

인터페이스 구현

(응용SW개발)

인터페이스 구현 검증



한국기술교육대학교
온라인평생교육원

학습 내용

- 인터페이스 구현 검증도구 및 감시도구
- 인터페이스 구현 검증

학습 목표

- 인터페이스 구현 검증도구 및 감시도구에 대해 설명할 수 있다.
- 인터페이스 구현 검증도구 준비 및 실행 절차를 설명할 수 있다.

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◆ 증명과 검증

증명(Verification)

프로그램소스, 함수 등 개발 산출물을 조사하여
처음 설계대로 구현되었는지 확인하는 과정

제품을 올바르게 생성하고 있는가?
(Are we building the product **right**?)

- 증명 또는 확인으로 번역
- 처음 설계한 명세목록과 맞는지 확인
- 처음 설계관점에서 개발이 정확하게 진행되었는지 확인
- 화이트박스 테스트(White Box Test) 시행
- 각 단계에서 개발자의 시각으로 검토함

이전 단계에서 생성된 산출물이
현 단계의 산출물에 정확히 반영되었는지 검토

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◇ 증명과 검증

증명(Verification)

요구사항 명세서의 내용이
최종 산출물에 맞도록 구현되었는지 보는 것이 아니라



요구사항 명세서의 내용이 분석 및 설계에 반영되었는지,
분석 및 설계의 내용이 프로그램에 구현되었는지
단계별로 확인하는 작업

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◇ 증명과 검증

검증(Validation)

개발된 목적물을 실행하여
처음 사용자가 요청한대로 구현되었는지 확인하는 과정

올바른 제품 생성하고 있는가?
(Are we building the right product?)

- 처음 사용자가 요청한 요구조건 목록과 맞는지 확인
- 처음 요건관점에서 개발이 정확하게 진행되었는지 확인
- 블랙 박스 테스트(Black Box Test) 시행
- 최종단계에서 사용자의 시각으로 검토함

최초 요구사항이 최종 정보시스템에 잘 반영되었는지,
요구사항 명세서의 내용이 최종 산출물에 맞게 구현되었는지
확인하는 작업

1. 인터페이스 구현 검증 도구 및 감시 도구

1 검증과 테스트

◆ 증명 및 검증의 목표

◆ 증명 및 검증의 목표

- 소프트웨어 시스템의 목적에 대한 적합성 및 소프트웨어 시스템의 신뢰성 확보
- 소프트웨어를 의도대로 사용하는데 충분한지 확인됨
- 두 과정 모두 품질향상을 추구함

증명과 검증과정이 끝났다고 반드시
소프트웨어의 결점이 전혀 없는 것은 아님

구현과정을 끝낸 구현 목적물에 대한
증명과 검증을 통한 품질 향상

개발 생명주기 전 단계에 걸친
산출물의 정합성 검토를 통한 품질 향상

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◇ 증명 및 검증의 목표

좁은 의미의 증명과 검증

- 증명과 검증을 단순히 **테스트 기법**으로 보는 시각

구현 단계 후 구현된 목적물에
증명 테스트(화이트박스 테스트 기법수행)와
검증 테스트(블랙박스 테스트)를 시행하는 것

—// 구현 목적물의 정합성 증명 및 검증에 국한 //

넓은 의미의 증명과 검증

- 증명과 검증을 **광의의 시각**으로 해석하는 것이 일반적임

정보시스템 개발 생명주기(SDLC) 전 과정에
걸쳐서 벌어지는 **품질관리(QM)차원**에서
증명과 검증을 이해함

—// 단순히 구현된 목적물 뿐만 아니라
관련 명세서 등 **단계별 산출물**에 대해
증명 및 검증 수행 //

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

❖ 테스트

테스트를 통해 구축된 시스템이 검증되므로
소프트웨어 구현 만큼
소프트웨어 테스트 과정도 중요

테스트 진행순서	수행 내용
단위테스트 시행	시스템을 구성하는 모듈 하나 하나씩 각각 테스트
인터페이스 결합 테스트 시행	해당 모듈 간의 결합, 즉 인터페이스 테스트
종합테스트 시행	시스템 전체의 기능과 성능 테스트
인수테스트 시행	① 모든 단계별 테스트의 문제점 및 오류를 찾아 해결 ② 최종적으로 시스템이 사용할 수 있게 모든 준비가 되어 있는지 확인 ③ 시스템 사용자의 인수가 가능한지를 측정

