

11주차 1차시 유지보수 종류

【학습목표】

1. 사후 유지보수의 주된 기능과 전임조직의 유지보수부문의 위치를 구분하여 설명할 수 있다.
2. 트러블 상담 및 원인 해석을 설명할 수 있으며, 고프트웨어 불량률의 수정작성 기준을 구분할 수 있다.

학습내용1 : 사후 유지보수

1. 사후 유지보수

1) 사후 유지보수란?

- 전임조직, 비전임 조직에 의해서 이루어지는 것

2) 사후 유지보수의 주된 기능

- 발생한 트러블을 해석하여 그 내용에 따라서 수정하는 것

3) 전임조직의 유지보수부문 위치



4) 사후 유지보수를 수행할 때 트러블 해결 지원작업의 흐름 요약



2. 트러블 상담

- 1) 트러블의 발생원인을 구분하여 신속하게 사용자의 업무를 정상상 태로 복구하는 것을 목적으로 실시
- 2) 트러블 상담의 단계
 - 트러블 접수
 - 문진과 구분
- 3) 트러블 접수
 - 사용자로부터 발생한 트러블을 접수하는 것으로부터 시작
 - 트러블을 접수할 때 사용자명, 시스템의 종류, 예방수정 적용 레벨, 업무에 영향 정도 등을 파악하여 트러블 관리 시스템(trouble management system)에 등록함
- 4) 문진과 구분
 - 유지보수 책임자는 접수한 트러블 내용, 사용자 업무에 영향 정도를 고려하여 가장 적합한 유지보수 요원에게 할당하여 대처토록 함
 - 이 경우 핵심적인 요소는 트러블 내용에 따라 구분
 - 하드웨어 고장에 의한 트러블인가
 - 소프트웨어 사용법의 착오에 의한 트러블인가
 - 문진(問診)을 할 때 상세한 정보를 청취해야 하는 사항
 - 어떤 시스템 환경에서
 - 어떤 조작을 해서
 - 어떤 트러블이 발생했는가
 - 긴급도는 정도(level)에 따라서 4단계 정도로 분류

긴급도		1	2	3	4
항목					
설정 기준	트러블의 영향도	시스템 운용을 계속 할 수 없다.	시스템 운용을 계속 하기에는 커다란 제한이 필요하다.	시스템 운용을 계속하는데 오퍼레이션이나 사용 프로그램등의 일부에 제한을 가하는 정도이다.	시스템의 운용을 계속하는데 거의 제한을 가하지 않고 가능하다.

긴급도		1	2	3	4
항목					
대응 기준	기술자의 파견	파견	필요에 따라 파견	-	-
	회피조치 (목표)	1일 이내에 마련한다.	3일 이내에 마련한다.	2주일 이내에 마련한다.	1개월 이내에 마련한다.

- 긴급도 설정
 - 사용자에게 대하여 문진(問診)시에 업무에 영향을 미치는 정도를 감안하여 담당자가 결정하는 것이 원칙
- 행위계획
 - 트러블 해결을 위한 행동계획
 - 관계자들의 문제인식을 일치시킬 수 있으며 트러블 해결을 위한 행동계획이 명확해지기 때문에 트러블 조기 해결에 유효

분류	내용
원인 규명을 위한 행위계획	<ul style="list-style-type: none"> • 누가, 언제까지 어떤 방법으로 원인규명 작업을 할 것인가 • 위의 기간에 원인규명이 되지 않을 경우에 누가 원인 규명을 위한 대책을 수립하는가
소프트웨어 불량 수정을 위한 행위계획	<ul style="list-style-type: none"> • 누가 언제까지 소프트웨어 불량수정을 하여 사용자에게 제공하는가 • 일반 사용자에게 불량 수정의 제공 스케줄의 결정

5) 회피조치

- 트러블 처리에서 가장 중요한 사항
 - 시스템 전체를 장시간 정지하지 않고 트러블을 극소화시켜 가능한 범위 내에서 업무를 다시 시작하는 것
- 트러블 처리의 단계와 회피조치



- 구체적으로 트러블에 대응한 회피대책은 다음과 같은 관점에서 수립
 - 트러블 현상 확인 수준에서 회피(avoidance) 대책 : 트러블 원인은 불명확하지만 현상을 해석할 수 있다면 조작방법을 변경하는 것과 같은 조치로 운용의 측면에서 트러블을 회피하는 대책을 수립할 수 있음
 - 트러블 원인 확인 수준에서 회피대책
 - 트러블 발생원인이 명확하게 해석되는 경우 운용 제한 등으로 트러블의 회피가 가능함
 - 그리고 트러블 내용이 정확하게 분류가 된 경우 원인 별로 대책을 수립해 두어야 함.

