네트워크 프로토콜 및 기능 동작을 표준화하는 프로세스입니다. 가용성 관리는 메트릭, 개선 목표 및 개선 프로젝트에 따라 가용성을 향상시키는 프로세스입니다.

Cisco IOS 운영을 위한 전략 및 툴

Cisco IOS 환경을 관리하는 데 도움이 되는 다양한 품질 전략 및 툴이 있습니다. Cisco IOS 운영을 위한 첫 번째 핵심 전략은 가능한 한 환경을 간소화하여 구성과 Cisco IOS 버전의 변화를 최대한 방지하는 것입니다. Cisco IOS 인증은 이미 논의되었지만 구성 일관성은 또 다른 핵심 영역입니다. 아키텍처/엔지니어링 그룹은 컨피그레이션 표준을 만드는 책임을 져야 합니다. 구현 및 운영 그룹은 Cisco IOS 버전 제어 및 구성 표준/제어를 통해 표준을 구성하고 유지 관리할 책임이 있습니다.

Cisco IOS 운영을 위한 두 번째 전략은 네트워크 결함을 식별하고 신속하게 해결할 수 있는 기능입니다. 네트워크 문제는 일반적으로 사용자가 전화를 걸기 전에 작업 그룹에서 식별해야 합니다. 또한 환경에 더 큰 영향을 주거나 변경 없이 가능한 한 빨리 문제를 해결해야 합니다. 이 영역에서 몇 가지 주요 모범 사례는 문제 관리 및 Cisco IOS Syslog 관리입니다. Cisco IOS 소프트웨어 충돌을 신속하게 진단하는 데 도움이 되는 툴은 Cisco Output Interpreter입니다.

세 번째 전략은 일관된 개선입니다. 기본 프로세스는 품질 기반 가용성 개선 프로그램을 개선하는 것입니다. 조직은 Cisco IOS 관련 문제를 비롯한 모든 문제에 대한 근본 원인 분석을 수행하여 테스트 범위를 개선하고 문제 해결 시간을 단축하며 운영 중단 영향을 제거하거나 줄이는 프로세스를 개선할 수 있습니다. 또한 일반적인 문제를 살펴보고 프로세스를 구축하여 문제를 더 빨리 해결할 수 있습니다.

Cisco IOS 운영 도구에는 소프트웨어 버전 제어를 위한 인벤토리 관리(CiscoWorks2000 RME), Syslog 메시지를 관리하는 Syslog 관리, 디바이스 컨피그레이션 일관성을 관리하기 위한 디바이스 컨피그레이션 관리자가 포함됩니다.

Syslog 관리

Syslog 메시지는 디바이스에서 수집 서버로 전송하는 메시지입니다. 이러한 메시지는 오류(예: 링크가 다운됨)일 수도 있고, 누군가가 디바이스에서 터미널을 구성하기 위해 로그인한 경우 등의 정보를 제공할 수도 있습니다.

Syslog 관리 툴은 라우터 및 스위치에서 수신한 Syslog 메시지를 기록하고 추적합니다. 일부 도구에는 중요한 메시지를 손상시킬 수 있는 원치 않는 메시지를 제거할 수 있는 필터가 있습니다. 또한 Syslog 툴을 사용하면 수신된 메시지에 따라 보고를 생성할 수 있습니다. 보고 기능은 기간, 디바이스, 메시지 유형 또는 메시지 우선 순위별로 표시할 수 있습니다.

Cisco IOS 관리를 위한 가장 인기 있는 Syslog 툴은 CiscoWorks2000 RME Syslog 관리자입니다. SL4NT, Net의 공유 프로그램 및 OpenSystems의 Private I를 비롯한 다른 툴도 사용할 수 있습니다.

CiscoWorks 장치 구성 관리자

CiscoWorks2000 Device Configuration Manager는 액티브 아카이브를 유지 관리하며 여러 Cisco 라우터 및 스위치에서 구성 변경 사항을 쉽게 업데이트할 수 있는 방법을 제공합니다. 컨피그레이

션 관리자는 네트워크를 모니터링하여 컨피그레이션 변경 사항을 감지하면 아카이브를 업데이트하고 변경 감사 서비스에 변경 정보를 기록합니다. 웹 기반 사용자 인터페이스를 사용하면 아카이브를 검색하여 특정 구성 속성을 검색하고 두 구성 파일의 내용을 비교하여 차이점을 쉽게 식별할 수 있습니다.

Cisco 출력 인터프리터

Cisco Output Interpreter는 소프트웨어 강제 충돌을 진단하는 데 사용되는 툴입니다. 이 툴은 Cisco TAC(Technical Assistance Center)에 문의하지 않고 소프트웨어 결함을 식별하는 데 도움이 될 수 있으며, 소프트웨어 강제 충돌 후 TAC에 대한 기본 정보로 사용될 수도 있습니다. 이 정보는 일반적으로 최소한 필수 정보 수집의 측면에서 문제를 신속하게 해결하는 데 도움이 됩니다.

소프트웨어 버전 제어

소프트웨어 버전 제어는 표준화된 소프트웨어 버전만 구현하고 비버전 규정 준수로 인해 소프트웨어를 검증하거나 변경할 수 있도록 네트워크를 모니터링하는 프로세스입니다. 일반적으로 소프트웨어 버전 제어는 인증 프로세스 및 표준 제어를 사용하여 수행됩니다. 많은 조직이 중앙 웹 서버에 버전 표준을 게시합니다. 또한 구현 직원은 실행 중인 버전을 검토하고 표준을 준수하지 않을 경우 버전을 업데이트하도록 교육을 받습니다. 일부 조직에서는 감사를 통해 2차 검증을 완료하여 구현 중에 표준을 준수하도록 하는 품질 게이트 프로세스가 있습니다.

운영 중에 네트워크에서 비표준 버전이 나타나는 경우가 종종 있습니다. 특히 네트워크 및 운영 직원이 많은 경우 더욱 그렇습니다. 이는 교육을 받지 않은 새로운 직원, 잘못 구성된 boot 명령 또는 선택되지 않은 구현 때문일 수 있습니다. 모든 장치를 Cisco IOS 버전별로 정렬할 수 있는 CiscoWorks 2000 RME와 같은 툴을 사용하여 소프트웨어 버전 표준을 주기적으로 검증하는 것이 좋습니다. 비표준이 식별되면 즉시 플래그를 지정하고 해당 버전을 식별된 표준으로 가져오기 위해 문제 티켓 또는 변경 티켓을 시작해야 합니다.

사전 대응적 Syslog 관리

Syslog 수집, 모니터링 및 분석은 다른 방법으로 식별하기 어렵거나 불가능한 더 많은 Cisco IOS 특정 네트워크 문제를 해결하기 위해 권장되는 장애 관리 프로세스입니다. Syslog 수집, 모니터링 및 분석은 더 심각한 네트워크 문제가 발생하거나 사용자가 보고하기 전에 사전에 많은 결함을 식별하고 해결하여 문제 해결 시간을 단축하는 데 도움이 됩니다. 또한 Syslog는 다수의 MIB 변수에 대한 일관된 SNMP 폴링과 비교하여 다양한 문제를 수집하는 더 효율적인 방법을 제공합니다. Syslog 수집, 모니터링 및 분석은 올바른 Cisco IOS 구성, CiscoWorks2000 RME 및/또는 Syslog 이벤트 관리와 같은 Syslog 상관관계 툴을 활용하여 수행됩니다. Syslog 이벤트 관리는 확인된 중요한 메시지에 대해 수집된 Syslog 데이터를 구문 분석한 다음 실시간 알림 및 해결을 위해 이벤트 관리자에게 알림 또는 트랩을 전달하여 수행됩니다.

