

# 신뢰성 데이터 수집 및 분석을 위한 SPR/SCR 관리 시스템의 설계 및 구현

이 창 희<sup>†</sup> · 한 판 암<sup>††</sup>

## 요 약

소프트웨어를 개발하고 운영하는 동안에 직면하는 문제들은 소프트웨어나 하드웨어 또는 운영의 결함에 기인한다. 이들의 다양성 때문에, 문제들을 조정하고 체계적으로 원인을 규명하기 위한 문제 발생의 규명 및 수정 작업을 중점적으로 관리하는 시스템이 요구되고 있다.

본 논문에서는 소프트웨어 개발 수명 주기(Software Development Life Cycle:SDLC)에서 획득할 수 있는 결함 데이터(Fault Data), 프로세스 데이터(Process Data), 프로덕트 데이터(Product Data)를 수집하고 분석하기 위한 소프트웨어 문제 보고서(Software Problem Report:SPR)와 소프트웨어 변경 보고서(Software Change Report:SCR)를 처리하는 SPR/SCR 관리 시스템을 개발하고 적용하였다.

SPR/SCR 관리 시스템의 목적은 4 가지가 있다. 첫째로 모든 소프트웨어 결함들을 문서화하고, 수정하며, 무시되지 않는다. 둘째로 SPR의 타당성에 대해 평가된다. 셋째로 SPR의 진행 상태를 개발자 또는 사용자에게 피드백한다. 넷째로 소프트웨어 품질과 신뢰성을 예측하고 측정하기 위한 기본 자료를 제공한다. 이들 목표는 SPR/SCR 관리 시스템의 적용에 의해 충족시킬 수 있다. 또한 소프트웨어 신뢰성의 달성 비율, 테스트의 종료 기준, 릴리즈 시기의 예측, 효율적인 개발 관리에 반영될 수 있다.

## The Design and Implementation of the SPR/SCR Management System for Reliability Data Collection and Analysis

Chang-Heui Lee<sup>†</sup> · Pan-Am Han<sup>††</sup>

### ABSTRACT

Problems encountered during software development or operation may result from defects in the software, hardware, or their operations. Because of their diversity, the determination of the sources of a problem and its corrective action requires a centrally controlled system for monitoring problems and determining systematic causes.

In the paper, the SPR/SCR management system was developed and applied for the collection and analysis of the error data, process data and product data detected by each phase in SDLC.

There are four objectives in the SPR/SCR management system. The first is to assure that software errors are documented, corrected, and not forgotten. The second is to assure that SPR are assessed for their validity. The third is to provide feedback on SPR status to the developer and the user. The fourth is to provide data for measuring and predicting software quality and reliability. These goals could be satisfied by the application of the

† 정 회 원:진주간호보건전문대학 사무자동화과 조교수  
†† 종신회원:경남대학교 공과대학 컴퓨터공학과 교수  
논문접수:1996년 9월 17일, 심사완료:1996년 12월 18일

SPR/SCR management system. Also, the accomplishment rate of software reliability, criterion of test completion, estimation of release time, efficient development and management can be reflected by applying the SPR/SCR management system.

### 1. 서 론

최근의 컴퓨터 시스템은 사람의 생명을 맡기는 철도나 항공기의 운행 관리, 자동차의 제동 제어 등을 포함한 개인의 재산 관리, 기업의 생산 활동, 지역 사회와 국가의 안보 및 방위 등의 다양한 분야에서 활용되고 있다[1].

이와 같은 상황에서 컴퓨터 시스템에 대한 기대가 크면 클수록 소프트웨어의 비중이 계속 증대되고, 시스템의 신뢰도를 좌우하는 핵심 요소가 소프트웨어로 인식됨에 따라서 소프트웨어 신뢰도에 관한 연구가 소프트웨어 공학의 일부분으로서 활발히 진행되고 있다[2, 3].

이를 위해 종래에는 SDLC 과정의 테스트 단계에서 발견되는 결함 데이터, 즉 소프트웨어 결함 건수를 이용한 연구가 활발하였지만, 최근에는 결함 데이터, 프로세스 데이터와 프로덕트 데이터를 모두 고려한 연구가 강조되고 있다[3, 4]. 특히 소프트웨어를 개발하고 운영하는 동안에 직면하는 문제들을 조정하고 체계적으로 문제 발생의 원인을 규명하며 수정 작업을 중점적으로 관리하는 시스템이 요구되고 있다[5].

그러나 지금까지 국내의 소프트웨어 개발이 사용자와 개발자가 원하는 시스템을 조기 구축에만 치중하고 관리적인 측면은 부담으로 여기거나 경시하는 경향이 있기 때문에 SDLC 과정에서 획득할 수 있는 결함 데이터, 프로세스 및 프로덕트 데이터의 체계적인 수집이 현실적으로 어려우므로 다른 분야에 비해 필요성은 크지만, 프로젝트 개발 현장에 직접 적용할 수 있는 연구가 미비한 실정이다.

본 논문에서는 소프트웨어 신뢰도를 측정 및 예측하기 위하여 SDLC 과정에서 획득할 수 있는 결함 데이터, 프로세스 데이터, 프로덕트 데이터를 체계적으로 수집하고 분석하기 위한 SPR/SCR 관리 시스템을 개발하여 적용하였다.

SPR/SCR 관리 시스템은 첫째로 모든 소프트웨어 결함들을 문서화하고, 수정하며, 무시되지 않는다. 둘째로 SPR의 타당성에 대해 평가된다. 셋째로 SPR의

진행 상태를 개발자 또는 사용자에게 피드백한다. 넷째로 소프트웨어 품질과 신뢰성을 예측하고 측정하기 위한 기본 자료를 제공하는데 그 목적이 있다[5, 6].

이와 같은 목표 달성을 위해서 본 논문의 2장은 소프트웨어 신뢰성 데이터인 결함 데이터, 프로세스 및 프로덕트 데이터를 조사하여 기존 방안에 대한 특징과 문제점을 비교 분석하고, 3장은 문제점을 보완한 SPR/SCR 관리 시스템을 제안하여 그 기대 효과를 제시하고, 제안된 시스템을 설계하고 구현하였으며, 4장은 SPR/SCR 관리 시스템을 적용한 사례 연구의 결과를 요약하고 검토하였다. 5장은 결론 및 향후 연구 과제에 대해 기술한다.

### 2. 이론적 배경 및 고찰

신뢰성 데이터에 대한 수집은 소프트웨어 신뢰도를 예측하고 측정하기 위해 필수적으로 선행되어야 하므로 SDLC 과정에서 결함 데이터, 프로세스 및 프로덕트 데이터가 수집되어야 한다.

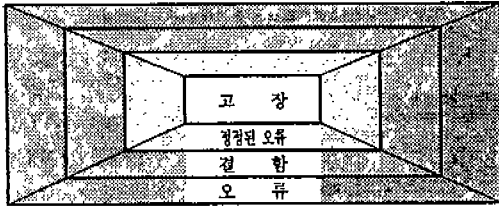
#### 2.1 신뢰성 데이터 조사

##### (1) 소프트웨어 결함 데이터

소프트웨어 오류(Software Error)는 요구 사항의 잘못 이해, 또는 설계 및 코딩 오류에 의해 발생된 것으로 개발자의 실수로 인한 설계 오류를 말한다. 소프트웨어 결함(Software Fault)은 소프트웨어 오류에 의해 발생된 것을 의미하며, 소프트웨어 고장(Software Failure)은 시스템 고장을 유발하는 소프트웨어 결함을 의미한다. 정정된 오류(Corrected Error)는 소프트웨어 오류가 소프트웨어 결함으로 결정되어 수정이 완료된 것을 말한다. 이들 관계를 그림 1과 같이 나타낼 수 있다[4]. 여기서 소프트웨어 오류, 결함, 고장을 "결함"이라는 개념으로 통일하여 사용하는 것으로 한다.

소프트웨어 결함 데이터는 신뢰성을 분석하고 연구하기 위해 SDLC의 각 단계에서 수집되어야 한다. 소프트웨어 신뢰성의 한 부분인 결함 데이터 수집은

표준 분류법에 적용될 수 있는 결함 발생 단계, 결함 원인, 성능 효과와 같은 특성들에 초점을 두고 진행되어야 한다[7, 8].



오류의 수는 단지 추정될 수 있다.  
 - 오류의 수 → 결정된 오류의 수  
 - 오류의 수 → 결함의 수  
 - 오류의 수 → 고장의 수

(그림 1) 소프트웨어 오류, 정정된 오류, 결함, 고장  
 (Fig. 1) Software errors, corrected errors, faults, and failures

① 결함 발생 단계

소프트웨어 결함은 표 1에 나타난 것처럼 요구 사항, 설계, 코딩, 유지 보수 단계에 따라 분류할 수 있다[7].

<표 1> 소프트웨어 결함 발생 단계  
 <Table 1> Software error source phase categories

기호	발생 단계	결함내용
RQ	요구 사항	하나의 변경 사항, 부적당한 표현, 또는 빈약한 요구 사항을 표현
DE	설계	개략 설계 또는 상세 설계에서의 잘못된 문제
CO	코딩	코드 구현에서 만들어진 결함
MN	유지 보수	문서의 결함을 수정하든지 또는 하나의 변경을 구현하는 과정에서 발생한 결함

② 결함 원인 분류

소프트웨어 결함은 표 2에 나타난 것과 같이 원인에 따라 분류할 수 있다. 프로젝트 관리자의 입장에서 단지 중요한 몇 가지 종류만을 사용할 수 있지만, 대형 프로젝트에서는 결함 종류를 아주 상세하게 제안하고 있다. 이들 결함 종류의 목적은 결함의 증상보다 원인을 식별하는데 있다[4, 7, 9, 20].

<표 2> 소프트웨어 결함 원인 분류  
 <Table 2> Software error cause categories

기호	소프트웨어 결함 원인
A00	계산 오류
A10	연산식에서 부적당한 연산자/피연산자 사용
A20	부호(Sign) 규칙 오류
A30	부적당한/부정확한 연산식 사용
A40	혼합 연산 방식으로 인한 정밀도 상실
A50	잘못된 계산
A60	자리 올림 또는 자리 절삭 오류
B00	논리 오류
B10	논리적 표현에서 부적당한 연산자/피연산자 사용
B20	순서를 벗어난 논리적인 범위 지정
B30	잘못된 변수가 점검됨
B40	누락된 논리 또는 조건 테스트
B50	부적당한 횡수를 반복하는 루프(무한 루프 포함)
B60	중복된 논리
C00	데이터 정의 오류
C10	데이터가 적절하게 초기화되지 않음
C20	부적당한 데이터 단위 또는 스케일링(Scaling)
C30	부적당한 변수 형태
D00	데이터 취급 오류
D10	부적절한 데이터 참조 또는 저장
D20	프래그 또는 인덱스가 부적절하게 사용된 변수
D30	비트(Bit) 조각이 부적당하게 수행됨
D40	부적당한 변수 형태
D50	데이터 패킹/언패킹 오류
D60	단위 또는 데이터 변환 오류
D70	철자 오류
E00	설계 오류
E10	요구 사항에 대해 불확실함
E20	불일치한 또는 불완전한 데이터 베이스
E30	부적당한 또는 불완전한 인터페이스
E40	부적당한 또는 불완전한 프로그램 구조
E50	전혀 무시된 조건
F00	인터페이스 오류
F10	잘못된 또는 존재하지 않은 서브루틴 호출
F20	일치하지 않은 매개변수의 서브루틴 호출
F30	루틴에 의해 데이터베이스의 부적당한 사용/세팅
F40	부적당한 인터럽트의 취급
G00	기타 오류
G10	새로운 소프트웨어 환경
G20	시스템/하드웨어 변경
G30	프로젝트 표준에 적합하지 않음
G40	리스팅 프롤로그/주석을 사용하지 않음
G50	비효율적/불필요한 코드 또는 설계

G60	오퍼레이터 오류
G70	사무 직원 오류
G80	문제 보고서 각각
G90	관련된 문서 오류

또한 결함 원인 종류별로 발생 빈도를 Motley and Brooks[10], Lipow[11] 및 Pointer *et al.*[12]의 조사 결과에서 대부분 결함은 설계 단계에서 혼입된 논리 오류의 발생 빈도가 높은 것을 알 수 있다.