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◆ 테스트의 분류

1 단위 테스트

- 설계의 최소 단위인 모듈 TEST
- 화이트박스 기법 이용

◆ 단위 테스트 유형

유형	내용
인터페이스	다른 모듈과의 데이터 인터페이스에 대하여 TEST
자료구조	모듈 내의 자료 구조상 오류가 없는지를 TEST
수행경로	구조 및 루프 TEST 등에 의해 논리 경로 TEST
오류처리	각종 오류들이 모듈에 의해 적절히 처리되는지 TEST
경계	오류가 발생하기 쉬운 경계 값으로 TEST 사례

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

📦 테스트의 분류

2 통합 테스트

- 단위 TEST를 거친 모듈 인터페이스(Interface)의 오류발견이 목적
- 모듈 간 체계적인 조합이 이루어 졌는지 확인하는 과정으로 TEST를 통해 모듈간 인터페이스와 관련된 결함을 발견하고 제거하는 작업

통합 테스트 유형

하향식 테스트

상향식 테스트

샌드위치 형
테스트

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

테스트의 분류

2 통합 테스트

하향식 테스트

- 상위 모듈을 하위 모듈보다 먼저 TEST
- 중요 모듈을 가능한 먼저 TEST
- Stub(Dummy Module) 모듈 사용
- 4GL과 같은 Menu-Driven 화면구성 방식 사용

회귀 TEST (Regression Test)

- 수정에 의한 새로운 결함 발생의 가능성에 대비하여, 이미 실시했던 TEST 사례들의 전부 혹은 일부를 재시도

특성

- 실제 적용 가능한 테스트
- Menu 방식 소프트웨어 개발에 적용
- Stub Module 또는 대응모듈 요구

장점

- 실제 응용가능, 테스트 사례 풍부

단점

- Stub Module 구현이 어려움

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◆ 테스트의 분류

2 통합 테스트

상향식 테스트

- 하위 모듈을 상위 모듈보다 먼저 TEST
- 입/출력과 관련된 모듈을 먼저 TEST
- TEST DRIVEN 작성 필요
- 소프트웨어 계층 구조의 최하위부터 점진적으로 모듈들을 통합시켜 나아가는 방식

특성

- 대규모 시스템에 적용
- 모듈의 신뢰성 향상 가능
- 최종 결과 산출이 어려움

장점

- 대형시스템 테스트에 적용

단점

- Cluster 분류가 어려움

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◆ 테스트의 분류

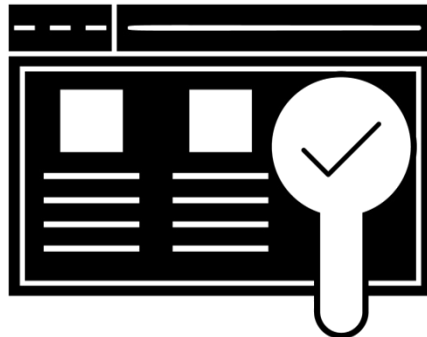
2 통합 테스트

샌드위치 형 테스트

- 하향식과 상향식을 절충한 통합 방식
- 우선적으로 통합을 시도할 중요 모듈들을 선정ان 후, 그 모듈을 중심으로 통합 수행

3 시스템 테스트

- 사용자의 신뢰성을 확보하기 위하여 컴퓨터, 네트워크 제반 모든 사항에 관계된 테스트



1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

◆ 테스트의 분류

3 시스템 테스트

유형	내용
회복 테스트	<ul style="list-style-type: none">• 소프트웨어가 다양한 방법으로 실패하도록 유도하고 회복이 적절하게 수행되는지 검증• 시스템에 의해 자동으로 회복이 수행되면 재 초기화, 데이터 회복, 재 시작 등의 방법에 의해 정상적으로 회복되는지 평가• 운영체제, DBMS, 통신용 소프트웨어 등의 완전성 TEST
안전 테스트	<ul style="list-style-type: none">• 시스템 내의 보호 기능이 불법적인 침투로부터 시스템을 보호하는지 검증• 목적: 해커 등의 불법적 침입자로부터 시스템의 보호