Syslog 모니터링에서는 Syslog 데이터의 구문 분석 및 보고를 지원하기 위해 NMS 툴 지원 또는 스크립트가 필요합니다. 여기에는 날짜 또는 시간, 디바이스, Syslog 메시지 유형 또는 메시지 빈도별로 Syslog 메시지를 정렬하는 기능이 포함됩니다. 대규모 네트워크에서는 Syslog 데이터를 구문 분석하고 이벤트 관리 시스템이나 운영 및 엔지니어링 담당자에게 알림 또는 알림을 보내기 위해 툴 또는 스크립트를 구현할 수 있습니다. 다양한 Syslog 데이터에 대한 알림을 사용하지 않을 경우, 조직은 우선 순위가 높은 Syslog 데이터를 적어도 매일 검토하고 잠재적인 문제에 대한 문제 티켓을 생성해야 합니다. 정상적인 모니터링을 통해 볼 수 없는 네트워크 문제를 사전에 탐지하려면, 즉각적인 문제를 나타내지 않지만 서비스에 영향을 미치기 전에 문제를 나타내는 표시를 제공할 수 있는 상황을 탐지하기 위해 기간별 Syslog 데이터를 정기적으로 검토하고 분석해야 합니다.

문제 관리

많은 고객이 문제 관리 프로세스의 부족으로 인해 추가적인 다운타임을 경험하고 있습니다. 네트워크 관리자가 문제 식별, 정보 수집 및 잘 분석된 솔루션 경로에 시간을 허비하지 않고 서비스에 영향을 주는 명령 또는 구성 변경 사항의 조합을 사용하여 문제를 신속하게 해결하려고 할 때 추가적인 다운타임이 발생할 수 있습니다. 이 영역에서 관찰된 동작에는 문제 및 근본 원인을 조사하기 전에 디바이스를 다시 로드하거나 IP 라우팅 테이블을 지우는 작업이 포함됩니다. 경우에 따라 이는 1단계 지원 문제 해결 목표 때문에 발생합니다. 모든 소프트웨어 관련 문제의 목표는 연결 또는 서비스를 복원하기 전에 근본 원인 분석에 필요한 정보를 신속하게 수집하는 것입니다.

대규모 환경에서는 문제 관리 프로세스가 권장됩니다. 이 프로세스에는 특정 수준의 기본 문제 설명과 두 번째 계층으로 에스컬레이션하기 전에 **show** 명령 모음이 포함되어야 합니다. 첫 번째 계층 지원은 경로를 지우거나 디바이스를 다시 로드하는 것이어서는 안 됩니다. 가장 좋은 방법은 1단계 조직에서 신속하게 정보를 수집하고 2차 계층으로 에스컬레이션하는 것입니다. 초기에 문제 식별 또는 문제 설명에 단 몇 분만 더 시간을 할애하면 근본 원인 검색 가능성이 훨씬 높아지므로 해결 방법, 랩 식별 및 버그 보고 기능을 사용할 수 있습니다. 2차 레벨 지원은 Cisco가 문제를 진단하거나 버그 보고서를 제출하는 데 필요한 정보 유형에 대해 잘 알고 있어야 합니다. 여기에는 메모리 덤프, 라우팅 정보 출력 및 device **show** 명령 출력이 포함됩니다.

구성 표준화

글로벌 디바이스 컨피그레이션 표준은 동일한 디바이스 및 서비스 전반에 걸쳐 표준 글로벌 컨피그레이션 매개변수를 유지 관리하여 전사적 글로벌 컨피그레이션 일관성을 유지하는 방식을 나타냅니다. 전역 컨피그레이션 명령은 개별 포트, 프로토콜 또는 인터페이스가 아닌 전체 디바이스에 적용되는 명령입니다. 글로벌 컨피그레이션 명령은 일반적으로 디바이스 액세스, 일반 디바이스 동작 및 디바이스 보안에 영향을 미칩니다. Cisco IOS에서는 서비스 명령, IP 명령, vty 명령, 콘솔 포트 명령, 로깅 명령, AAA/TACACS+ 명령, SNMP 명령 및 배너 명령이 포함됩니다. 또한 글로벌 디바이스 컨피그레이션 표준에서 중요한 것은 관리자가 디바이스의 DNS(Domain Name System) 이름을 기반으로 디바이스, 디바이스 유형 및 디바이스 위치를 식별할 수 있도록 하는 적절한 디바이스 명명 규칙입니다. 글로벌 컨피그레이션 일관성은 네트워크 환경의 전반적인 지원 가능성과 신뢰성에 있어 중요합니다. 네트워크 복잡성을 줄이고 네트워크 지원 가능성을 높이는 데 도움이 되기 때문입니다. 부정확하거나 일관성 없는 디바이스 동작, SNMP 액세스 및 일반 디바이스 보안으로 인해 컨피그레이션 표준화가 이루어지지 않는 경우가 많습니다.

글로벌 디바이스 컨피그레이션 표준을 유지 관리하는 것은 일반적으로 유사한 네트워크 디바이스에 대한 글로벌 컨피그레이션 매개변수를 생성하고 유지 관리하는 내부 엔지니어링 또는 운영 그룹에 의해 수행됩니다. 또한 전역 컨피그레이션 파일의 복사본을 TFTP 디렉토리에 제공하여 처음에 새로 프로비저닝된 모든 디바이스에 다운로드할 수 있도록 하는 것이 좋습니다. 또한 각 컨피그레이션 매개변수에 대한 설명과 함께 표준 컨피그레이션 파일을 제공하는 웹 액세스 가능 파일도 유용합니다. 일부 조직에서는 정기적으로 유사한 디바이스를 전체적으로 구성하여 글로벌 컨피그레이션의 일관성을 보장하거나 디바이스를 주기적으로 검토하여 올바른 글로벌 컨피그레이션 표준을 수립합니다. 프로토콜 및 인터페이스 컨피그레이션 표준은 인터페이스 및 프로토콜 컨피그레이션에 대한 표준을 유지 관리하는 방식을 나타냅니다.

프로토콜 및 인터페이스 컨피그레이션 일관성은 네트워크 복잡성을 줄이고, 예상되는 디바이스 및 프로토콜 동작을 제공하고, 네트워크 지원 가능성을 개선하여 네트워크 가용성을 향상시킵니다. 프로토콜 또는 인터페이스 컨피그레이션 불일치로 인해 예기치 않은 디바이스 동작, 트래픽 라우팅 문제, 연결 문제 증가, 사후 대응적 지원 시간 증가 등이 발생할 수 있습니다. 인터페이스 컨피그레이션 표준에는 CDP 인터페이스 설명자, 캐싱 컨피그레이션 및 기타 프로토콜별 표준이 포함되어야 합니다. 프로토콜별 컨피그레이션 표준에는 다음이 포함될 수 있습니다.

- IP 라우팅 컨피그레이션
- DLSW 컨피그레이션
- 액세스 목록 구성

- ATM 컨피그레이션
- 프레임 릴레이 컨피그레이션
- 스패닝 트리 구성
- VLAN 할당 및 구성
- VTP(Virtual Trunking Protocol)
- HSRP

참고: 네트워크 내에서 구성된 내용에 따라 다른 프로토콜별 컨피그레이션 표준을 가질 수 있습니다.

IP 표준의 예는 다음과 같습니다.

- 서브넷 크기
- 사용된 IP 주소 공간
- 사용된 라우팅 프로토콜
- 라우팅 프로토콜 컨피그레이션

프로토콜 및 인터페이스 컨피그레이션 표준을 유지하는 것은 일반적으로 네트워크 엔지니어링 및 구현 그룹의 책임입니다. 엔지니어링 그룹은 표준을 식별, 테스트, 검증 및 문서화하는 업무를 담당해야 합니다. 그러면 구현 그룹은 엔지니어링 문서 또는 구성 템플릿을 사용하여 새 서비스를 프로비저닝할 책임이 있습니다. 엔지니어링 그룹은 일관성을 보장하기 위해 필요한 표준의 모든 측면에 대한 문서를 작성해야 합니다. 구성 표준을 적용할 수 있도록 구성 템플릿도 만들어야 합니다. 운영 그룹도 표준에 대한 교육을 받아야 하며 비표준 구성 문제를 파악할 수 있어야 합니다. 구성 일관성은 테스트, 검증 및 인증 단계에서 매우 유용합니다. 사실, 표준화된 컨피그레이션 템플릿이 없으면 중간 규모의 네트워크에 대해 CIsco IOS 버전을 적절하게 테스트, 검증 또는 인증하는 것은 거의 불가능합니다.