③ 성능 효과 분류

소프트웨어 결함에 대한 성능 효과(심각성)는 표 3에 나타난 것처럼 중대함(Critical), 중요함(Major), 경미함(Minor), 영향 없음(None)으로 분류할 수 있다. 이들 분류에 대한 심각성은 모든 소프트웨어 결함에 관련되는 의미를 제공하고 있다[4, 7].

〈표 3〉 소프트웨어 결함에 대한 성능 효과  
 〈Table 3〉 Performance effect categories for software errors

기호	성능 효과	결함내용
CR	중대함	시스템의 조기 종료 또는 임무 수행의 여러 가지 중대한 영향을 주는 고장
MA	중요함	임무 수행에 중요한 영향을 주는 부적당한 값 또는 출력 정보
MI	경미함	임무 수행에 경미한 영향을 주는 부적당한 값 또는 출력 정보
NO	영향 없음	임무 수행에 영향을 주지 않는 결함

④ 결함 데이터 매트릭스

소프트웨어 결함 데이터의 수집은 소프트웨어 신뢰도를 측정하고, 개선하기 위해 매우 중요하다. 표 4는 결함 데이터 매트릭스를 요약한 것이다[4, 13].

〈표 4〉 소프트웨어 결함 데이터 매트릭스 요약  
 〈Table 4〉 Summary of software error data metrics

메트릭스	참고사항
- 결함 발견 일자	- 결함 발견 시간
- 결함 내용 상세 서술	- 결함 내용 영역 기술
- 결함 발견 요원	- 결함 처리 상태
- 결함 식별자	- 우선 순위/심각성

소프트웨어 문제/ 변경 보고서	- 결함 내용 수정 일자	- 결함 내용 수정 요원
	- 유지 보수 활동 조치	- 수정된 모듈 수
	- 수정된 모듈 명칭	- 버전 관리 정보
	- 결함 수정 내용 기술	- 결함 수정 소요 시간
	- 수정 내용 검증 일자	- 수정 내용 검증 요원

결함 발견 능력 건수	- 소프트웨어 안정성에 시각적인 지침서인 동시에 결함을 및 신뢰성을 결정하기 위해 사용됨
결함 수정 능력 건수	- 수정 작업을 요구하는 결함에 대해 얼마나 이해하고 있는가를 결정하는데 사용됨
결함 검증 능력 건수	- 결함이 수정되어 검증되었기 때문에 중요한 것임. 즉 문제가 해결되어 완료된 건수임.
결함 발견 비율	- 결함의 경향을 나타내는데 사용됨.
결함 수정 비율	- 알려진 결함이 얼마나 수정되었지를 나타내기 위해 결함 발견 비율이 사용될 수 있음.
영역당 결함 건수	- 한 영역에서 많은 결함의 경향이 있는지를 결정하기 위해 기능 또는 서브 기능에 따라 결함이 추적됨.
성능 효과(심각성)	- 결함이 임무 수행에 대해 잠재적인 것인지 또는 안정성을 파괴시키는 것인지를 나타냄.
영역당 구조적 복잡도	- 결함을 발생시키는 영향이 무엇인지를 결정하기 위해 다른 매트릭스와 함께 사용됨.
영역당 기능적 복잡도	- 결함을 발생시키는 영향이 무엇인지를 결정하기 위해 다른 매트릭스와 함께 사용됨.

(2) 소프트웨어 프로세스 데이터

많은 신뢰도 모델들은 소프트웨어 프로세스 데이터를 사용하고 있다. 수집되어야 하는 데이터는 개발, 코딩, 테스트 및 유지 보수하는데 소요되는 시간의 양이다. 표 5는 프로세스 데이터 매트릭스를 요약한 것이다[4, 13].

〈표 5〉 소프트웨어 프로세스 데이터 매트릭스 요약  
 〈Table 5〉 Summary of software process data metrics

메트릭스	참고사항
단계별 발견된 결함 건수	- 결함이 발견됨에 따라, 결함이 수행 주기의 어떤 단계에서 발생하였는지를 결정할 수 있음.
단계별 발견된 결함 건수	- 결함이 발견된 단계를 추적함에 따라, 발생한 버그를 발견하는 것을 지연시킬 것 인지를 결정할 수 있음.
총 분석, 설계, 코딩 소요 시간	- 수행 주기의 자원을 사용할 단계를 결정하기 위해 사용됨.
총 단위 테스트 소요 시간	- 수행 주기의 자원을 사용할 단계를 결정하기 위해 사용됨.

총 시스템 테스트 소요 시간	-수명 주기의 자원을 사용할 단계를 결정하기 위해 사용됨.
총 유지 보수 소요 시간	-소요 시간은 모든 유지 보수 활동의 시간을 합하여 계산됨.
평균 유지 보수 관리 시간	-결함이 수정되기 전후에 관리 임무를 수행하는 시간이 얼마나 많이 소요되었는지를 나타냄.
평균 수정 작업 시간	-수정 작업의 평균 소요 시간은 결함을 추적하고, 결함을 해결하는 소요 시간이 포함됨.
수정 작업에 대한 이유	①사전 유지 보수 활동    ②새로운 요구 사항    ③요구 사항 변경 ④요구 사항 잘못 이해    ⑤요구 사항 누락    ⑥예외 모호한 요구 사항 ⑦소프트웨어 환경 변경    ⑧하드웨어 환경 변경    ⑨코드상의 오류 ⑩성능 오류    ⑪기타
최대, 최소, 평균 수정 작업 시간	-최대, 최소, 평균 수정 작업 시간은 정해진 기간당 유지 보수 활동의 분포에 따라 기록 추적됨.
일정표, 시작 및 인도 일자	-주요 일정표가 만족되는지, 야간작을 결정하기 위해 모든 프로젝트에서 필요함.
역사적인 데이터	-가장 중요한 데이터의 구성 요소를 수집한 것임.
검열 및 검토 소요 시간	-설계, 코드 검열 및 검토를 수행하는 총 소요 시간을 추적하는 것임.
테스트 및 검증된 기능 비율	-테스트 범위 매트릭스로서, 테스트가 효율적인지를 결정하기 위해 사용됨.
테스트 및 검증된 독립경로 비율	-테스트 범위 매트릭스는 구조적 테스트가 충분히 완료되었는지를 결정하는 것임.
테스트 및 검증된 원시코드 비율	-테스트 범위 매트릭스는 모든 코드가 테스트에 포함되었는지를 결정하는 것임.

<표 6> 소프트웨어 프로덕트 데이터 매트릭스 요약  
<Table 6> Summary of software product data metrics

매트릭스	참고사항
하나 이상의 기능을 수행하는 모듈의 개수와 비율	-전체 설계의 응집도(Cohesion)를 나타냄.
높은 구조적 복잡도를 가지는 모듈의 개수와 비율	-전체 설계가 복잡성을 감소하기 위해 재 설계해야 할 필요성 여부를 나타냄.
하나의 입구와 출구를 가지는 모듈의 개수와 비율	-응집성있는 설계의 지표로서 아주 훌륭한 것임.
표준화된 규격에 따라 문서화된 모듈의 개수와 비율	-유지 보수성 문제를 대두시킬 때 사용될 수 있음.
작성된 요구 사항을 추적 가능한 모듈의 개수와 비율	-코드가 모든 요구 사항을 포함하고 있는지를 결정하기 위한 것임.
재사용 코드에서 발견된 결함의 개수와 비율	-재사용 코드를 사용할 것인지에 결함의 개수와 비율

에서는 소프트웨어 결함을 추적 및 보고할 수 있는 시스템이 요구되고 있다.

따라서 프로젝트 관리자가 당면하고 있는 문제 중에서 가장 중요한 것은 잔존하는 결함을 파악하여 미래의 소프트웨어의 신뢰도를 예측하고, 궁극적으로는 경제적인 테스트 종료 시기를 결정하여 운용 시기를 예측하며, 사용자에게 개발된 소프트웨어를 인도하는 것이다.

### (3) 소프트웨어 프로덕트 데이터

소프트웨어 프로덕트 데이터는 개발되는 소프트웨어의 크기, 기능성, 복잡성 등의 특성을 지정한 것이다. 프로덕트 데이터 항목은 실행 가능한 코드의 수를 추정할 수 있는 것으로 수집해야 한다. 프로덕트 데이터 매트릭스는 하나의 명세서를 추적할 수 있는 기능의 수, 전체 기능의 수, 전체 호출 루틴의 수, 전체 데이터 참조의 수가 있다. 주요한 매트릭스는 모듈 또는 기능당 순환 복잡도(Cyclomatic Complexity)이다. 이것은 소프트웨어 각 모듈의 구조적 복잡도(Structural Complexity)이다[14, 15]. 표 6에 나타낸 것은 프로덕트 데이터 매트릭스를 요약한 것이다[4, 13].

요약하면 결함 데이터, 프로세스 데이터, 프로덕트 데이터의 수집은 소프트웨어 신뢰도를 측정할 뿐만 아니라 신뢰도를 개선시키는데 매우 중요하다. 전형적으로 많은 양의 결함이 발생하는 대형의 프로젝트

### 2.2 기존 방안 분석

소프트웨어 개발 과정에서 발생하는 SPR/SCR 데이터를 기록하기 위한 방법은 수작업 기록(Hand Logging)에서 컴퓨터 기록(Computer Logging)까지 서로 다른 종류들이 많이 있다. 일반적으로 각각의 프로젝트들은 개별적인 요구 사항을 만족시킬 수 있는 방안을 제안하고 있다[7, 13, 16, 17, 18].