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

테스트의 분류

3 시스템 테스트

유형	내용
강도 테스트	<div><ul style="list-style-type: none">비정상적인 값, 양, 빈도의 자원 입력에 대한 정상 수행상태를 TEST소프트웨어에게 다양한 스트레스를 가해 보는 TEST<div>민감도 TEST(Sensitivity Test)</div><div>유효한 입력 유형 중에서 정보시스템을 불안정하게 하거나 부적절한 결과를 일으키는 데이터의 조합을 밝히도록 함</div></div>
성능 테스트	<div><ul style="list-style-type: none">통합시스템의 전 후 관계에서 소프트웨어의 실행시간 TEST소프트웨어의 효율성을 진단자원 이용, 처리시간, 요구된 응답 반응 등 성능 TEST</div>

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

테스트의 분류

4 인수 테스트

유형	내용
인수 테스트	<ul style="list-style-type: none"> 사용자측 관점에서 소프트웨어가 요구 사항을 충족시키는지 평가 소프트웨어가 고객의 합리적인 기대에 따라 제 기능을 발휘하는지 여부 TEST
	<div>알파 테스트</div> <ul style="list-style-type: none"> 특정 사용자들에 의해 개발자 관점에서 수행 개발자는 사용상의 문제를 기록하여 반영되도록 함
	<div>베타 테스트</div> <ul style="list-style-type: none"> 선정된 다수의 사용자들이 자신들의 사용환경에서 일정 기간 동안 사용해 보면서 문제점이나 개선 사항 등을 기록하고 개발 조직에게 통보하여 반영되도록 함

1. 인터페이스 구현 검증도구 및 감시도구

1 검증과 테스트

📦 테스트의 분류

4 인수 테스트

유형	내용
설치 테스트	<ul style="list-style-type: none">• 목적: 소프트웨어를 사용자 환경에 설치하는 과정 중 나타날 수 있는 결함 발견• 하드웨어/소프트웨어 구성사항 테스트• 파일분배 적재 테스트• 타 소프트웨어와 연결관련 테스트

1. 인터페이스 구현 검증도구 및 감시도구

2 검증도구 및 감시도구

◈ 인터페이스 구현 검증도구

자동화된 인터페이스 구현 및 감시도구를 이용해
구현된 인터페이스의 동작 상태의
검증 및 감시(Monitoring)

1. 인터페이스 구현 검증도구 및 감시도구

2 검증도구 및 감시도구

🔹 인터페이스 구현 검증도구

🔹 인터페이스 구현 검증도구

인터페이스 구현을 검증하기 위해서는?

인터페이스 단위 기능 및
시나리오에 기반한 통합 테스트가 필요

테스트 자동화 도구를 이용하여
단위 및 통합 테스트의 효율성을 높일 수 있음

산출물 명	설명
xUnit	Java(Junit), C++(Cppunit), .Net(Nunit) 등 다양한 언어를 지원하는 단위 테스트 프레임워크
STAF	서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크
FitNesse	웹 기반 테스트 케이스 설계/실행/결과 확인 등을 지원하는 테스트 프레임워크

1. 인터페이스 구현 검증도구 및 감시도구

2 검증도구 및 감시도구

🔹 인터페이스 구현 검증도구

🔹 인터페이스 구현 검증도구

산출물 명	설명
NTAF	Naver 테스트 자동화 프레임워크이며, STAF와 FitNesse를 통합
Selenium	다양한 브라우저 지원 및 개발언어를 지원하는 웹 애플리케이션 테스트 프레임워크
Watir	Ruby 기반 웹 애플리케이션 테스트 프레임워크

1. 인터페이스 구현 검증도구 및 감시도구

2 검증도구 및 감시도구

◆ 인터페이스 감시도구

인터페이스의 동작이 잘 진행되는지 확인하기 위해서는 ?