가용성 관리

가용성 관리는 네트워크 가용성을 품질 개선 메트릭으로 사용하여 품질을 개선하는 프로세스입니다. 현재 많은 조직에서 가용성 및 중단 유형을 측정하고 있습니다. 운영 중단 유형에는 하드웨어, 소프트웨어, 링크/캐리어, 전원/환경, 설계 또는 사용자 오류/프로세스가 포함될 수 있습니다. 복구 즉시 운영 중단을 파악하고 근본 원인 분석을 수행하여 가용성을 높일 방법을 파악할 수 있습니다. 고가용성을 실현한 거의 모든 네트워크에는 품질 개선 프로세스가 있습니다.

부록 A - Cisco IOS 릴리스 개요

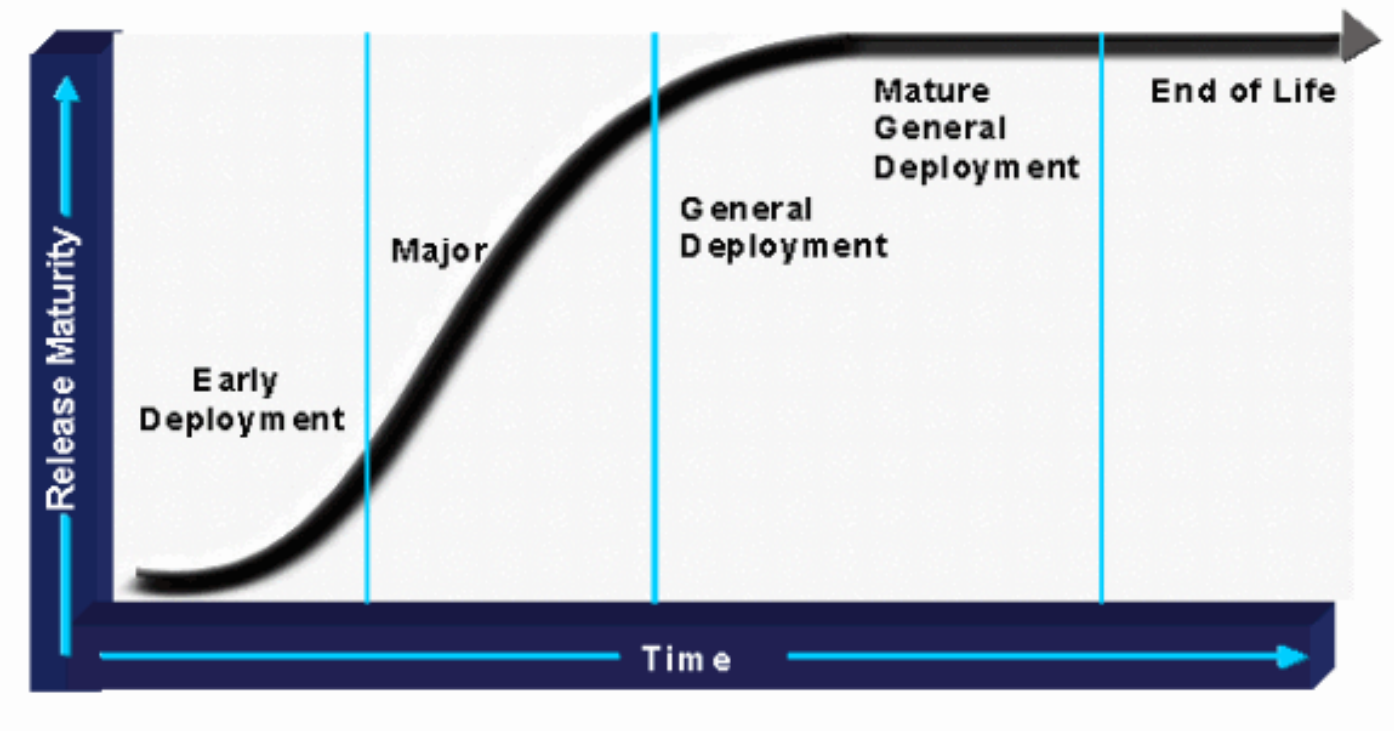
Cisco IOS 소프트웨어 릴리스 전략은 Cisco 고객 네트워크의 성공에 필수적인 건전한 소프트웨어 개발, 품질 보증 및 신속한 출시 기간을 기반으로 합니다.

이 프로세스는 네 가지 릴리스 카테고리를 기준으로 정의되며, 아래에 설명되어 있습니다.

- 초기 구축 릴리스(ED)
- 주 릴리스
- 제한적인 구축 릴리스(LD)
- 일반 배포 릴리스(GD)

Cisco는 개별 릴리스, 대상 시장, 마이그레이션 경로, 새로운 기능 설명 등에 대한 정보를 포함하는 IOS 로드맵을 만들고 유지 관리합니다.

아래 그림은 Cisco IOS 소프트웨어 릴리스 라이프사이클을 보여줍니다.



ED 릴리스

Cisco IOS ED 릴리스는 시장에 새로운 발전을 가져오는 제품입니다. ED 릴리스의 각 유지 보수 버전에는 버그 픽스뿐만 아니라 일련의 새로운 기능, 새로운 플랫폼 지원, 프로토콜 및 Cisco IOS 인프라에 대한 일반적인 개선 사항이 포함됩니다. 1~2년마다 ED 릴리스의 기능과 플랫폼은 다음 메인라인 Cisco IOS 릴리스로 이동됩니다.

ED 릴리스에는 4가지 유형이 있으며, 각각 릴리즈 모델과 라이프 사이클 시점이 약간 다릅니다. ED 릴리스는 다음과 같이 분류할 수 있습니다.

- **CTED(Consolidated Technology Early Deployment) 릴리스**—새로운 Cisco IOS 릴리스 모델은 "T" 교육이라고도 하는 통합 ED 릴리스 열차를 사용하여 Cisco IOS에 새로운 기능, 새로운 하드웨어 플랫폼 및 기타 개선 사항을 소개합니다. 내부 BU(Business Unit) 및 LOB(Line Of Business) 정의를 넘어서는 통합 기술이라고 합니다. 통합 기술 릴리스의 예로는 Cisco IOS 11.3T, 12.0T 및 12.1T가 있습니다.
- **STED(Specific Technology Early Deployment) 릴리스**—STD 릴리스는 특정 기술 또는 시장 지역을 대상으로 한다는 점을 제외하면 CTED 릴리스와 유사한 기능 약정 특성을 갖습니다. 이러한 제품은 항상 특정 플랫폼에서 릴리스되며 Cisco BU의 감독만 받습니다. STED 릴리스는 주 릴리스 버전에 추가된 두 개의 문자를 사용하여 식별됩니다. STED 릴리스의 예는 Cisco IOS 11.3NA, 11.3MA, 11.3WA 및 12.0DA입니다.
- **SMED(Specific Market Early Deployment) 릴리스**—Cisco IOS SMED는 특정 수직 시장 부문(ISP, 기업, 금융 기관, Telcom 기업 등)을 대상으로 한다는 점에서 STED와 차별화됩니다. SMED에는 의도된 수직 시장에서 활용되는 관련성의 특정 플랫폼에 대한 특정 기술 기능 요구 사항만 포함됩니다. CTED는 수직 시장과의 연관성을 가진 특정 플랫폼에만 구축되고 CTED는 더 광범위한 기술 요구 사항을 기반으로 더 많은 플랫폼을 구축한다는 사실에 따라 CTED와 차별화할 수 있습니다. Cisco IOS SMED 릴리스는 CTED와 마찬가지로 주 릴리스 버전에 추가된 하나의 알파벳 문자로 식별됩니다. SMED의 예로는 Cisco IOS 12.0S 및 12.1E가 있습니다.
- **X Releases(XED)라고도 하는 단기 조기 구축 릴리스**—Cisco IOS XED 릴리스는 새로운 하드웨어 및 기술을 시장에 출시합니다. 소프트웨어 유지 보수 개정판을 제공하지 않으며 정기적인 소프트웨어 임시 개정을 제공하지 않습니다. CTED와 통합되기 전에 XED에서 결함이 발견되면 소프트웨어 재구축이 시작되고 이름에 번호가 추가됩니다. 예를 들어 Cisco IOS 릴리스

12.0(2)XB1 및 12.0(2)XB2는 12.0(2)XB 재구축의 예입니다.