그러나 모든 프로젝트에 적용될 수 있도록 입, 출력 양식들과 처리 절차를 표준화하고, 자동화된 SPR/SCR 시스템을 개발하여 운영할 필요가 있다. 또한, SPR/SCR 상태 보고는 대형 프로젝트의 성공에 중요한 역할을 담당한다. SPR/SCR 상태 보고는 첫째로 무슨 일이 발생했는가? 둘째로 누가 그것을 수행했는가? 셋째로 언제 발생했는가? 넷째로 어떠한 것들이 영향을 받게 될 것인가? 라는 등의 의문점을 해결하

는 중요한 활동이다[19]. 이와 같은 의문점을 해결해주는 대표적인 방법은 Trivedi-Shooman[18], Shooman-Bolsky[17], Bowen-Intire[7, 16], Grady-Caswell[13]에 의한 방안들이 있다.

위의 대표적인 방안들을 결합 기록 방법(수작업 기록, 컴퓨터 기록 등), SPR 양식을 작성할 때 고려되어야 하는 주요 항목인 결함 발견 단계(분석, 설계, 코딩, 단위 테스트, 통합 테스트, 확인 테스트, 시스템 테스트, 인수 테스트, 유지 보수 등), SPR 처리 상태(개방, 수정, 해결, 완료 등), 성능 효과 분류(중대함, 중요함, 경미함, 영향 없음 등)와 SCR 양식을 작성할 때 고려되어야 하는 주요 항목인 결함 원인 분류(표 2 참조), SCR 처리 상태(검증 통과, 검증 오류 등), 문서화 영향(영향 있음, 영향 없음 등), SPR/SCR 처리 절차의 제공 여부, 제공되는 출력 정보(상태 보고서, 통계 보고서 등), 프로세스 및 프로덕트 데이터의 고려 여부 등을 비교 분석한 결과는 표 7과 같다.

<표 7> 대표적인 방안의 비교 분석

<Table 7> Comparing and analyzing representative methods

항목 \ 방안	Trivedi-Shooman 방안	Shooman-Bolsky 방안	Bowen-Intire 방안	Grady-Caswell 방안
결함 기록 방법	수작업 기록	수작업 기록	컴퓨터 기록	수작업 기록
결함 발견 단계	테스트 단계 적용	테스트 단계 적용	전 수명주기 단계 적용	테스트 단계 적용
SPR 처리 상태	○	×	○	○
성능 효과 분류	×	○	○	○
결함 원인 분류	4가지로 분류	13가지로 분류	40가지로 분류	10가지로 분류
SCR 처리 상태	○	×	○	○
문서화 영향 여부	×	×	○	×
SPR/SCR 처리 절차	○	×	○	○
상태 보고서 제공	○	×	○	○
통계 보고서 제공	○	○	○	○
프로세스 데이터 제공	×	○	×	×
프로덕트 데이터 제공	×	○	×	×

(○: 고려되고 있음, ×: 고려되지 않음)

표 7을 살펴보면 기존의 방안들은 대부분 수작업으로 이루어져 왔는데 이의 비효율적인 면(문제점을 들면 첫째로 신뢰성 데이터(결함, 프로세스 및 프로덕트 데이터)의 세부적인 종류가 다양하고 방대하므로 소프트웨어 문제들에 필요한 문서들을 보존, 관리 및 검색하는데 많은 요원이 필요하다. 둘째로 일상적이고 반복적인 일이 많아 실수를 유발하기 쉬우며, 시간이 많이 소비된다. 셋째로 SPR/SCR의 주요 내용이 정보 처리 시간에 비추어 오래된 것이기 쉽다. 넷째로 요구되는 신뢰성 정보를 필요한 때에 즉시 제공하기가 어려워 관리의 가시성이 없다. 다섯째로 SPR/SCR 작성 과정이 매우 지루한 것이 되기 쉬우므로 생각하고 넘어가는 경우가 많다. 여섯째로 불필요한 이중 작업이나 이중 표기가 많아지기 쉽다. 일곱째로 현재의 정보가 들어가 있지 않은 잘못된 버전이 사용되거나, 이전의 버전이 새로운 버전으로의 작업이 자동적으로 되지 않아 정보의 제어가 곤란해진다.

상기의 이유 등으로 공통적인 문제점을 보완하여 모든 프로젝트에 적용될 수 있도록 입·출력 양식과 처리 절차를 표준화하고 자동화된 SPR/SCR 관리 시스템을 개발하는 것이 필요하다.

### 3. SPR/SCR 관리 시스템 개발

#### 3.1 시스템 제안

SPR/SCR 관리 시스템은 소프트웨어 제품 변경의 준비, 조정, 인가, 실행을 위한 방법을 확립하는데 있다. 또한 SPR/SCR 추적과 관리에 도움을 줄 수 있는 상태 보고, 자동화 도구의 추천 방법 등이 포함되어 있다. SPR/SCR의 사용은 모든 시스템의 구성 관리(Configuration Control)를 유지하기 위해 필수적인 활동이며, 사용자에게 인도되기 전에 자체 테스트, 현장 설치, 성능 시험 동안에 발생하는 결함의 효과적인 해결책을 보증하는 것이다.

#### (1) SPR 양식

SPR은 다중-복사 양식으로 문제 중심의 보고서이다. 그래서, 각각의 SPR은 하나의 문제를 기술해야 한다. 하나 이상의 문제를 하나의 SPR에 보고하는 것은 허용되지 않는다. 각각의 SPR 양식은 SPR 처리 담당자가 SPR 번호를 배정하기 전에 제안자가 추적

할 수 있도록 미리 번호를 부여한다. SPR의 관리 번호는 순서적으로 배정되며, 1000으로 시작하여 4자리 숫자로 구성한다. SPR 양식은 그림 2에 나타난 것과 같다.

① 제안자 부분

- 제안자 - SPR를 제안한 사람의 성명을 기재한다.
- 프로젝트 명칭 - 문제가 발생한 프로젝트의 명칭을 기재한다.
- 발견 일자 - 문제 발견 일자를 기재한다.
- SPR 번호 - 라이브러리인 또는 구성 관리 요원에 의해 순서적인 관리 번호가 배정된다.
- 제목 - 최대 40 문자를 사용하여 문제의 제목 명칭을 정확하고, 간결하게 기술한다.
- 컴퓨터 명칭 - 문제가 발생한 컴퓨터 명칭을 기재한다.
- 발견 버전 - 문제가 발생한 시스템 버전 번호를 기재한다.
- 문제 증상 서술 - 제안자가 문제를 상세하게 기술하는 것이 아니라, 문제를 연구, 조사하고 개선하기 위한 정보를 제공한다.
- 문제 유형 - 문제가 프로그램에 있는지, 또는 문서화에서 불일치 되는지를 나타낸다. 명세서와 프로그램에서 결함이 발생한 경우 해당 박스에 체크한다.
- 문제 반복 여부 - 문제가 동일한 작업 순서에서 반복적으로 발생하는지를 나타낸다.
- 제공 자료 존재 - SPR과 함께 문제를 해결하는데 도움을 주는 자료가 제공되는지를 나타낸다.
- 참고 명세서 - 프로그램의 변경에 영향을 주든지, 또는 명세서에 영향을 준다면, 명세서 제목과 해당 문장의 문단 번호를 제공한다.
- 성능 효과 - 발생한 문제가 시스템의 성능 효과

(심각성)에 영향을 주는 정도를 나타낸다.

- 발견 단계 - 문제가 발견된 프로젝트의 해당 단계를 나타낸다.
- 기능 영역 - 제안자가 생각하는 문제 해결의 기능 영역을 기술한다.
- 승인 - 문제 서술에 승인하는 것으로 제안자 팀장의 서명 또는 날인을 나타낸다.
- 해결 권고 사항 - 제안자가 발생 문제의 해결안을 제시할 수 있다면, 해결 권고 사항을 기술한다.

소프트웨어 문제 보고서

		PAGE OF	
제안자	프로젝트 명칭	발견 일자	SPR 번호
제목		컴퓨터 명칭	발견 버전
제 안 자	문제 증상 서술		
	문제 유형	<input type="checkbox"/> 프로그램 <input type="checkbox"/> 문서화	참고명세서 _____
	문제 반복 여부	<input type="checkbox"/> YES <input type="checkbox"/> NO	상용 효과 _____
	제공 자료 존재	<input type="checkbox"/> YES <input type="checkbox"/> NO(YES이면 자료 첨부)	발견 단계 _____
	기능 영역	승 인 _____	
해결 권고 사항			프로그램어
S C R B	소프트웨어 변경 검토 위원회		
	<input type="checkbox"/> 승 인		<input type="checkbox"/> 기 각
승 인 _____			승인 일자 _____
비고란			

(그림 2) 소프트웨어 문제 보고서 양식 (Fig. 2) Software problem report form

② SCR B 부분

- 승인 - 문제 서술에 동의하는 것으로 승인란에 체크되며, SCR B 위원장의 서명 또는 날인을 한다. 이것은 수정 작업이 진행된다. SPR이 기각될 수도 있다.
- 기각 - SCR B가 SPR을 기각란에 체크하면, 기각된 이유에 대해 비교란에

- 기술한다.
- 조정자 - 문제 해결을 위해 배정된 팀장 또는 요원의 성명을 기재한다.
- 승인 일자 - SCR이 승인 또는 기각된 일자를 기재한다.
- 비고란 - SPR이 기각된 이유를 서술한다.

(2) SCR 양식

SCR은 SPR 양식의 연장으로 소프트웨어 제품의 모든 변경을 서술하고 문서화하는데 사용된다. SCR은 컴파일/어셈블리 중심의 보고서이다. 만약 문제를 해결하기 위한 변경이 여러 개의 컴파일/어셈블리 작업이 요구된다면, 이것은 요구되는 만큼의 추가적인 SCR을 작성하는 것은 해당 프로그래머의 임무이다. SCR은 SPR에 관련되어 있기 때문에, 각각의 SCR은 하나의 문제를 해결하게 된다. 하나의 SCR을 가지고 하나 이상의 문제를 해결하는 것은 허용되지 않는다. SCR은 SPR에 첨부되어 해결된다. SCR 번호는 SPR에 배정된 4자리 숫자에 첨가하여 “-”와 3자리 숫자가 뒤에 부여된다. 예를 들면, SPR 번호가 1000이면, -100, -200, -300 등을 배정한다. SCR은 해결자, 라이브러리언 및 검증자 부분으로 구분된다. SCR은 해결자, 라이브러리언 및 검증자 부분으로 구분된다. 해결자 부분은 SPR을 해결하기 위해 배정된 프로그래머가 문제에 대해 해결안을 제시하는 것이며, 라이브러리언 부분은 제시된 해결안을 수정하는데 필요한 사항을 기재하는 것이며, 검증자 부분은 수정 작업이 완료된 문제에 대해 검증 작업 후 그 검증 결과를 제시한다. SCR 양식은 그림 3에 나타난 것과 같다.

① 해결자 부분

- 프로그래머 - 문제를 수정하기 위해 배정된 프로그래머 또는 요원의 성명을 기재한다.
- 제출 일자 - 문제 해결 일자를 기재한다.
- SCR 번호 - SPR 번호 다음에 각각의 SCR에 순서적으로 숫자 010부터 시작하여 추가한다.
- 제목 - SPR 양식의 제목과 동일하다.
- 컴퓨터 명칭 - 문제가 발생한 프로그램 명칭을 기재한다.