애플리케이션 모니터링 툴
(APM: Application Performance Management)을
사용하여 동작 상태를 감시

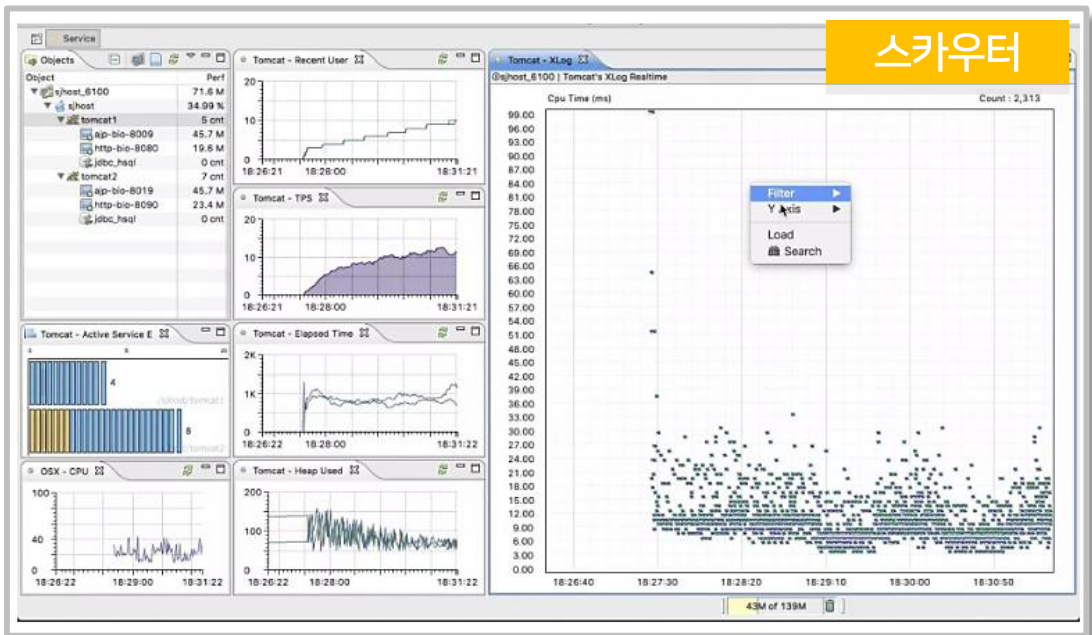
- ◆ 상용 제품 및 오픈 소스를 이용한 애플리케이션 모니터링 툴이 있음
- ◆ 데이터베이스, 웹 애플리케이션의 트랜잭션과 변수 값, 호출 함수, 로그 및 시스템 부하 등 종합적인 정보를 조회하고 분석할 수 있음