주요 릴리스

주요 릴리스는 Cisco IOS 소프트웨어 제품을 위한 기본 구축 수단입니다. C isco IOS Technology Division에서 관리하며 이전 ED 릴리스에서 널리 사용되는 기능, 플랫폼, 기능, 기술 및 호스트를 통합합니다. Cisco IOS 주요 릴리스는 더 우수한 안정성과 품질을 추구합니다. 따라서 주요 릴리스에서는 기능이나 플랫폼 추가를 허용하지 않습니다. 각 유지 보수 버전은 버그 수정만 제공합니다. 예를 들어 Cisco IOS Software 릴리스 12.1 및 12.2는 주요 릴리스입니다.

주요 릴리스에는 완전히 회귀 테스트를 거친 유지 보수 릴리스라는 정기 유지 관리 업데이트가 있으며 최신 버그 수정 사항을 통합하고 새로운 플랫폼이나 기능을 지원하지 않습니다. 주요 릴리스의 릴리스 번호는 주요 릴리스와 유지 보수 레벨을 나타냅니다. Cisco IOS Software 릴리스 12.0(7)에서 12.0은 주요 릴리스의 번호이며 7은 유지 관리 수준입니다. 전체 릴리스 번호는 12.0(7)입니다. 마찬가지로, 12.1은 주요 릴리스이고 12.1(3)은 주요 Cisco IOS Software 릴리스 12.1의 세 번째 유지 보수 릴리스입니다.

LD(Limited Deployment) 릴리스

LD는 FCS와 기본 릴리스를 위한 일반 구축 간의 Cisco IOS 성숙도 단계입니다. Cisco IOS ED 릴리스는 GD 인증을 획득하지 않으므로 제한된 구축 단계에서만 실행됩니다.

일반 배포(GD) 릴리스

Cisco는 릴리스 수명 주기 동안 GD 인증을 위한 주요 릴리스를 선언합니다. 주 릴리스만 GD 상태를 달성할 수 있습니다. Cisco가 릴리스가 다음과 같은 조건을 충족한 경우 GD 인증 이정표를 충족합니다.

- 다양한 네트워크에서 광범위한 시장 노출을 통해 검증되었습니다.
- 안정성 및 버그 트렌드에 대해 분석된 메트릭으로 적격
- 고객 만족도 설문조사를 통해 검증
- 고객의 표준화된 트렌드가 감소하면서 이전 4개 유지 보수 릴리스에 비해 릴리스에서 결함이 발견되었습니다.

TAC 엔지니어, AES(Advanced Engineering Services) 엔지니어, System Test Engineering 및 Cisco IOS Engineering으로 구성된 고객 지원 GD 인증은 릴리스의 모든 미결 결함을 평가하기 위해 구성됩니다. 이 팀은 GD 인증을 최종 승인합니다. 릴리스가 GD 상태가 되면 릴리스의 모든 후속 개정도 GD입니다. 따라서 릴리스가 GD로 선언되면 자동으로 제한된 유지 관리 단계로 들어갑니다. 이 단계에서 주요 코드 재작업의 버그 수정을 포함한 코드의 엔지니어링 수정은 프로그램 관리자에 의해 엄격하게 제한되고 제어됩니다. 이렇게 하면 GD 인증 Cisco IOS 소프트웨어 버전에 부정적인 버그가 발생하지 않습니다. GD는 특정 유지 관리 버전으로 달성됩니다. 해당 릴리스의 후속 유지 보수 업데이트도 GD 릴리스입니다. 예를 들어, Cisco IOS Software 릴리스 12.0은 12.0(8)에서 GD 인증을 받았습니다. 따라서 Cisco IOS Software 릴리스 12.0(9), 12.0(10) 등은 GD 릴리스입니다.

실험 또는 진단 이미지

실험 또는 진단 이미지는 종종 엔지니어링 스페셜이라고 하며 중요한 소프트웨어 문제가 확인된 경우에만 생성됩니다. 이러한 이미지는 일반적인 릴리스 프로세스의 일부가 아닙니다. 이 카테고리의 이미지는 문제를 진단하거나, 버그 수정을 테스트하거나, 즉시 문제를 해결할 수 있도록 설계된 고객별 구축입니다. 다음 중간 또는 유지 보수 릴리스를 기다리는 옵션이 아닌 경우 즉시 수정 사항을 제공할 수 있습니다. 모든 릴리스 유형의 유지 보수 또는 중간 버전을 포함하여 지원되는 모든 소프트웨어 베이스에 실험 또는 진단 이미지를 구축할 수 있습니다. 공식 명명 규칙이 없지만 대부분의 경우 개발자는 기본 이미지 이름에 이니셜, exp(실험적) 또는 추가 숫자를 추가합니다. Cisco TAC

및 Cisco IOS 릴리스 작업에서는 기호 테이블 또는 기본 이미지 이력 같은 지원 문서를 유지 관리하지 않으므로 이러한 이미지는 Cisco 개발과 함께 일시적으로 지원됩니다. 이러한 이미지는 Cisco 내부 테스트를 거치지 않습니다.

릴리스 라이프 사이클 마일스톤

GD 릴리스는 최신 네트워킹 기술로 대체되는 경우도 있습니다. 따라서 릴리스 처분 프로세스는 다음 세 가지 주요 이정표로 설정됩니다.

- **EOS(End of Sales)**—주요 릴리스의 경우 EOS 날짜는 FCS(First Commercial Shipment) 날짜 이후 3년입니다. 새 시스템에 대해 릴리스를 구매할 수 있는 최종 날짜를 설정합니다. EOS 릴리스는 유지 보수 업그레이드를 위해 CCO(Cisco Connection Online)에서 다운로드할 수 있습니다.
- **EOE(End of Engineering)**—EOE 릴리스는 GD 릴리스에 대한 마지막 유지 보수 릴리스이며, 일반적으로 EOS 릴리스 후 약 3개월 후에 제공됩니다. 고객은 Cisco TAC에서 기술 지원을 계속 받을 수 있으며 CCO에서 EOE 릴리스를 다운로드할 수 있습니다. EOS 및 EOE 릴리스 및 날짜를 알리는 제품 게시판은 계획된 EOS 날짜 1년 전에 게시됩니다. 현재 고객은 최신 네트워킹 기술을 활용하기 위해 Cisco IOS 소프트웨어 업그레이드를 조사하기 시작해야 합니다.
- **EOL(End of Life)**—릴리스 수명 주기가 끝날 때 Cisco IOS 소프트웨어 릴리스에 대한 모든 지원이 종료되며 EOL 날짜에 더 이상 다운로드할 수 없습니다. 일반적으로 EOL 날짜는 EOE 날짜 후 5년입니다. EOL 제품 게시판은 실제 EOL 날짜 약 1년 전에 게시됩니다.

Cisco IOS 버전 명명 규칙

Cisco IOS 이미지 이름 지정 규칙은 릴리스된 모든 이미지의 전체 프로필을 제공합니다. 이름은 항상 주 릴리스 식별자와 유지 관리 릴리스 식별자를 포함합니다. 이름에는 열차 지정자, 재작성 지정자(유지 관리 릴리즈의 경우), 사업부(BU) 특정 기능 지정자 및 BU 특정 기능 지정자 rebuild-identifier가 포함될 수도 있습니다. 형식은 다음과 같이 분류할 수 있습니다.