소프트웨어 변경 보고서

		PAGE _____ OF _____
해 결 자	프로그램어	제출 일자 SCR 번호
	제목	컴퓨터 명칭
	해결안 서술	
	현모듈 버전 _____ 문서 명칭 _____ 관련된 SCR _____	결함원인 _____ 복잡도 _____
L I B R	라이브러리언	<input type="checkbox"/> GO <input type="checkbox"/> NO GO 세 엑스터 버전 _____      세 모듈 버전 _____      일자 _____ 동시에 운영될 SCR _____
	검증자	검증 <input type="checkbox"/> GO <input type="checkbox"/> NO GO    검증자 _____ 검증 버전 _____      검증 일자 _____      승인 _____
	비고란	

(그림 3) 소프트웨어 변경 보고서 양식  
(Fig. 3) Software change report form

- 해결안 서술 - 해결안을 상세하게 기술한다. 가능하면 라인 번호를 편집한 원시코드를 제시한다. 변경된 프로그램 모듈을 제시하지 못할 경우는 변경에 영향을 주는 모든 모듈을 리스트로 출력하여 첨부한다.
- 현모듈 버전 - 개별적인 모듈의 버전 번호를 기재한다.
- 문서 명칭 - 변경에 영향을 받는 문서 명칭을 기재한다. 영향이 없으면, None를 기재한다.
- 관련된 SCR - 문제에 관련된 모든 SCR 번호를 기재한다.
- 승인 - SCR를 승인하고 해결안을 검증하는 해결자 팀장의 서명 또는 날인을 한다.
- 결함 원인 - 결함의 원인을 3자의 문자로 기재한다(표 2 참조).
- 복잡도 - 해당 모듈의 복잡도를 기재한다.



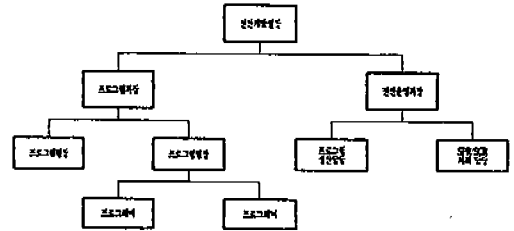
- ② 라이브러리(LIBR) 부분
  - GO - 컴파일/어셈블리에서 오류가 발생하지 않았을 경우에 체크한다.
  - NO GO - 컴파일/어셈블리에서 오류가 발생하였을 경우에 체크한다.
  - 새 마스터 버전 - 변경 작업이 완료되어 운영될 새로운 마스터 버전 번호를 기재한다.
  - 새 모듈 버전 - 개별 모듈들이 버전 번호를 가진다면 새로운 모듈 버전 번호를 기재한다.
  - 일자 - 변경되어 운영될 일자를 기재한다.
  - 동시에 운영될 SCR - 동시에 변경되어 운영될 모든 SCR의 번호를 기재한다.

- ③ 검증자 부분
  - GO, NO GO - 문제의 해결안이 정확하게 수정된 것을 검증한 테스트일 경우에 체크한다.
  - 검증자 - 해결안을 검증하기 위해 테스트를 수행한 요원의 성명을 기재한다.
  - 검증 버전 - 해결안을 검증하기 위해 사용된 시스템 버전 번호를 기재한다.
  - 검증 일자 - 검증을 수행한 일자를 기재한다.
  - 승인 - 검증을 승인하는 팀장의 서명 또는 날인을 한다.
  - 비고란 - 해결안 검증 방법을 서술하고, 만약 SCR이 NO GO이면, 그 이유를 서술한다.

(3) SPR/SCR 처리 절차

SPR/SCR은 개방, 수정, 해결, 완료의 4단계 과정으로 처리된다. “개방” 상태는 소프트웨어 문제가 발생하여 SPR이 제기된 것을 의미하며, “수정” 상태는 제기된 SPR를 해결하기 위해서 SCR을 제안한 것을 의미한다. “해결” 상태는 제안된 SCR의 내용을 토대로 새로운 시스템 버전을 생산한 것을 의미하며, “완료” 상태는 생산된 새로운 시스템 버전을 검증하여

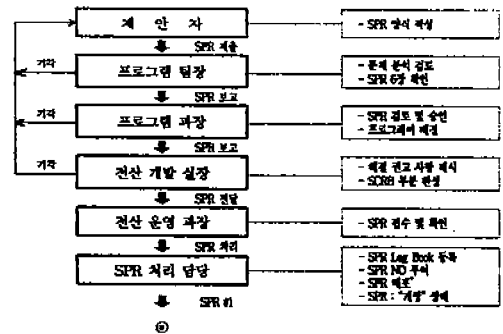
SPR에서 제기된 소프트웨어 결함이 해결된 것을 의미한다. 또한 SPR/SCR 처리는 소형 프로젝트보다 대형 프로젝트의 개발 조직에서 필요한 것이 때문에 그림 4와 같은 프로젝트 개발 조직을 제안할 수 있다.



(그림 4) 프로젝트 개발 조직  
(Fig. 4) Project development organization

① SPR/SCR 처리:개방 상태

SPR/SCR 처리의 개방 상태를 담당자별/업무 흐름도를 나타내면 그림 5와 같다.



(그림 5) SPR/SCR 처리 흐름도: 개방 상태  
(Fig. 5) Flow chart of the SPR/SCR processing : open status

② SPR/SCR 처리:수정 상태

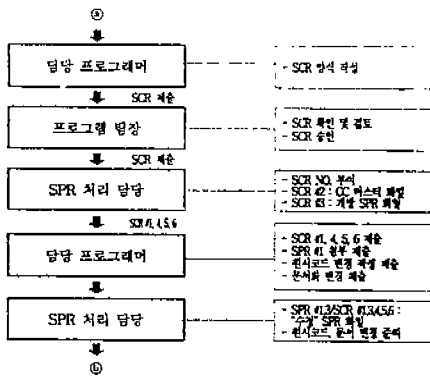
SPR/SCR 처리의 수정 상태를 나타내면 그림 6과 같다.

③ SPR/SCR 처리:해결 상태

SPR/SCR 처리의 해결 상태를 나타내면 그림 7과 같다.

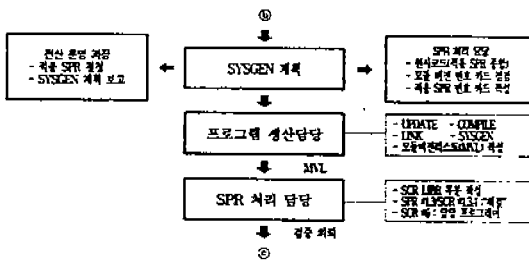
④ SPR/SCR 처리:완료 상태

SPR/SCR 처리의 완료 상태를 나타내면 그림 8과 같다.



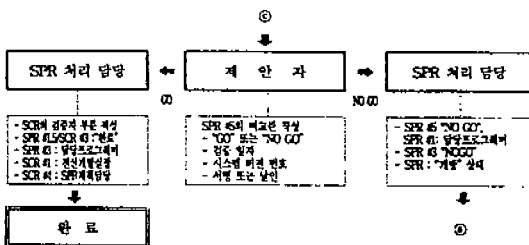
(그림 6) SPR/SCR 처리 흐름도: 수정 상태

(Fig. 6) Flow chart of the SPR/SCR processing : fixed status



(그림 7) SPR/SCR 처리 흐름도: 해결 상태

(Fig. 7) Flow chart of the SPR/SCR processing : resolved status



(그림 8) SPR/SCR 처리 흐름도: 완료 상태

(Fig. 8) Flow chart of the SPR/SCR processing : closed status

(4) SPR/SCR 관리 시스템의 기대 효과

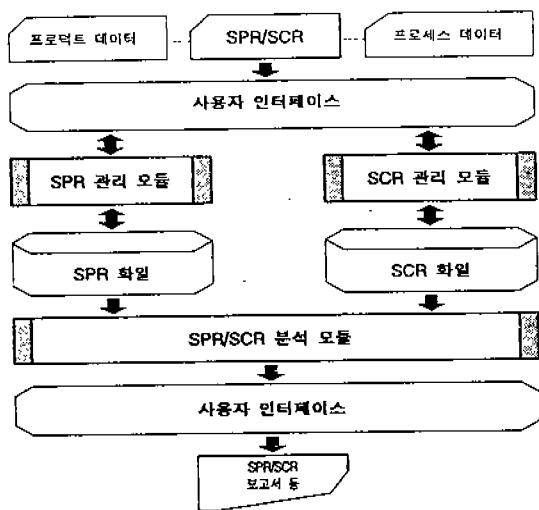
SPR/SCR 관리 시스템의 효율적인 면(기대 효과)을 들면 첫째로 개발자들을 일상적이고 반복적인 일로부터 해방시켜 주며, 정보에 대한 접근을 제어해 주며, 최근의 갱신된 보고서들을 제공해 준다. 둘째로 온라인 변경 이력을 유지하여 즉시 필요한 신뢰성 분석 정보를 제공하며, 현 시점에서의 개방, 수정, 해결, 완료된 SPR의 건수를 포함한 정기적인 상태 보고서를 제공해 준다. 셋째로 개방, 수정, 해결, 완료된 SPR의 누적 건수를 제공하며, 주별 및 기능 영역별로 개방, 수정, 해결, 완료된 SPR의 전체 건수를 통계적으로 제공해 준다. 넷째로 모든 프로젝트에 적용될 수 있도록 각종 입·출력 자료가 표준화되고, 적용된 결과의 개발자, 사용자, 관리자들이 이해하기 쉬운 시각적인 형태로 신뢰성 분석 정보를 제공하며, 프로젝트의 진행 상황을 용이하게 파악할 수 있다.

3.2 시스템 설계

3.1 절에서 제안한 SPR/SCR 관리 시스템은 대형 프로젝트 개발의 각 단계에서 발생하는 소프트웨어 결함 데이터, 프로세스 데이터, 프로덕트 데이터를 수용할 수 있는 신뢰성 데이터 수집 및 분석을 위한 필수적인 도구임을 알 수 있었다.

(1) SPR/SCR 관리 시스템의 구조

SPR/SCR 관리 시스템의 구조는 그림 9에 나타난 것처럼 사용자 인터페이스, SPR 관리 모듈, SCR 관리 모듈, SPR/SCR 분석 모듈로 구분된다. 사용자 인터페이스는 각 기능을 호출하여 SPR 및 SCR 자료를 입력할 수 있는 화면과 입력된 SPR/SCR 자료를 분석한 결과를 화면에 출력하는 기능을 제공한다. SPR 관리 모듈은 SPR 자료의 등록, 갱신 및 삭제 기능을 수행하여 SPR 화일을 생성하고 관리하며, SCR 관리 모듈은 SCR 자료의 등록, 갱신 및 삭제 기능을 수행하여 SCR 화일을 생성하고 관리한다. SPR/SCR 분석 모듈은 SPR 관리 모듈과 SCR 관리 모듈에 의해 생성된 SPR 화일과 SCR 화일의 내용을 토대로 SPR 상태 보고서, SPR 통계 보고서, 개인별 SPR 보고서 등의 정보를 출력하여 프로젝트 요원에게 제공하는 기능으로서 프로젝트 진행 상황을 쉽게 파악할 수 있도록 구성한다.