1. 인터페이스 구현 검증도구 및 감시도구

2 검증도구 및 감시도구

◆ 인터페이스 감시도구

◆ 오픈 소스 감시도구 예시



2. 인터페이스 구현 검증

1 검증 및 감시도구 준비

◆ 검증 및 감시도구의 요건 분석



구현된 인터페이스 명세서를 참조하여
구현 검증에 필요한 검증 및 감시도구의 요건 분석



구현된 인터페이스 명세서의 세부 기능을 참조하여
구현 검증 및 감시에 필요한 기능 분석



각 기능의 특징에 맞게 요건 도출

2. 인터페이스 구현 검증

1 검증 및 감시도구 준비

◆ 검증 및 감시도구의 요건 분석

기능 구현 정의	검증도구 요건	감시도구 요건
1. 송신 측에서 인터페이스 대상 선택 전송	<ul style="list-style-type: none">• 입력한 대상과 생성된 인터페이스 객체의 정보가 일치하는 지 확인	<ul style="list-style-type: none">• 데이터베이스 SQL 모니터링• 조회 Transaction 모니터링• JSON 생성 객체 모니터링
2. 인터페이스 객체 전송	<ul style="list-style-type: none">• 암호화된 통신으로 올바른 수신 측에 전달되었는지 확인• 전달된 정보가 수신된 정보와 일치하는지 확인• 파싱된 정보가 송신된 정보와 일치하는지 확인	<ul style="list-style-type: none">• 통신 암호화 모니터링• 패킷 정보 모니터링• 연결된 Transaction 변수 모니터링
3. 수신 후 수신 측 트랜잭션과 결과 반환	<ul style="list-style-type: none">• 수신된 데이터와 연관 있는 이후 트랜잭션의 기대 값과 일치 여부	<ul style="list-style-type: none">• 객체 입력, 출력 값 모니터링• 객체 동작 성공, 실패 여부

2. 인터페이스 구현 검증

1 검증 및 감시도구 준비

◆ 검증 및 감시도구 준비

구현 검증 및 감시에 필요한 **도구의 요건 확인**

시장 조사 및 솔루션 조사

감시 및 검증에 필요한 적절한 **도구 선택 및 구매**

—// **오픈 소스 감시도구도 많이 활용되고 있으니**
기능을 분석하여 도입 검토 //—

2. 인터페이스 구현 검증

2 모니터링

❖ 외부 시스템과 연계 모듈의 동작 상태 확인

- ❖ 인터페이스 구현 검증도구를 이용해 외부 시스템(송·수신)과 연계 모듈(송·수신)의 동작 상태 확인

1 전반적인 인터페이스 동작 프로세스상에서 예상되는 결과와 검증 값 비교



최초 입력 값, 입력 값에 의해 선택되는 데이터,
생성되는 객체의 데이터 등

- 2 각 단계별 에러 처리도 적절하게 구현되어 있는지 확인
- 3 인터페이스 검증 시나리오 작성 및 확인

2. 인터페이스 구현 검증

2 모니터링

외부 시스템과 연계 모듈의 동작 상태 확인

인터페이스 검증 시나리오

예 | 발령 시나리오

인터페이스 기능	시나리오	예상 값	검증도구 확인
1. 최초 데이터 입력	과장 10명 입사 발령 확정	10명의 사번이 확정 FLAG를 가지고 저장	예상 값과 동일
2. DB에서 조회	입사자 과장 10명 정보 조회	10명의 인사 발령 정보, 기본 정보가 Rowset 형태로 선택	예상 값과 동일
3. 송신 객체 생성	전송 버튼 클릭	DB에서 조회된 동일 정보가 확정된 10명에 대해서 JSON 형태로 생성	예상 값과 동일
4. 송신 객체 전송		10명 과장 입사 발령 확정 내용이 Submit Flag 를 가지고 송신 이력 적재	예상 값과 동일

2. 인터페이스 구현 검증

2 모니터링

외부 시스템과 연계 모듈의 동작 상태 확인

인터페이스 검증 시나리오

예 | 발령 시나리오

인터페이스 기능	시나리오	예상 값	검증도구 확인
5. 수신 및 파싱		수신된 객체를 파싱한 결과는 송신한 결과와 동일	예상 값과 동일
6. 데이터 트랜잭션	수신 회사 인사 정보로 입력	파싱된 결과는 수신 측의 업무 정의에 따라 트랜잭션 진행 (인사 기본 정보, 발령 정보에 입력)	예상 값과 동일
7. 수신 결과 반송	전송 회사 입사자 정보 조회	수신 결과가 송신 측으로 반환(TRUE) 반환 이력 테이블에 해당 내역 입력	예상 값과 동일

2. 인터페이스 구현 검증

2 모니터링

❖ 외부 시스템과 연계 모듈의 동작 상태 감시(Monitoring)