- 트러블 원인 확인 수준에서 회피대책 : 소프트웨어 트러블의 조치

트러블 원인	조치 내용
하드웨어 장애	• 하드웨어 엔지니어에 의한 조사
소프트웨어 사용법의 착오	• 착오의 설명, 오해의 해소
검출 완료된 소프트웨어 불량	<ul style="list-style-type: none"> • 수정 데이터가 이미 제공 완료된 경우에는 어떤 수정을 적용해야 하는가를 통지한다. • 수정데이터가 아직 제공되지 않은 경우에는 수정 데이터를 작성 및 제공한다.

트러블 원인	조치 내용
신규 소프트웨어 불량	<ul style="list-style-type: none"> • 회피수단의 통지 • 수정 데이터의 작성
매뉴얼 불완전	<ul style="list-style-type: none"> • 올바른 명세서 및 대책의 설명 • 매뉴얼 정정처리
기타 (신규 소프트웨어에 불량의 의문이 있는 경우)	<ul style="list-style-type: none"> • 필요한 데이터의 수집지시 • 대체방법 (다른 기능)의 제시 • 원인의 상세한 조사 지시 • 잠정적인 수정의 제공

3. 원인 해석

- 1) 트러블 발생시에 나타난 데이터 중에서 유효하게 사용할 수 있는 것

- 메모리 덤프(memory dump)
 - 트러블이 발생할 당시에 시스템의 상태를 파악하는데 매우 유효한 정보
- 트레이스 데이터(trace data)
 - 회선이나 입출력 계통, 오퍼레이팅 시스템의 매크로(macro)등의 명령(command)의 출력 상태를 추적하는 것
 - 하드웨어에 관계되는 회선이나 입출력 계통의 트러블을 구분하는데 유효
- 로깅 데이터(logging data)
 - 소프트웨어 및 하드웨어 에러 정보 등의 로깅 데이터는 시스템의 이상이 발생한 상황을 파악하는데 유효함
- 파일 덤프(file dump)
 - 문제가 있는 파일 내용을 해석하는데 유효하게 사용됨

4. 소프트웨어 불량률의 수정

- 1) 트러블의 원인이었던 소프트웨어 불량을 제거하고 정확히 수정되었는지 여부를 확인한 다음 사용자의 업무를 차질 없이 운용토록 하는 것

2) 수정 데이터의 작성

- 소프트웨어 불량으로 판명되는 경우 그 불량이 사용자에게 주는 영향을 고려하여 수정 작업을 수행
- 수정은 그 불량 성격에 구별하여 대응하는 것이 효율적
 - 전체에 영향을 미치기 때문에 긴급대처를 요하는 불량
 - 긴급대처를 요하며 사용법이 특정화되는 불량
 - 그다지 영향을 미치지 않는 불량
- 수정을 작성하는 방법
 - 잠정 수정
 - 정식 수정
- 소프트웨어 불량 수정 작성 기준

수정의 종류	수정 방법	제공 시기	제공 범위	목적
잠정적 수정	Batch 수정	수시	트러블이 발생한 사용자 또는 모든 사용자	트러블 회피
정식 수정	소스 프로그램 레벨의 수정	수시	모든 사용자	예방 유지보수

3) 확인

- 데스크에서 확인
 - 객관적 판단이 가능하도록 최소한 두 사람 이상이 참여하여 수정 데이터의 확인
 - 수정 방법은 명세서에 문제가 없는지 여부를 점검
 - 수정 데이터는 소프트웨어 불량을 수정하는데 충분한지 여부를 검토해야 함
 - 수정데이터는 다른 것에 악영향을 미치지 않는지 여부를 분석 검토해야 함
- 기계에 의한 확인
 - 데스크에서 확인을 충실하게 이행한 다음에는 기계 (machine)로 수정 데이터의 확인 테스트를 실시
 - 소프트웨어 불량은 수정 가능한 것인가
 - 다른 것에 영향은 없는가
 - 테스트 방법의 검토, 사용자에게 의한 확인방법을 검토해야 함
 - 기계에 의한 확인 테스트는 사용자 시스템을 활용하여 실시
 - 수정이 다른 부분에 영향을 미치지 않음을 확인하는 테스트
 - 테스트의 자동화가 되도록 하여 각종 패턴(pattern)의 테스트가 자동적으로 이루어지도록 테스트 세트 (test set)를 만들어 두는 게 효과적임

- 수정한 내용을 제공하면 개발부문에서 테스트, 테스트 전담부서에서 테스트를 거쳐서 완전한 수정 여부를 확인함
- 수정의 내용을 철저히 정리하기 위해서 중요사항을 정리
 - 유지보수용 문서의 충실과 갱신관리
 - 소스 코드의 갱신관리
 - 테스트 환경정비 (테스트의 자동화)
 - 수정작업 절차의 표준화

【학습정리】

1. 사후 유지보수를 이해한다.
2. 트러블 상담을 알아본다.
3. 원인 해석을 파악한다.