(그림 9) SPR/SCR 관리 시스템의 구조

(Fig. 9) Construction of the SPR/SCR management system

(2) 화일 설계

① SPR 화일 설계

SPR 화일의 구성은 그림 2의 SPR 양식의 내용을 토대로 표 8에 제시된 항목으로 설계하였다.

<표 8> SPR의 화일 구성

<Table 8> File layout of software problem report

순번	설명	필드명칭	형태	길이	비고
1	제안자	SprOrgNam	문자형	8	성명
2	프로젝트 명칭	SprProjct	문자형	3	프로젝트 고유 명칭
3	발견 일자	SprDetDat	문자형	6	년월일 표현
		SprDetWek	숫자형	3	주 단위 표현
4	SPR 번호	SprNumber	숫자형	4	SPR에 부여된 번호
5	제목	SprTitles	문자형	40	문제 제목 명칭 서술
6	컴퓨터 명칭	SprProgld	문자형	1	문제가 발생한 프로그램 명칭
7	발견 버전	SprDetVer	문자형	6	문제가 발견된 시스템 버전 명칭
8	처리 상태	SprStatus	문자형	1	O(개방), F(수정), R(해결), C(완료)
9	성능 효과	SprPerEff	문자형	2	CR(중대), MA(중요), MI(경미)...
10	발견 단계	SprDetPha	문자형	2	RQ(분석), DE(설계), CO(코딩)...
11	기능 영역	SprFunAre	문자형	2	AP(응용), OS(지원)...
12	조정자	SprCoindr	문자형	8	성명

13	수정 일자	SprFixDat	문자형	6	년월일 표현
		SprFixWek	숫자형	3	주 단위 표현
14	해결 일자	SprRevDat	문자형	6	년월일 표현
		SprRevWek	숫자형	3	주 단위 표현
15	해결 버전	SprRevVer	문자형	6	문제를 해결한 시스템 버전 명칭
16	검증 일자	SprVerDat	문자형	6	년월일 표현
		SprVerWek	숫자형	3	주 단위 표현
17	검증 버전	SprVerVer	문자형	6	문제를 검증한 시스템 버전 명칭
18	완료 일자	SprClsDat	문자형	6	년월일 표현
		SprClsWek	숫자형	3	주 단위 표현
19	비고	SprRemark	문자형	20	참고 사항 기재

② SCR 화일 설계

SCR 화일의 구성은 그림 3의 SCR 양식의 내용을 토대로 표 9에 제시된 항목으로 설계하였다.

<표 9> SCR의 화일 구성

<Table 9> File layout of software change report

순번	설명	필드명칭	형태	길이	비고
1	SPR 번호	SprNumber	숫자형	4	일련번호(예:1000)
2	SCR 번호	ScrNumber	숫자형	3	일련번호(예:100)
3	프로그래머	ScrProgrm	문자형	8	성명
4	제출 일자	ScrSugDat	문자형	6	년월일 표현
		ScrSugWek	숫자형	3	주 단위 표현
5	제목	ScTitles	문자형	40	문제 제목 명칭 서술
6	컴퓨터 명칭	ScrProgld	문자형	1	문제가 발생한 프로그램 명칭
7	처리 상태	ScrStatus	문자형	1	X(NO GO), G(GO), W(새 SCR)
8	현 모듈 버전	ScrCurVer	문자형	6	현 모듈 명칭과 버전 번호
9	문서 영향	ScrDocEff	문자형	3	N(없음), Y(있음), V(해결)
10	결함 원인	ScrErrCau	문자형	3	표 2 참조
11	복잡도	ScrComplex	문자형	3	해당 모듈의 복잡도를 기재
12	새 모듈 버전	ScrNewVer	문자형	6	새 모듈 명칭과 버전 번호
13	비고	ScrRemark	문자형	20	참고 사항 기재

(3) 주요 화면 설계

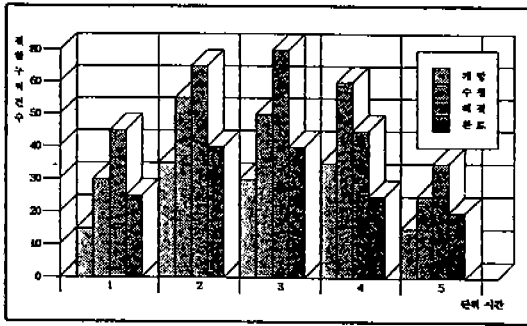
① 주메뉴 화면 설계

SPR/SCR 주메뉴 화면의 구성은 그림 10과 같이 설계하였다.



발자와 관리자는 개발 및 설계 작업의 향상을 위해 유용하게 활용할 수 있다.

사용 버전별 결함 발견 건수는 각 버전별 발견된 건수를 통해 설계 방법론을 개선하고, 단위 테스트를 위한 절차를 작성하며, 개발되는 소프트웨어를 위한 절차를 확립하는데 사용될 수 있다.



(그림 14) 처리 상태별 결함 누적 건수

(Fig. 14) Cumulative error counts for status versus unit times

결함 원인별 결함 발견 건수는 논리 오류, 데이터 취급 오류, 인터페이스 오류, 데이터 정의의 오류, 데이터 베이스 오류 등을 나타내는 것으로서 여기서는 논리 오류에 대한 결함 건수의 비율이 높게 나타나므로 코딩 과정을 개선하는 노력이 필요한 것을 알 수 있다.

발견 단계별 결함 발견 건수는 각 테스트 단계에서 발견되는 결함 건수를 통해 적용된 테스트 기법의 개선 방향을 결정할 수 있으며, 발견된 결함을 추적하는 지표가 될 수 있다.

심각성별 모듈 변경 수는 하나의 SPR에 대해 수정이 가해지는 SCR의 건수를 나타내는 것으로서 하나의 SPR에 대해 모듈 변경의 개수가 많을수록 설계 과정의 개선이 필수적으로 요구되는 것을 알 수 있다.

복잡도별 모듈 개수는 복잡도와 발견된 결함 건수 간의 관계가 있는지를 결정하는데 사용된다.

(5) 코드 설계

SPR/SCR 관리 시스템에서 사용되는 코드들은 표 10에 나타난 것처럼 SPR의 처리 상태, 성능 효과, 발견 단계, 기능 영역, SCR의 결함 원인, 문서화 영향, 처리 상태가 있다.

<표 10> SPR/SCR 관리 시스템의 코드 구성

<Table 10> Code layout of the SPR/SCR management system

구분	종류	길이	코드	의미	비고
SPR	처리 상태	1	O	SPR 개방 상태	Open
			F	SPR 수정 상태	Fixed
			R	SPR 해결 상태	Resolved
			C	SPR 완료 상태	Closed
	성능 효과	2	CR	중대함	Critical
			MA	중요함	Major
			MI	경미함	Minor
			NO	영향 없음	None
	발견 단계	2	UT	단위 테스트	Unit Testing
			IT	통합 테스트	Integration Testing
			VT	확인 테스트	Validation Testing
			ST	시스템 테스트	System Testing
			AT	인수 테스트	Acceptance Testing
			MN	유지 보수	Maintenance
	기능 영역	3	APS	응용 프로그램	Application Program Set
SPS			지원 프로그램	Supporting Program Set	
결함 원인	3	A00-G90	결함 원인 종류	표 2 참조	
문서 영향	1	N	영향 없음	No	
		Y	영향 있음	Yes	
		V	해결됨	Resolved	
처리 상태	1	X	검증 통과 오류	NO GO	
		G	검증 통과	GO	
		W	새로운 SCR	New SCR	

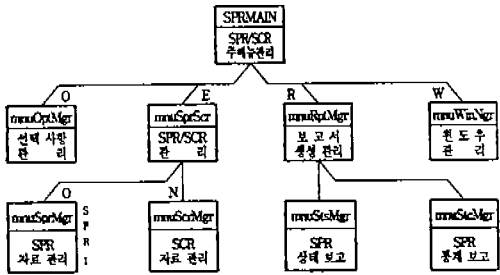
(6) 프로그램 설계 구조도

SPR/SCR 시스템은 선택 사항 프로그램, SPR/SCR 관리 프로그램, 보고서 생성 프로그램, 윈도우 관리 프로그램 등으로 구성되며, 개략적인 프로그램 설계 구조도는 그림 15와 같이 나타낼 수 있다.

선택 사항 관리 프로그램은 코드 관리, 유저 관리, 작업 종료등의 기능을 담당하는 것으로서 SPR/SCR 관리 시스템에서 사용되는 코드 부호 및 사용자에 대한 자료를 확인하고 점검하기 위해 관리되는 것이다.

윈도우 관리 프로그램은 계단형 정렬, 수평형 정렬, 수직형 정렬, 아이콘 정렬 등의 기능을 담당하는 것

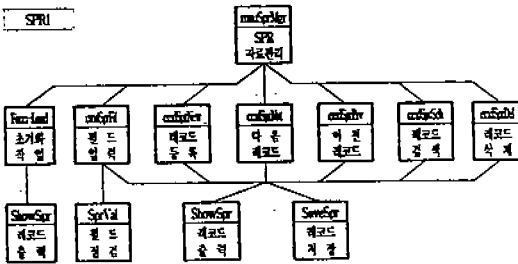
으로서 SPR/SCR 관리 시스템이 실행시 화면에 전시되는 하위 윈도우들을 정렬시키는데 사용된다.



(그림 15) SPR/SCR 관리 시스템의 설계 구조도  
(Fig. 15) Structured chart of the SPR/SCR management system

① SPR 자료 관리 프로그램 구조도

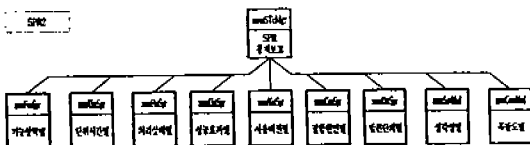
SPR 자료 관리 프로그램은 그림 16과 같이 초기화, 레코드 등록, 필드 입력, 다음 레코드, 이전 레코드, 레코드 검색, 레코드 삭제, 작업 종료 모듈 등으로 구성되어 있다.



(그림 16) SPR 자료 관리 프로그램의 설계 구조도  
(Fig. 16) Structured chart of the SPR data management program

② SPR 통계 보고 프로그램 구조도

SPR 통계 보고 프로그램은 그림 17과 같이 9가지의 통계 자료를 그래프로 제공하도록 구성되어 있다.