- ❖ 외부 모듈이 서비스를 제공하는 동안 정상적으로 동작하는지 감시도구를 통해 확인



인터페이스 구현 기술에 따라
인터페이스 검증 및 감시도구의 요건이 달라짐



인터페이스 동작 여부, 에러 발생 여부 등
감시도구에서 제공해 주는 리포트 활용



예상되는 결과와 다른 결과가 나오는지 확인



2. 인터페이스 구현 검증

2 모니터링

◇ 모니터링

모니터링을 위한 관리작업 시행 순서

1 성능 및 상태 측정 시행

- 정보시스템 각각의 구성요소
(서버, 네트워크, DBMS, 응용 소프트웨어)
- 정보시스템 성능의 최적화를 위하여
다음단계에서 이용될 성능데이터 수집

2 성능, 상태에 대한 분석작업 시행

- 1단계에서 추출된 자료를 기초로
정보시스템의 성능과 상태 분석
- 구성요소에 대한 개별적인 분석과
전체적인 통합 성능분석 환경 마련

3 시스템 구성요소 조정 및 검증 시행

- 성능 제고를 위해 1, 2단계에서 얻어진 자료를 기초로
시스템 구성요소를 조정 및 검증 과정
- 각 구성요소 별 연계분야와 상호관련성을 고려하여 진행

2. 인터페이스 구현 검증

2 모니터링

◆ 모니터링 활동

◆ 성능지표 관리

성능지표

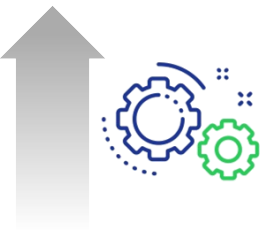
- 정보시스템의 서비스품질(QoS)를 결정하는 속성들 중의 하나
- 성능을 나타내는 지표
응답시간, 시간당 처리량,
자원 사용량, 효율성 등

성능지표 관리목표

- 응답시간



- 시간당 처리량
- 자원사용량
- 효율성



2. 인터페이스 구현 검증

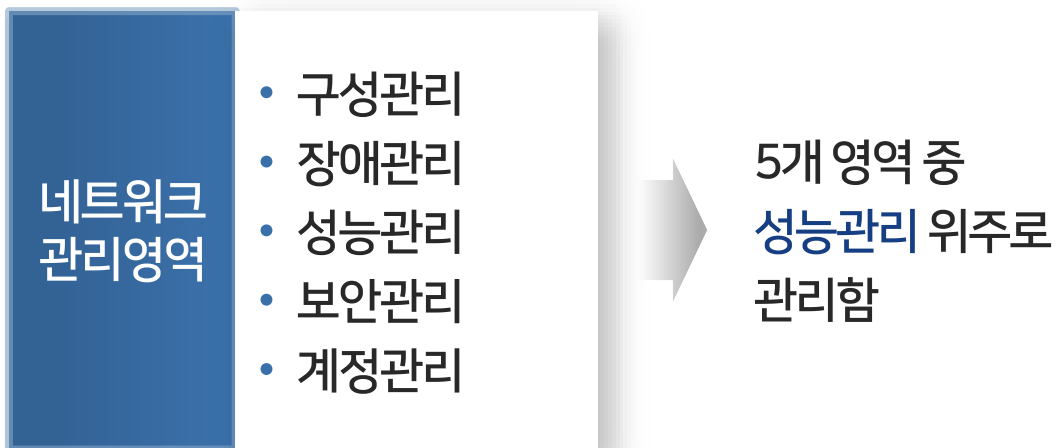
2 모니터링

◆ 모니터링 활동

◆ 모니터링 활동

- 정보시스템을 구성하는 자원에 대한 데이터 수집 및 분석 후 그 결과를 검증하는 활동을 포함
- 시스템 및 조직의 목표, 투자비용 대비 효과를 고려해서 성능관리활동 수행