[x.y (z[p])] [A] [o [u(v[p])]] 12.1(8a)E6

명명 규칙 섹션	설명
x.y	'.'로 구분된 두 개의 개별(하나 또는 두 개의) 숫자 식별자 조합 주요 릴리스 값을 식별합니다. 이 값은 Cisco IOS 마케팅에 의해 결정됩니다. 예: 12.1
z	x.y의 유지 보수 릴리스를 식별하는 1~3자리 숫자입니다. 이것은 8주마다 일어난다. 값은 beta에서 0, FCS에서는 1, 첫 번째 유지 보수 릴리스의 경우 2입니다. 예: 12.1(2)
p	x.y(z)의 재구축을 식별하는 하나의 알파 문자. 값은 첫 번째 재구축의 경우 소문자 "a"로 시작하고, 그 다음에는 "b" 등으로 시작합니다. 예: 12.1(2a)
A	1~3개의 알파벳 문자는 릴리스 열차의 지정자이며 CTED, STED 및 X 릴리스에 필수적입니다. 또한 제품

	또는 플랫폼 제품군을 식별합니다. 기술 ED 릴리스는 두 개의 문자를 사용합니다. 첫 번째 문자는 기술을 나타내고 두 번째 문자는 차별화를 위해 사용됩니다. 예를 들면 다음과 같습니다. 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) 릴리스 이름의 첫 번째 위치에 있는 "X"는 CTED "T" 기차를 기반으로 한 일회성 릴리스를 식별합니다. 예: XA, XB, XC 등 릴리스 이름의 두 번째 위치에 있는 "X" 또는 "Y"는 STED 릴리스를 기반으로 또는 관련된 단기 ED 릴리스를 식별합니다. 예를 들어 11.3NX(11.3NA 기반), 11.3WX(11.3WA 기반) 등이 있습니다.
o	특정 릴리스 값의 재구축을 식별하는 하나 또는 두 자리 숫자 지정자(선택 사항). 재구축을 나타내지 않으면 비워 둡니다. 1, 2 등으로 시작합니다. 예: 12.1(2)T1, 12.1(2)XE2
u	BU 특정 릴리스의 기능을 식별하는 하나 또는 두 자리 숫자 지정자입니다. 이 값은 BU 마케팅 팀에 의해 결정됩니다. 예: 11.3(6)WA4, 12.0(1)W5
v	BU 특정 코드의 유지 관리 릴리스를 식별하는 1~2자리 숫자 지정자 값은 beta에서 0, FCS에서는 1, 첫 번째 유지 관리 릴리스에서는 2입니다. 예: 11.3(6)WA4(9), 12.0(1)W5(6)
p	특정 기술 릴리스의 재구축을 식별하는 하나의 알파 문자 지정자입니다. 값은 첫 번째 재구축의 경우 소문자 "a"로 시작하고, 그 다음 "b" 등으로 시작합니다. 예: 11.3(6)WA4(9a)는 11.3(6)WA4(9)의 재구축입니다.

다음 그래프는 Cisco IOS 명명 규칙의 다른 섹션에 레이블을 지정합니다.

부록 B - Cisco IOS 신뢰성

Cisco IOS 신뢰성은 Cisco가 지속적으로 개선하려는 영역입니다. 고객 지향 모범 사례에 대해 논의하기 전에 Cisco 내부 IOS 품질 및 신뢰성 노력에 대한 이해가 필요합니다. 이 섹션에서는 주로 Cisco IOS 소프트웨어 품질에 대한 Cisco의 최근 노력 및 소프트웨어 안정성과 관련하여 고객이 무엇을 추정해야 하는지 간략히 설명합니다.

Cisco IOS 품질 프로그램

Cisco는 GEM GEM(Great Engineering Methodology)이라는 잘 정의된 IOS 개발 프로세스를 보유하고 있습니다. 이 프로세스에는 3단계 라이프 사이클이 있습니다.

- 전략 및 계획
- 실행
- 구축

라이프사이클 내의 일반적인 영역에는 기능 소개 우선 순위 지정, 개발, 테스트 프로세스, 소프트웨어 소개 단계, FCS(First Customer Shipped), GD, 지속적인 엔지니어링 등이 있습니다. Cisco는 또한 ISO(International Standards Organization), Telcordia(이전의 Belcore), IEEE 및 Carnegie Melon Software Engineering Institute와 같은 조직에서 제공하는 다양한 소프트웨어 품질 모범 사례 지침을 따릅니다. 이러한 지침은 Cisco의 GEM 프로세스에 통합되어 있습니다. Cisco 소프트웨어 개발 프로세스는 ISO 9001(1994) 인증을 받았습니다.

Cisco IOS 소프트웨어 품질 개선을 위한 기본 프로세스는 Cisco가 고객의 의견을 듣고, 목표와 메트릭을 정의하고, 모범 사례를 구현하고, 결과를 모니터링하는 고객 중심 프로세스입니다. 소프트웨어 품질 향상을 위해 최선을 다하는 조직 간 팀이 이 프로세스를 실행합니다. Cisco IOS 품질 개선 프로세스의 다이어그램은 다음과 같습니다.

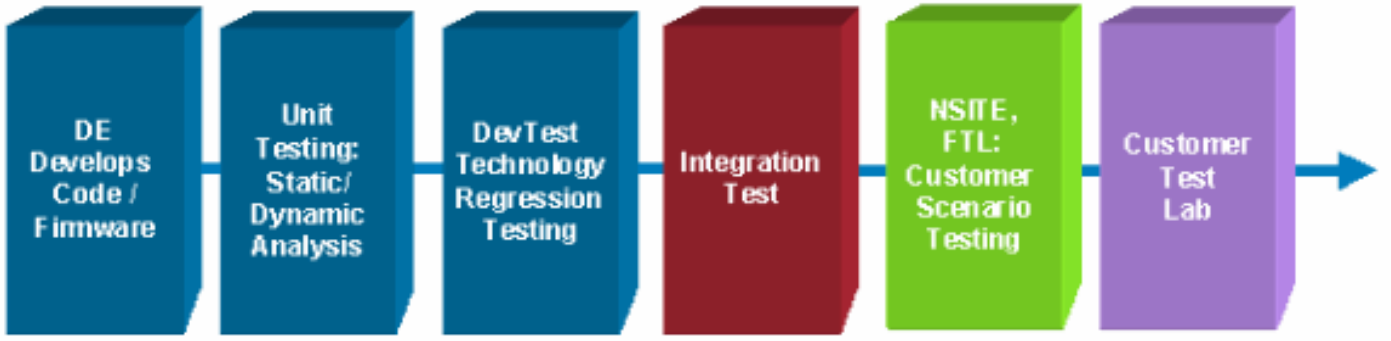
품질 개선 프로세스에는 FY2002 이상의 측정 가능한 목표가 있습니다. 이러한 목표의 주요 초점은 테스트 주기 초기에 소프트웨어 문제를 식별하여 결함을 줄이고, 결함 백로그를 줄이고, 기능의 일관성 및 소프트웨어 릴리스 명확성을 개선하고, 예측 가능한 일관성 있는 릴리스 일정과 소프트웨어 품질을 제공하는 것입니다. 이러한 영역을 해결하기 위한 이니셔티브에는 새로운 테스트 커버리지 툴(테스트 커버리지가 취약한 영역 식별), 테스트 해결 조치 프로세스 개선, Cisco IOS 시스템 회귀 테스트 개선 등이 포함됩니다. 이러한 문제를 해결하기 위해 추가 리소스가 적용되었으며, 모든 주요 Cisco IOS 소프트웨어 릴리스에 대해 경영진 및 부서간 책임이 있습니다.

Cisco IOS 릴리스 테스트

Cisco는 소프트웨어 신뢰성 품질 노력의 핵심 요소로서 테스트의 품질, 범위 및 범위를 파악하고 있습니다. 전반적으로 Cisco는 다음과 같은 IOS 품질 목표를 가지고 있습니다.

- 발견된 Cisco 내부 회귀 결함을 줄입니다. 여기에는 개발 품질이 향상되고 정적/동적 분석에서 더 많은 문제를 식별하는 기능이 포함됩니다.
- 고객이 발견한 결함 감소
- 총 미결 결함 감소
- 소프트웨어 릴리스 명확성 및 기능 일관성 향상
- 일정 및 품질과 함께 기능 및 유지 관리 릴리스 제공

Cisco 내부 테스트는 테스트 단계에서 서로 다른 결함이 식별되는 프로세스로 간주될 수 있습니다. 전체적인 목표는 올바른 Lab에서 적절한 종류의 결함을 찾는 것입니다. 이는 여러 가지 이유로 중요합니다. 첫 번째로 중요한 것은 적절한 테스트 커버리지가 이후 테스트 단계에서 존재하지 않을 수 있다는 것입니다. 또한 이전 단계에서 자동화할 수 있는 능력과 나중에 요구되는 복잡성과 전문 지식의 증가로 인해 테스트 비용이 단계별로 크게 증가합니다. 다음 다이어그램은 Cisco IOS의 테스트 스펙트럼을 보여줍니다.