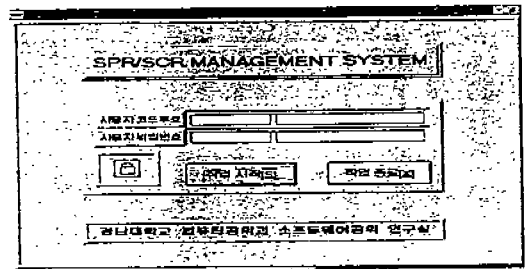
(그림 17) SPR 통계 보고의 설계 구조도  
(Fig. 17) Structured chart of the SPR statistics reporting program

3.3 시스템 구현

SPR/SCR 관리 시스템의 구현 환경은 다음과 같다. 개발에 사용된 하드웨어 환경은 IBM PC 호환 기종인 팬티엄 PC를 이용하였으며, 소프트웨어 환경의 운영 체제는 한글 Windows 3.1이며, 개발 언어는 Visual Basic 3.0을 이용하였다. 개발된 SPR/SCR 관리 시스템의 주요 특징은 다음과 같다. 첫째로 사용자가 쉽게 사용할 수 있도록 메뉴 중심(Menu Driven)인 동시에 명령 중심(Command Driven)으로 실행이 가능하도록 하였으며, 해당 메뉴를 선택하는 방법은 마우스로 클릭하거나 키보드의 방향키를 사용하도록 하였다. 둘째로 SPR과 SCR 양식에 있는 입력 자료의 대부분은 코드화되어 있기 때문에 자료를 입력시키는 소요 시간을 절약시키고, 화일에 대한 기억 장소의 크기를 최소화하였다. 셋째로 제공되는 대부분의 출력 정보는 개발자, 사용자 및 관리자들이 이해하기 쉬운 시각적인 형태로 제공되어 신뢰성 분석 정보 및 프로젝트의 개발 진행 상황을 쉽게 파악할 수 있다.

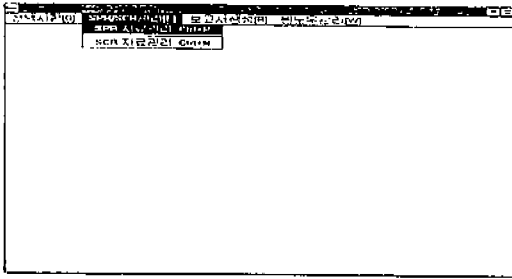
(1) 시스템 초기 화면

시스템 초기 화면은 그림 18과 같이 구성되어 있으며, 시스템을 운용할 수 있는 사용자의 코드 번호와 비밀 번호가 정확하게 일치해야만 사용이 가능하다.



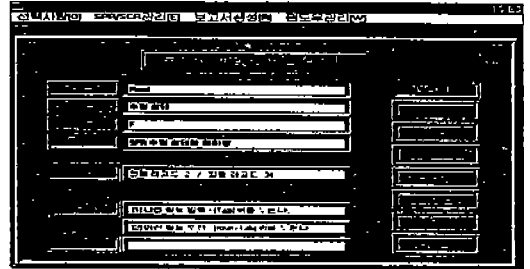
(그림 18) SPR/SCR 관리 시스템의 초기 화면  
(Fig. 18) Initial Display of SPR/SCR management system

주메뉴 관리 프로그램은 그림 19에 나타난 것과 같다. 주메뉴는 선택 사항, SPR/SCR 관리, 보고서 생성, 윈도우 관리 모듈로 구성되어 있다. 선택 사항 메뉴는 코드 관리, 사용자 관리 및 작업 종료로 구분되며, SPR/SCR 관리 메뉴는 SPR 자료 관리와 SCR 자료 관리로 구분된다.



(그림 19) SPR/SCR 관리 시스템의 주메뉴 화면  
(Fig. 19) Main menu of SPR/SCR management system

드 부호를 등록하는 기능과 필요시 해당 레코드를 검색 및 삭제하는 기능 등을 수행할 수 있다.



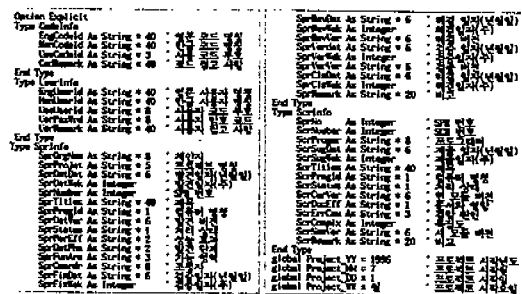
(그림 21) 코드 관리 화면 및 작업 선택 메뉴  
(Fig. 21) Code management layout and job selection menu

또한 보고서 생성 메뉴는 SPR 상태 보고와 통계 보고로 분류된다. SPR 상태 보고 메뉴에서는 개인별, 상태별, 영역별, 효과별 SPR 상태 보고서를 생성하고, SPR 통계 보고 메뉴에서는 SPR 통계 현황 보고, 단위 시간별 결함 발견 건수, 처리 상태별 결함 누적 건수, 성능 효과별 결함 누적 건수, 사용 버전별 결함 발견 건수, 결함 원인별 결함 발견 건수, 심각성별 모듈 변경수, 복잡도별 모듈 개수 등의 보고서를 생성한다. 윈도우 관리 메뉴는 윈도우의 모양을 계단형, 수평형, 수직형, 아이콘 정렬 등을 수행한다.

SPR/SCR 관리 시스템에서 사용되는 전역 모듈의 일반 선언부는 그림 20에 나타난 것과 같다.

코드 관리 프로그램은 11개의 주요 프로시저들로 구성되어 있으며, 프로시저 명칭과 기능은 표 11에 나타난 것과 같다.

<표 11> 코드 관리 프로시저의 명칭과 기능  
<Table 11> Procedure name and function of code management



(그림 20) 전역 모듈의 일반 선언부  
(Fig. 20) General declarations of global module

순번	프로시저 명칭	기능
1	Code.Frm	frmCode 양식의 일반 선언부
2	SaveCurrentCode()	Code.Dat 파일에 레코드를 저장하는 모듈
3	ShowCurrentCode()	Code.Dat 파일을 읽어 레코드를 화면에 출력하는 모듈
4	Form_Load()	Code.Dat 파일을 생성 및 개방하고 초기 화시키는 모듈
5	cmdCodNew_Click()	새로운 레코드를 Code. Dat 파일에 추가 하는 모듈
6	cmdCodFil_Click()	Code.Dat 파일의 필드를 점검하는 모듈
7	cmdCodNxt_Click()	다음 레코드를 화면에 출력하는 모듈
8	cmdCodPrv_Click()	이전 레코드를 화면에 출력하는 모듈
9	cmdCodSch_Click()	Code.Dat 파일에 해당 레코드를 검색하는 모듈
10	cmdCodDel_Click()	Code.Dat 파일에서 해당 레코드를 삭제하는 모듈
11	cmdCodExt_Click()	현재 레코드를 저장하고 프로시저를 종료 하는 모듈

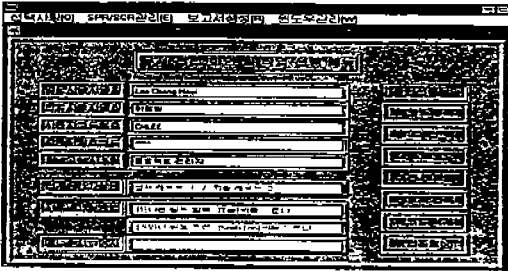
(2) 선택 사항 관리 프로그램

① 코드 관리 프로그램

코드 관리 프로그램은 그림 21에 나타난 것과 같이 입력 화면과 작업 선택 메뉴로 구성되었다. 입력 화면 필드는 영문 코드 명칭, 한글 코드 명칭, 사용 코드 부호, 참고 사항으로 구성되며, 작업 선택 메뉴는 코

② 유저 관리 프로그램

유저 관리 프로그램은 그림 22에 나타난 것과 같이 입력 화면과 작업 선택 메뉴로 구성되었다.



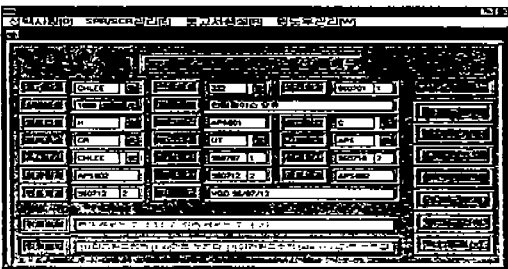
(그림 22) 유저 관리 화면 및 작업 선택 메뉴  
(Fig. 22) User management layout and job selection menu

입력 화면 필드는 영문 사용자 명칭, 한글 사용자 명칭 등으로 구성되며, 작업 선택 메뉴는 코드 부호를 등록하는 기능과 필요시 해당 레코드를 검색 및 삭제하는 기능 등을 수행할 수 있다. 유저 관리 프로그램은 코드 관리 프로그램과 유사하게 11개의 프로시저로 구성되어 있다.

(3) SPR/SCR 관리 프로그램

① SPR 관리 프로그램

SPR 관리 프로그램은 그림 23에 나타난 것과 같이 입력 화면과 작업 선택 메뉴로 구성되었다.



(그림 23) SPR 관리 화면 및 작업 선택 메뉴  
(Fig. 23) SPR management layout and job selection menu

입력 화면 필드는 SPR 번호, 제안자 및 발견 일자 등으로 구성되며, 작업 선택 메뉴는 SPR 자료를 등록

하는 기능과 필요시 해당 레코드를 검색 및 삭제하는 기능 등을 수행할 수 있다. SPR 관리 프로그램은 11개의 주요 프로시저들로 구성되어 있으며, 프로시저 명칭과 기능은 표 12에 나타난 것과 같다. 이들에 대한 주요 프로시저의 알고리즘은 다음과 같다.