◆ 인터페이스 관련 네트워크(통신환경)측 모니터링 주요활동



2. 인터페이스 구현 검증

2 모니터링

◆ 모니터링 활동

◆ 네트워크(통신환경)측 모니터링 대상 및 범위

- 이용률, 에러율 등의 성능자료를 계산하고 일정수준의 서비스를 지속적으로 제공하기 위한 기능 제공

2. 인터페이스 구현 검증

2 모니터링

◆ 인터페이스 관련 모니터링 주요활동

1 장비 성능관리

- 관리대상 장비 CPU, 메모리 사용률(평균값, 최대값) 측정, 분석
- 관리대상 장비성능을 장비별, 지역별, 기간별(시간별, 일별, 주간별, 월별)로 측정, 분석
- 장비성능의 임계치 관리 방식에 따른 경보체계 구축 및 관리

장비자체의 성능 측정에 한하며
IPS, IDS, F/W 장비의 S/W적인 문제(성능)는 배제함

2. 인터페이스 구현 검증

2 모니터링

◆ 인터페이스 관련 모니터링 주요활동

2 세션 성능관리

- 조회 조건에 따라 각 스위치 장비의 세션 추이를 측정, 분석
- 스위치별 세션(SLB, FLB, Redirection) 생성시간 및 세션 수량을 측정, 분석
- 스위치별 세션(SLB, FLB, Redirection) 현황을 장비별, 지역별, 기간별로 측정, 분석

3 회선 성능관리

- 네트워크 관리의 핵심사항
- 장비별 회선의 성능을 사용률을 기준으로 측정, 분석
- 장비별 회선 Bandwidth 측정, 분석
- LAN, WAN상의 Packet Error율, Discard율 측정, 분석
- 회선 성능의 임계치 관리 방식에 따른 경보 체계 구축 및 관리

2. 인터페이스 구현 검증

2 모니터링

◆ 인터페이스 관련 모니터링 주요활동

4 응답시간 관리

- 관리대상 장비의 평균 응답시간, 최대 응답시간, 평균 Packet Loss율, 최대 패킷 손실률(Packet Loss Rate) 측정, 분석
- 지역별, 장비별, 기간별(시간별, 일별, 주간별, 월별)로 분석

핵심요약

인터페이스 구현 검증도구 및 감시도구

■ 인터페이스 구현 검증 및 감시

- 인터페이스 구현의 마지막 단계에서 구현된 인터페이스가 적절한지 테스트를 통해 구현 검증
- 인터페이스 시스템이 가동되면 시스템의 운영을 감시하고 모니터링 함

■ 테스트

- 인터페이스에 대한 테스트는 전체 테스트 시행의 일부이며, 전체 테스트 관점에서 인터페이스가 올바르게 구현되고 실행가능한지 판단하여야 함

■ 인터페이스 구현 검증도구

- 인터페이스 구현을 검증하기 위해서는 인터페이스 단위 기능 및 시나리오에 기반한 통합 테스트가 필요하며, 테스트 자동화 도구를 이용하여 단위 및 통합 테스트의 효율성을 높일 수 있음

핵심요약

인터페이스 구현 검증도구 및 감시도구

■ 인터페이스 감시도구

- 인터페이스의 동작이 잘 진행되는지 확인하기 위해서는 애플리케이션 모니터링 툴 (APM: Application Performance Management)을 사용하여 동작 상태를 감시할 수 있음

핵심요약

인터페이스 기능 구현 명세서 작성

■ 검증 및 감시도구의 요건 분석

- 구현된 인터페이스 명세서를 참조하여 구현 검증
필요한 검증 및 감시도구의 요건 분석

- ① 구현된 인터페이스 명세서의 세부 기능을 참조하여
구현 검증 및 감시에 필요한 기능을 분석
- ② 각 기능의 특징에 맞게 요건 도출

■ 검증 및 감시도구 준비

- ① 구현 검증 및 감시에 필요한 도구의 요건을 확인
- ② 시장 조사 및 솔루션 조사
- ③ 적절한 감시 및 검증에 필요한 도구를 선택 후 구매