첫 단계는 소프트웨어 개발입니다. Cisco는 초기 소프트웨어 품질을 개선하기 위해 이 분야에서 여러 노력을 기울이고 있습니다. 개발 그룹은 코드 검토 또는 여러 코드 검토를 수행하여 다른 개발자가 소프트웨어 변경 사항 또는 새 기능 코드를 승인하도록 합니다.

다음 단계는 단위 테스트입니다. 단위 테스트는 랩을 사용하지 않고 소프트웨어 상호 작용을 검사하는 도구를 사용합니다. DevTest는 기능 테스트 및 회귀 테스트를 포함하는 랩 테스트입니다. 기능/기능 테스트는 지정된 기능의 기능을 검사하도록 설계되었습니다. 여기에는 기능 사양에 정의된 대로 모든 기능 순열의 컨피그레이션, 컨피그레이션 취소 및 테스트가 포함됩니다. 회귀 테스트는 기능 및 동작을 지속적으로 검증하도록 설계된 자동화된 테스트 시설에서 수행됩니다. 이 테스트에서는 주로 ping과 제한된 트래픽 생성을 사용하는 다양한 네트워크 토폴로지에서 라우팅, 스위칭 및 기능 기능에 중점을 둡니다. 회귀 테스트는 가능한 순열의 수가 너무 많아 기능, 플랫폼, 소프트웨어 버전 및 토폴로지의 제한된 조합에서만 수행되지만, 현재 4,000개 이상의 회귀 테스트 스크립트가 사용됩니다. 통합 테스트는 보다 포괄적인 제품 및 상호 운용성을 위해 랩 테스트 기능을 확장하도록 설계되었습니다. 통합 테스트에서는 상호 운용성 테스트, 스트레스 및 성능 테스트, 시스템 테스트, 부정적 테스트(예기치 않은 이벤트 테스트)를 포함하도록 테스트를 확장하여 테스트 코드 범위

를 확대합니다.

다음 실습 단계에서는 일반적인 고객 환경에 대한 엔드 투 엔드 테스트를 제공합니다. 위의 다이어그램에 FTL(Financial Test Lab) 및 NSITE, 고객 시나리오 테스트로 나와 있습니다. FTL은 미션 크리티컬 금융 커뮤니티를 위한 테스트를 제공하도록 설계되었습니다. NSITE는 다양한 Cisco IOS 기술에 대해 보다 심층적인 테스트를 제공하는 그룹입니다. NSITE 및 FTL 랩은 확장성 및 성능 테스트, 업그레이드 가능성, 가용성 및 복원력, 상호 운용성, 서비스 가용성 등의 영역에 중점을 둡니다. 서비스 용이성은 대량 프로비저닝 문제, 이벤트 관리/상관관계 및 로드 시 문제 해결에 중점을 둡니다. Cisco에는 이러한 영역을 테스트하는 데 도움이 되는 다양한 수직 시장을 위한 다른 랩이 있습니다.

위 다이어그램에 표시된 최종 Lab은 고객 Lab으로 식별됩니다. 고객 테스트는 기능, 구성, 플랫폼, 모듈 및 토폴로지의 정확한 조합을 철저히 테스트하기 위해 품질 노력의 연장이며 고가용성 환경에 권장됩니다. 테스트 커버리지에는 식별된 토폴로지의 네트워크 확장성 및 성능, 특정 애플리케이션 테스트, 식별된 컨피그레이션의 부정적 테스트, Cisco 이외의 디바이스에 대한 상호 운용성 테스트, 번인 테스트 등이 포함되어야 합니다.

소프트웨어 MTBF

전반적인 신뢰성의 가장 일반적인 메트릭 중 하나는 평균 MTBF(Time between Failure)입니다. MTBF는 MTBF를 사용하여 하드웨어 신뢰성을 위해 개발된 분석 기능으로 인해 소프트웨어 안정성을 위한 MTBF가 유용합니다. 일부 기존 표준을 사용하여 하드웨어 안정성을 더욱 정확하게 확인할 수 있습니다. Cisco는 Telcordia Technologies의 표준 MTBF 데이터를 기반으로 부품 수 방법을 활용합니다. 그러나 MTBF 소프트웨어는 해당 분석 방법론이 없으며 MTBF 분석을 위해 현장 측정에 의존해야 합니다.

지난 3년 동안 Cisco는 Cisco 내부 IT 네트워크에 대한 소프트웨어 신뢰성 필드 측정을 수행했으며 이 작업은 Cisco 내에서 문서화되어 있습니다. 이 작업은 네트워크 관리 SNMP 트랩 정보 및 가동 시간 정보를 사용하여 측정할 수 있는 Cisco IOS 디바이스에 대한 소프트웨어 강제 충돌 사고를 기반으로 합니다. 이 연구는 식별된 소프트웨어 릴리스에 대한 통계적 로그 정상 분포 모델을 사용하여 소프트웨어 안정성을 식별합니다. 평균 MTTR(Time to Repair) 소프트웨어 장애 복구 시간은 평균 라우터 재시작 및 복구 시간을 기준으로 합니다. 6분 복구 시간은 엔터프라이즈 환경에 사용되며, 대규모 인터넷 서비스 공급자(ISP)는 15분 동안 사용됩니다. 이 지속적인 조사의 결과는 소프트웨어가 릴리스될 때 또는 몇 가지 유지 보수 버전 이후에 일반적으로 미세 9 가용성을 충족하며, 소프트웨어 강제 충돌을 유일한 다운타임 소스로 사용하여 측정한 결과 시간이 지남에 따라 더 높은 가용성을 제공한다는 것입니다. 이 연구는 잠재적 MTBF 값을 초기 구축 소프트웨어의 경우 5,000시간에서 일반 구축 소프트웨어의 경우 50,000시간 범위로 파악했습니다.

이 작업에 대한 가장 일반적인 반박은 소프트웨어 강제 작동 중단이 소프트웨어 신뢰성 문제로 인해 발생하는 모든 가동 중단 시간을 포함하지 않는다는 것입니다. 이 메트릭을 품질 개선 작업에 사용하는 경우 소프트웨어 강제 충돌 속도를 개선하는 데 도움이 될 수 있지만 기타 중요한 소프트웨어 안정성 영역은 무시할 수 있습니다. 통계적 방법론을 이용해 소프트웨어 안정성을 정확하게 예측하기가 쉽지 않아 이 발언은 큰 답이 아닐 수 없다. Cisco 소프트웨어 품질 통계 전문가들은 더 광범위한 중단 유형을 사용하여 소프트웨어 MTBF를 안정적으로 예측하려면 더 큰 샘플 데이터 세트가 필요하다는 결론을 내렸습니다. 또한 네트워크 복잡성, 소프트웨어 관련 문제 해결을 위한 직원의 전문성, 네트워크 설계, 지원 기능, 소프트웨어 관리 프로세스 등의 변수 때문에 이론적 통계 분석은 어려울 수 있습니다.

이 시점에서 이러한 유형의 민감한 데이터를 정확하게 수집하기 어려워서 현장 측정에 따라 소프트웨어 안정성을 더욱 정확하게 예측하기 위한 업계의 작업은 아직 이루어지지 않았습니다. 또한 대부분의 고객은 가용성 데이터의 독점적 특성으로 인해 Cisco가 네트워크에서 직접 가용성 정보를 수집하는 것을 원치 않습니다. 그러나 일부 조직은 소프트웨어 신뢰성에 대한 데이터를 수집하며

Cisco는 소프트웨어 중단으로 인한 가용성에 대한 메트릭을 수집하고 이러한 가동 중단에 대한 근본 원인 분석을 수행하도록 조직에 권장합니다. 소프트웨어 신뢰성이 높은 조직에서는 이러한 사전 대응적인 태도를 통해 제어할 수 있는 여러 가지 방식을 통해 소프트웨어 안정성을 향상시켰습니다 .