㉠ cmdSprNxt\_Click()의 알고리즘

```

Sub cmdSprNxt_Click ( )
  If CurrentRecord = LastRecord Then
    Beep
    MsgBox "화일의 마지막 레코드입니다!", 48
  Else
    SaveCurrentSpr
    CurrentRecord = CurrentRecord + 1
    ShowCurrentSpr
  End If
End Sub
    
```

<표 12> SPR 관리의 프로시저 명칭 및 기능  
(Table 12) Procedure name and function of SPR management

순번	프로시저 명칭	기능
1	SPR_Frm	frmSPR 양식의 일반 선언부
2	SaveCurrentSPR()	SPR.Dat 화일에 레코드를 저장하는 모듈
3	ShowCurrentSPR()	SPR.Dat 화일을 읽어 레코드를 화면에 출력하는 모듈
4	Form_Load()	SPR.Dat 화일을 생성 및 개방하고 초기화시키는 모듈
5	cmdSPRNew_Click()	새로운 레코드를 SPR. Dat 화일에 추가하는 모듈
6	cmdSPRFil_Click()	SPR.Dat 화일의 필드를 점검하는 모듈
7	cmdSPRNxt_Click()	다음 레코드를 화면에 출력하는 모듈
8	cmdSPRPrv_Click()	이전 레코드를 화면에 출력하는 모듈
9	cmdSPRSch_Click()	SPR.Dat 화일에 해당 레코드를 검색하는 모듈
10	cmdSPRDel_Click()	SPR.Dat 화일에서 해당 레코드를 삭제하는 모듈
11	cmdSPRExt_Click()	현재 레코드를 저장하고 프로시저를 종료하는 모듈

㉡ cmdSprPrv\_Click()의 알고리즘

```

Sub cmdSprPrv_Click ( )
  If CurrentRecord = 1 Then
    Beep
    MsgBox "화일의 첫번째 레코드입니다!", 48
  Else
    ...
    ...
  End If
End Sub
    
```



```

SaveCurrentSpr
CurrentRecord = CurrentRecord - 1
ShowCurrentSpr
End If
End Sub
    
```

㉔ cmdCodSch\_Click()의 알고리즘

```

Sub cmdSprSch_Click ()
Dim SprNoToSearch As String
Dim Found As Integer
Dim RecNum As Long
Dim TempSpr As SprInfo
SprNoToSearch = InputBox("검색할 SPR 번호를 입력하세요.", "Search")
If SprNoToSearch = "" Then
txtSprNumber.SetFocus
Exit Sub
End If
SprNoToSearch = UCase(SprNoToSearch)
Found = False
For RecNum = 1 To LastRecord
Get #5, RecNum, TempSpr
If SprNoToSearch = UCase(Trim(TempSpr.SprNumber)) Then
Found = True
Exit For
End If
Next
If Found = True Then
SaveCurrentSpr
CurrentRecord = RecNum
ShowCurrentSpr
Else
MsgBox "SPR 번호 " + SprNoToSearch + " 는 화일에 없습니다"
End If
End Sub
    
```

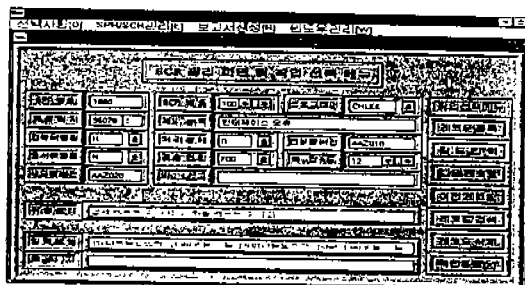
㉕ cmdSprDel\_Click()의 알고리즘

```

Sub cmdSprDel_Click ()
Dim DirResult
Dim TempSpr As SprInfo
Dim RecNum As Long
Dim TempRecNum As Long
If MsgBox("현재 레코드를 삭제하겠습니까?", 4) <> 6 Then
txtSprNumber.SetFocus
Exit Sub
End If
If Dir("Spr.Temp") = "Spr.Temp" Then
Kill ("Spr.Temp")
End If
Open "Spr.Temp" For Random As #6 Len = RecordLen
RecNum = 1
TempRecNum = 1
Do While RecNum < LastRecord + 1
If RecNum < CurrentRecord Then
Get #5, RecNum, TempSpr
Put #6, TempRecNum, TempSpr
TempRecNum = TempRecNum + 1
End If
RecNum = RecNum + 1
Loop
Close #5
Kill "Spr.Dat"
Close #6
Name "Spr.Temp" As "Spr.Dat"
Open "Spr.Dat" For Random As #5 Len = RecordLen
LastRecord = LastRecord - 1
If LastRecord = 0 Then
LastRecord = 1
End If
If CurrentRecord > LastRecord Then
CurrentRecord = LastRecord
End If
ShowCurrentSpr
End Sub
    
```

㉖ SCR 관리 프로그램

SCR 관리 프로그램은 그림 24에 나타난 것과 같이 입력 화면과 작업 선택 메뉴로 구성되었다.



(그림 24) SCR 관리 화면 및 작업 선택 메뉴  
(Fig. 24) SCR management layout and job selection menu

SCR 관리 프로그램은 입력 화면과 작업 선택 메뉴로 구성되었다. 입력 화면 필드는 SCR 번호, 프로그래머 및 제출 일자 등으로 구성되며, 작업 선택 메뉴는 SCR 자료를 등록하는 기능과 필요시 해당 레코드를 검색 및 삭제하는 기능 등을 수행할 수 있다. SCR 관리 프로그램은 SPR 관리 프로그램과 유사하게 11개의 프로시저들로 구성되어 있다.

(4) 보고서 생성 프로그램

보고서 생성 프로그램의 구성은 크게 SPR 상태 보고와 통계 보고로 분류된다. 이들 보고서에 대한 자세한 내용은 다음 장의 사례 연구에서 살펴보도록 한다. 보고서 생성 프로그램은 10개의 주요 프로그램들로 구성되어 있으며, 프로그램의 명칭과 기능은 표 13에 나타난 것과 같다.

〈표 13〉 보고서 생성 프로그램의 명칭 및 기능  
(Table 13) Name and function of report generation program

순번	프로그램 명칭	기능
1	STATREP.Frm	SPR 상태 보고서를 생성하는 프로그램
2	STATSPR.Frm	SPR 통계 현황 보고서를 생성하는 프로그램
3	ERRTIM.Frm	단위 시간별 결함 발견 건수를 생성하는 프로그램
4	ERRCNT.Frm	처리 상태별 결함 누적 건수를 생성하는 프로그램
5	ERRUNIT.Frm	성능 효과별 결함 누적 건수를 생성하는 프로그램
6	ERRVER.Frm	사용 버전별 결함 발견 건수를 생성하는 프로그램
7	ERRCAU.Frm	결함 원인별 결함 발견 건수를 생성하는 프로그램
8	ERRDET.Frm	발견 단계별 결함 발견 건수를 생성하는 프로그램
9	MODCNT.Frm	심각성별 모듈 변경 건수를 생성하는 프로그램
10	MODCOM.Frm	복잡도별 모듈수를 생성하는 프로그램

### 4. SPR/SCR 관리 시스템 적용

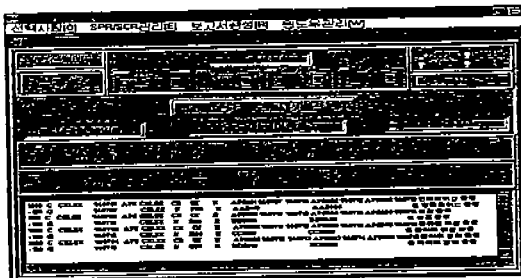
#### 4.1 사례 연구

사례 연구의 대상은 약 3년간 120여명의 개발 및 테스트 요원에 의해 개발된 실시간 지휘 통제 시스템을 선정하였다. 이 시스템에서 사용된 개발 언어는 JOVIAL이며, 전체 실행 코드는 대략 1,198,700개의 명령어로 구성되어 있다. 또한 이 소프트웨어는 크게 6개의 프로그램과 2,000여개의 모듈들로 구성되어 있다. 본 절에서는 실시간 지휘 통제 시스템의 테스트 단계에서 발견된 소프트웨어 결함 데이터를 SPR/SCR 관리 시스템에 적용하여 신뢰성 분석 정보를 중심으로 결과를 분석하고, 이들 보고서들이 실제 프로젝트 개발 진행상에 어떻게 적용되고, 어떠한 역할을 수행할 수 있는지를 살펴보기로 한다.

#### (1) SPR 상태 보고서

SPR 상태 보고서는 프로젝트 개발 중에 무슨 일이 발생하고, 누가 그것을 수행하고, 언제 그것이 발생했으며, 어떠한 것들이 영향을 받게 될 것인지 등의 의문점을 해결하는데 중요한 역할을 수행하는 보고서로서 그 세부 내용은 그림 25와 같다.

이 보고서는 개인별, 처리 상태별, 기능 영역별, 성능 효과별 등으로 분류되어 일정한 주기(주별, 월별 등)로 제공되어 위의 의문점을 해결해 주는 동시에 결함 수정 작업에 대한 우선 순위를 결정하는 지표가 될 수 있다.

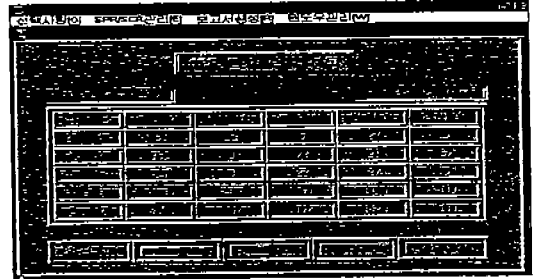


(그림 25) SPR 상태 보고서  
(Fig. 25) Status report of software problem report

#### (2) SPR 통계 현황 보고서

SPR 통계 현황 보고서는 기능 영역별(응용, 지원,

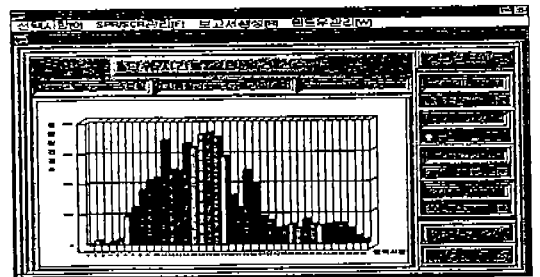
총괄)로 SPR 처리 상태(개방, 수정, 해결, 완료)와 성능 효과(중대함, 중요함, 경미함, 영향 없음)로 구분하여 발견된 SPR의 누적 건수를 통계적으로 제공함으로써 결함 수정 작업에 대한 진척 상황을 파악할 수 있는 것으로 그림 26과 같다. 즉 SPR 처리 상태에서 개방된 SPR 건수는 결함 수정 작업에 대한 투입 요원을 효율적으로 관리하고, 수정된 SPR 건수는 새로운 시스템 버전의 생산 시기를 결정하며, 해결된 SPR 건수는 생산된 시스템 버전의 적용 시기를 결정할 수 있다. 또한 SPR 성능 효과에서 중대한 결함 발견 건수는 결함 원인을 파악하여 해당 작업(예를 들면 설계, 코딩 등)의 개선 및 향상시킬 방안을 모색할 수 있다.



(그림 26) SPR 통계 보고서  
(Fig. 26) Statistics report of software problem report

#### (3) 단위 시간별 결함 발견 건수

단위 시간별 결함 발견 건수를 시각적인 그래프로 제공함으로써 결함 발견에 대한 전체적인 경향을 파악할 뿐만 아니라 결함 발견 비율을 결정하여 결함 발견 작업에 대한 필요한 자원(인력, 시간, 테스트 기법 등)을 관리하는데 사용될 수 있다. 또한 소프트웨

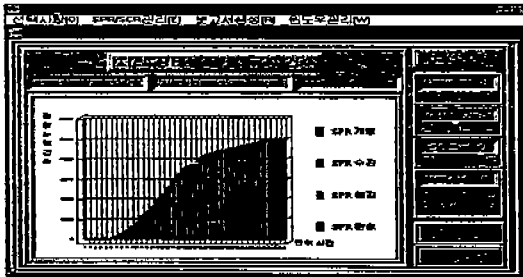


(그림 27) 단위 시간별 결함 발견 건수  
(Fig. 27) Detected error counts versus unit time(month)

어 신뢰성을 예측하고 측정하는 모델의 기본 자료로 제공될 수 있다. 그림 27은 테스트 기간 38개월 동안에 단위 시간별로 결함 발견 건수를 그래프로 나타낸 것이다. 즉 전체 테스트 기간동안 총 5,348건의 결함이 발견되었으며, 결함 발견에 대한 전체적인 경향을 파악할 수 있다.