소프트웨어 신뢰성 가정

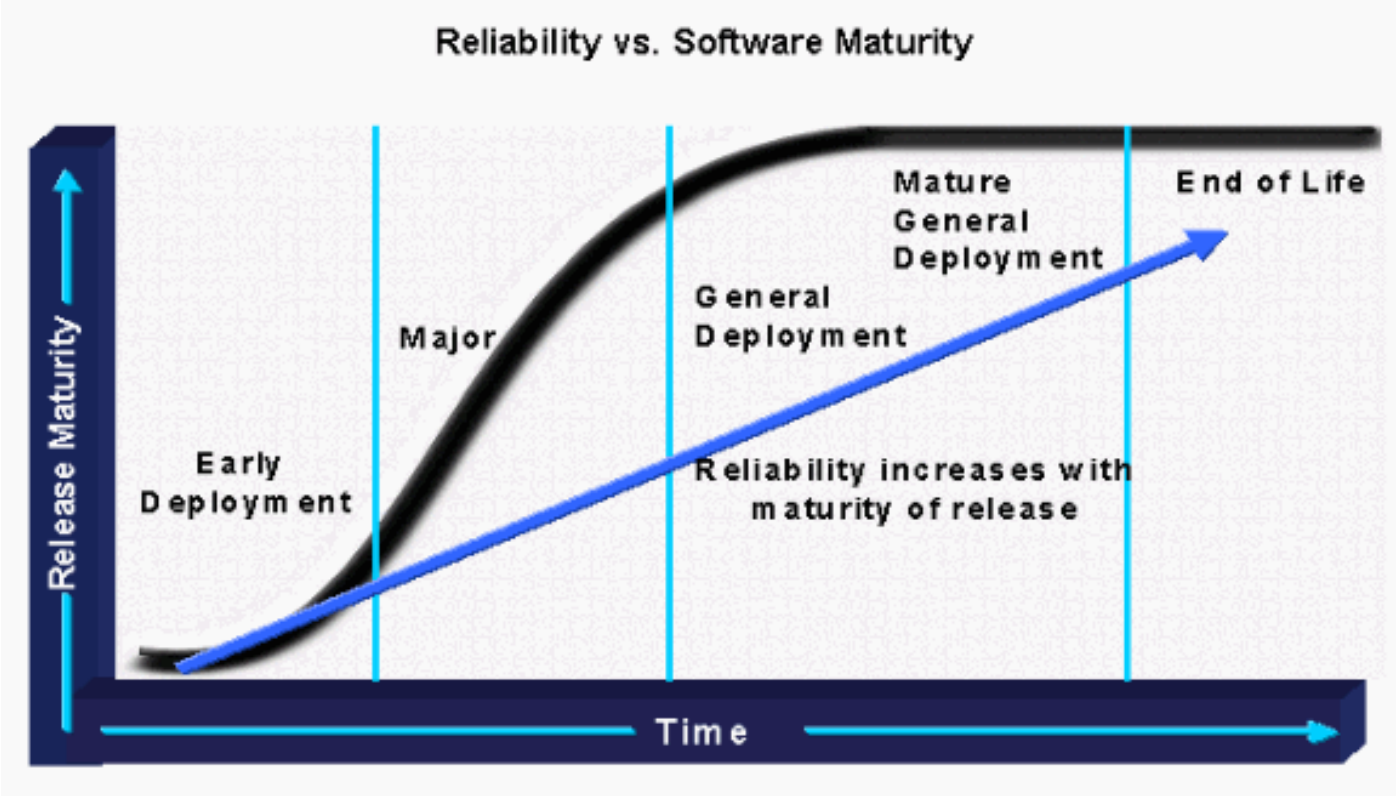
고객 피드백, Cisco IOS Technologies 그룹에서 수행한 사전 연구 및 Cisco Advanced Services 팀에서 수행한 근본 원인 분석 결과, 소프트웨어 안정성을 개선하는 데 도움이 되는 몇 가지 새로운 가정 및 모범 사례가 형성되었습니다. 이러한 가정에서는 테스트 책임, 소프트웨어 성숙도 또는 연령, 활성화된 기능, 구축된 소프트웨어 버전 수에 중점을 둡니다.

책임 테스트

첫 번째 새로운 가정은 테스트 책임을 다룹니다. Cisco는 새로운 기능 및 기능이 신제품에서 작동하도록 테스트/검증하는 업무를 항상 담당합니다. 또한 Cisco는 새로운 소프트웨어 버전이 이전 버전과 호환되는지 확인하기 위해 회귀 테스트를 수행합니다. 그러나 Cisco는 고객 환경에서 발생할 수 있는 모든 잠재적인 위험(설계 특성, 로드 및 트래픽 프로필)에 대해 모든 기능, 토폴로지 및 플랫폼을 검증할 수는 없습니다. 고객을 위한 고가용성 모범 사례에는 고객이 정의한 기능, 설계, 서비스 및 애플리케이션 트래픽을 사용하여 프로덕션 네트워크를 모방하는 축소된 랩 토폴로지에서 테스트를 수행하는 것이 포함됩니다.

신뢰성 vs. 소프트웨어 성숙도

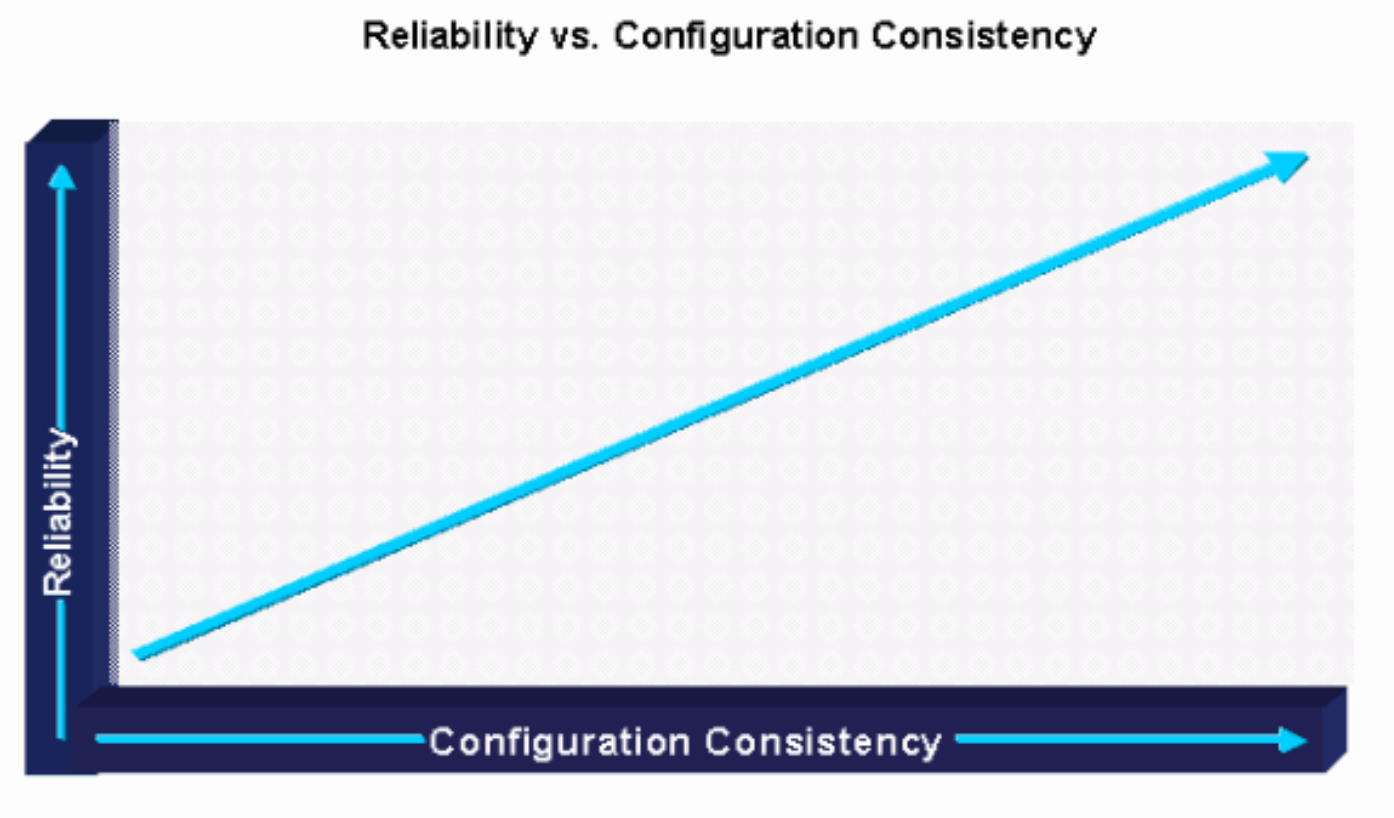
소프트웨어 안정성은 주로 소프트웨어 성숙도의 요인입니다. 노출(사용)을 수신하고 식별된 버그가 수정되면 소프트웨어가 성숙합니다. Cisco 릴리스 운영은 새로운 기능을 추가하지 않고도 소프트웨어가 제대로 작동하는지 확인하기 위해 열차 릴리스 아키텍처로 이동했습니다. 고가용성이 필요한 고객은 현재 필요한 기능을 갖춘 더욱 성숙한 소프트웨어를 찾고 있습니다. 그런 다음 소프트웨어의 성숙도, 가용성 요구 사항, 새로운 기능 또는 기능을 위한 비즈니스 동인 간에 절충이 이루어집니다. 많은 조직에는 허용 가능한 성숙도에 대한 표준 또는 지침이 있습니다. 어떤 사람들은 특정한 열차의 5번째 중간 출하를 받아들일 것이다. 다른 경우에는 9번째 또는 GD 인증이 될 수 있습니다. 궁극적으로 조직은 소프트웨어 성숙도 측면에서 허용 가능한 위험 수준을 결정해야 합니다.