(4) 처리 상태별 결함 누적 건수

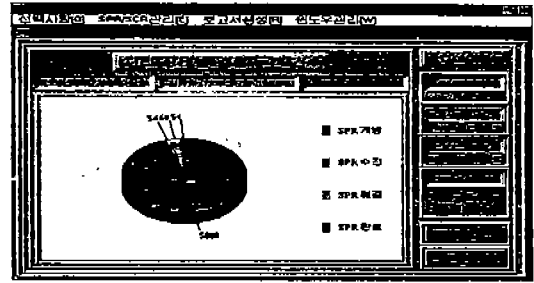
SPR 처리 상태별로 결함 누적 건수를 그림 28과 같이 시간적인 막대 그래프로 제공함으로써 결함 수정에 대한 경향을 분석할 뿐만 아니라 결함 수정 비율을 파악하여 결함 수정 작업에 대한 필요한 자원(인력, 시간, 방법 등)을 관리하는데 사용될 수 있다.



(그림 28) 처리 상태별 결함 누적 건수  
(Fig. 28) Cumulative error counts for status versus unit time

또한 SPR 처리 상태를 그림 29와 같이 현 시점에서 개방, 수정, 해결, 완료된 결함 건수를 통해 각각의 SPR 처리 상태의 비율을 분석할 수 있다. 즉 개방된 SPR 건수는 54건(1%), 수정된 SPR 건수는 160건(3%), 해결된 SPR 건수는 54건(1%), 완료된 SPR 건수는 5,080건(95%)으로 나타나고 있다. 여기서 완료된 SPR 건수의 비율이 높게 나타나고 있는 것은 결함 수정 작업에 대한 관리가 체계적이고 효율적으로 진행되었음을 알 수 있다.

그 외의 성능 효과별 결함 누적 건수 및 사용 버전별 결함 발견 건수 등에 대한 분석 내용은 앞에서 살펴본 처리 상태별 결함 누적 건수와 같이 가시적인 그래프로 제공되고 있다. 각각의 신뢰성 분석 정보에 대한 적용 및 역할은 다음 절에서 살펴본다.



(그림 29) 처리 상태별 결함 누적 건수  
(Fig. 29) Cumulative error counts versus status

4.2 요약 및 검토

SPR/SCR 관리 시스템을 선정된 실시간 지휘 통제 시스템에 적용한 신뢰성 분석 정보(10가지 보고서)에 대한 결과를 요약한 내용은 표 14와 같다.

<표 14> 신뢰성 분석 정보의 요약

<Table 14> Summary of reliability analysis information

순번	보고서 명칭	역할	기대 효과
1	SPR 상태 보고서	현 시점의 개인별, 기능별, 상태별 SPR 현황 제공	- 결함 수정 작업의 우선 순위 결정
2	SPR 통계 현황 보고서	SPR 처리 상태별/성능 효과별 SPR 누적 건수 제공	- 결함 수정 작업의 전체 현황 파악
3	처리 상태별 결함누적건수	개방, 수정, 해결, 완료된 SPR 누적 건수 제공	- 결함 발견 비율/수정 비율 분석 - 결함 발견/수정 작업의 자원 관리
4	단위 시간별 결함발견건수	주별/월별 결함 발견 건수 제공	- 결함 발견 작업의 자원 관리 - 신뢰도 예측 및 측정 자료 제공
5	성능 효과별 결함누적건수	성능 효과별 결함 누적 건수 제공	- 소프트웨어의 안정성 분석
6	사용 버전별 결함발견건수	각 사용 버전별 결함 발견 건수 제공	- 설계 방법론 개선 - 단위테스트의 체계적인 절차 수립
7	결함 원인별 결함발견건수	결함 원인(개선 오류, 논리 오류, 데이터 정의 오류, 데이터 취급 오류, 설계 오류, 인터페이스 오류, 기타 오류)별 결함 발견 건수 제공	- 결함 원인에 대한 경향 분석 - 각 결함 원인별 비율 분석 - 소프트웨어 개발 단계의 작업 과정 개선 결정
8	발견 단계별 결함발견건수	테스트 단계(단위 테스트, 통합 테스트, 확인 테스트, 시스템 테스트, 인수 테스트)별 결함 발견 건수 제공	- 테스트 단계별 결함 발견 건수 및 비율 분석 - 테스트 단계에 적용된 테스트 기법의 개선 방향 결정
9	심각성별 모듈 변경수	하나의 SPR에 대한 수정된 SCR 건수 제공	- 모듈 변경 건수 및 비율 분석 - 소프트웨어 개발 단계의 작업 과정 개선 결정

10	복잡도별 모듈 개수	모듈 복잡도별 개수 제공	-모듈 복잡도별 개수 및 비율 분석 -결합적 복잡도의 관련성 파악

### 5. 결론 및 향후 연구 과제

본 논문은 SDLC의 과정에서 발생하는 결함 자료를 수집 및 분석하기 위한 소프트웨어 문제 보고서와 소프트웨어 변경 보고서를 처리하는 SPR/SCR 시스템을 개발하여 적용하였다.

SPR/SCR 관리 시스템을 설계하기 위해 소프트웨어 개발 과정에서 수집될 수 있는 결함 매트릭스, 프로세스 매트릭스, 프로덕트 매트릭스에 대해 고찰하였다. 특히 결함 매트릭스에서 결함 출처 분류는 개발 프로젝트의 결함 경향을 발견하고 수정하기 위해서 필요한 것이며, 코딩 단계의 결함 발견 및 수정은 설계 단계에서의 결함 발견 및 수정에 비해 더 많은 비용이 투자되기 때문에 결함 출처 단계의 분류에 많은 관심을 가지게 된다. 또한 결함 원인 분류는 효과적인 테스트 계획을 수행하는 테스트 수행자를 도울 수 있다. 테스트 계획의 주요 목적은 결함을 발견하는 비율을 극대화하기 위한 것이다.

본 논문에서 설계되고 구현된 SPR/SCR 시스템은 서론에서 제시한 4가지 목적을 달성할 수 있다. 또한, 모든 프로젝트에 적용될 수 있도록 각종 입·출력 양식을 표준화하고, 적용된 결과는 개발자, 사용자 및 관리자들이 이해하기 쉬운 시각적인 형태로 신뢰성 분석 정보를 제공하며, 프로젝트의 개발 진행 상황을 용이하게 파악할 수 있다. 특히 10가지 보고서의 출력 정보를 제공함으로써 신뢰도 달성 비율, 테스트 종료 시기, 릴리즈 시기 등을 예측할 수 있고, 효율적인 개발 관리에 반영될 수 있다.

향후 연구 과제로는 구현된 SPR/SCR 시스템을 프로젝트 개발 현장에 직접 적용하여 개발 과정에서 수집되는 결함 매트릭스의 분석 뿐만 아니라 상세한 프로세스 및 프로덕트 매트릭스를 고려하여 신뢰성 분석 정보를 얻을 수 있는 자동화 도구를 개발하는데 있다.

### 참 고 문 헌

[1] 池田克夫, "컴퓨터시스템의 신뢰성", 電氣

學會雜誌, 100卷 8號, pp. 690-691, 昭和 55-8.  
 [2] 菊野, 松本, 鳥居, "ソフトウェア信頼性の實現技術", 電子情報通信學會論文誌, Vol. 73, No. 5, pp. 454-460, 1990.  
 [3] J. D. Musa, A. Iannino, and K. Okumoto, 'Software Reliability: Measurement, Prediction, Application', McGraw-Hill, 1987.  
 [4] A. M. Neufelder, 'Ensuring Software Reliability', Marcel Dekker, Inc., 1993.  
 [5] IEEE Standard Glossary of Software Engineering Terminology, ANSI-IEEE Standard 729, IEEE Publications, New York, 1983.  
 [6] R. L. Glass, 'Software Reliability Guidebook', Prentice-hall, Inc., 1979.  
 [7] J. Bowen, "Software Reliability", Hughes Aircraft Company, SEPN No. 3-3.3, 1981.  
 [8] TRW-Redondo Beach, "Software Reliability Study, Error Data Categorization", RADC-TR-76-238, 1976.  
 [9] M. Lipow, "Prediction of software failures", J. Systems and Software, Vol. 1, No. 1, pp. 71-75, 1979.  
 [10] R. W. Motley and W. D. Brooks, "Statistical Prediction of Programming Errors", RADC, 1977.  
 [11] M. Lipow, "Prediction of Software Failures", J. System and Software, Vol. 1, No. 1, pp. 71-75, 1979.  
 [12] D. Pointer, J. L. Albin, R. Ferreol, A. Bilodeau, "Experimenting with Computer Software Complexity and Reliability", Proc. 12th. int. Symp. Fault Tolerant Computing, pp. 381-387, 1982.  
 [13] R. Grady and D. Caswell, "Software Metrics: Establishing a Company Wide Program", Hewlett-Packard Company, Prentice-Hall, Englewood Cliffs, NJ, 1987.  
 [14] T. J. McCabe, "A Software Complexity Measure", IEEE Trans. Software Engineering, Vol. 2, No. 6, pp. 308-320, 1976.  
 [15] T. J. McCabe and C. W. Butler, "Design Complexity Measurement and Testing", CACM, Vol.

32, No. 12, pp. 1415-1425, 1989.

[16] G. Mc. Intire, "Program Trouble Report Procedure", Hughes Aircraft Company, SEP No. 3-5. 3, 1981.

[17] M. L. Shooman, 'Software Engineering: Design, Reliability and Management', McGraw-Hill, 1983.

[18] Trivedi, M. L. Shooman, "Error data collection in software systems", "Computer software reliability, Many state Markov techniques", RADC-TR-169, 1975.

[19] R. S. Pressman, 'Software Engineering: A Practitioner's Approach', 3th Ed., McGraw-Hill, 1995.

[20] G. J. Myers, 'The art of software testing', John wileys & Sons, Inc., 1979.



**이 참 희**

1989년 동국대학교 전자계산학과 졸업(공학사)  
 1992년 경남대학교 산업대학원 전자계산학과 졸업(공학석사)  
 1994년 경남대학교 대학원 컴퓨터공학과(박사과정)

1981년~1993년 공군전산장교 근무  
 1993년~현재 진주간호보건전문대학 사무자동화과 조교수  
 관심분야: 소프트웨어 공학(소프트웨어 테스트, 소프트웨어 신뢰도 평가)



**한 판 암**

1969년 동국대학교 졸업  
 1975년 동국대학교 경영대학원 졸업(경영학석사)  
 1989년 명지대학교 대학원 졸업(공학석사)  
 1992년 인천대학교 대학원 졸업(경영학박사)

1980년~현재 경남대학교 공과대학 컴퓨터공학과 교수  
 관심분야: 소프트웨어 품질 관리 및 신뢰성, 소프트웨어 개발 환경, 정보 공학