안정성 vs. 기능 및 표준 수량

소프트웨어 안정성은 또한 프로덕션 환경에서 테스트하고 실행하는 코드의 양을 나타냅니다. 다양한 하드웨어 플랫폼과 모듈의 양이 증가함에 따라, 사용되는 코드의 양도 증가하므로 일반적으로 소프트웨어 결함에 대한 노출이 증가합니다. 구성된 프로토콜의 수량, 다양한 컨피그레이션, 구현된 다양한 토폴로지 또는 설계도 마찬가지입니다. 설계, 구성, 프로토콜 및 하드웨어 모듈 요소는 실행되는 코드의 양과 소프트웨어 결함의 위험 또는 노출에 영향을 줄 수 있습니다.

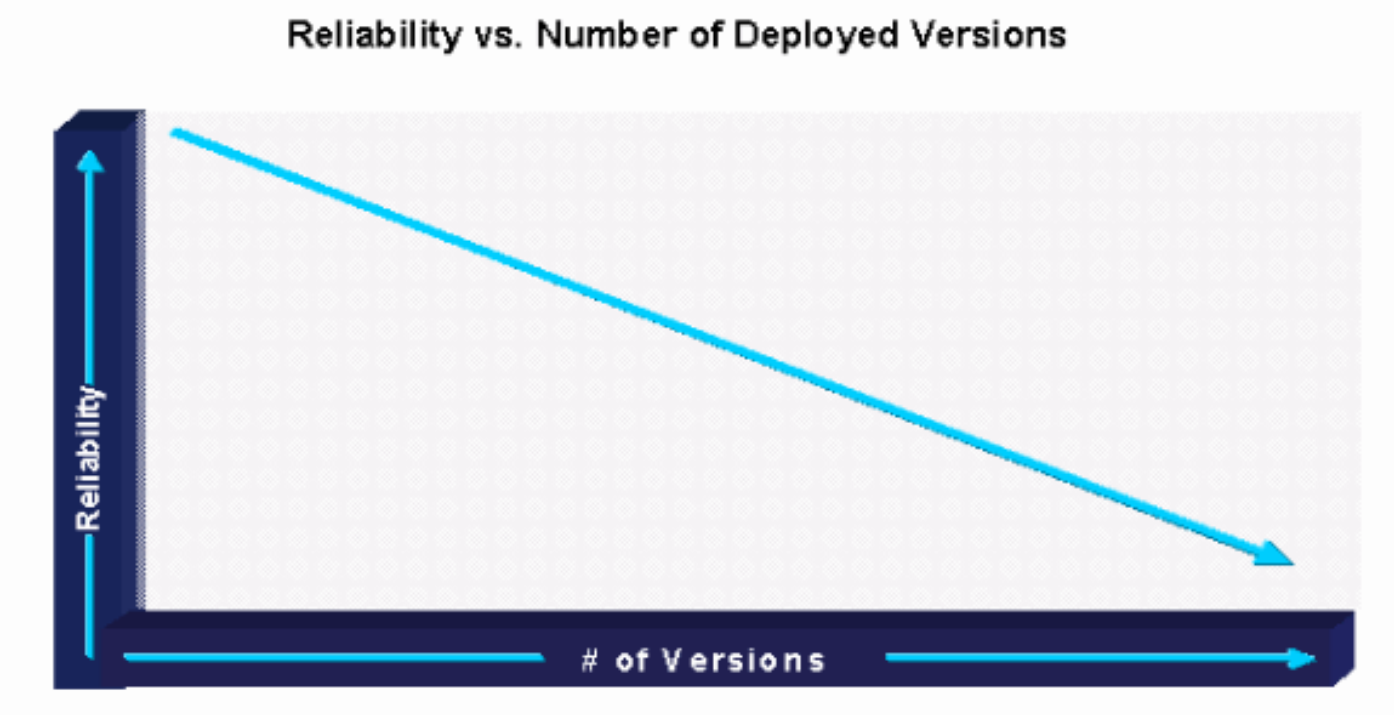
소프트웨어 릴리스 작업에는 일반적으로 특정 영역에서 사용할 수 있는 코드를 제한하는 특수 용도의 소프트웨어가 있습니다. 비즈니스 부서에서는 Cisco 내에서 보다 철저한 테스트를 거치고 고객이 보다 광범위하게 활용할 수 있는 설계 및 구성을 권장합니다. 또한 고객은 표준화된 모듈식 토폴로지 및 표준 구성에 대한 모범 사례를 채택하여 테스트되지 않은 코드 노출 양을 줄이고 전반적인 소프트웨어 안정성을 향상하기 시작했습니다. 일부 고가용성 네트워크에는 엄격한 표준 구성 지침, 모듈식 토폴로지 표준 및 소프트웨어 버전 제어가 있어 테스트되지 않은 코드 노출 위험을 줄일 수 있습니다.



안정성 vs. 구축된 버전 수

소프트웨어 신뢰성의 또 다른 요소는 버전 간의 상호 운용성과 여러 버전에서 실행되는 방대한 양의 코드입니다. 소프트웨어 버전의 수량이 증가하면, 실행된 코드의 양도 증가하여 소프트웨어 결함에 대한 노출이 증가합니다. 여러 버전에서 실행되는 추가 코드로 인해 신뢰성에 대한 위험이 거의 기하급수적으로 증가합니다. 이제 조직은 특정 기능 및 플랫폼 요구 사항을 충족하기 위해 네트워크에서 최소 몇 개의 버전을 실행해야 한다는 사실을 알게 되었습니다. 그러나 대부분의 동종 네트워크 환경에서 50개 이상의 버전을 실행하는 것은 일반적으로 이러한 여러 버전을 제대로 분석하거나 검증할 수 없기 때문에 소프트웨어 문제를 나타냅니다.

Cisco 개발에서는 소프트웨어 안정성을 개선하기 위해 소프트웨어 회귀 테스트를 수행하여 서로 다른 소프트웨어 버전이 호환되는지 확인합니다. 또한 소프트웨어 코드는 모듈형이 더 높으며 코어 모듈은 시간이 지남에 따라 버전 간에 상당히 변경될 가능성이 낮습니다. 또한 Cisco 릴리스 운영에서는 결함이 발견되면 알려진 결함 또는 상호 운용성 문제가 있는 버전이 CCO에서 신속하게 제거되므로 고객이 사용할 수 있는 소프트웨어의 양도 변경되었습니다.



관련 정보

- Cisco IOS(Internetworking Operating Systems)
- 기술 지원 및 문서 – Cisco Systems