

CMMI 기반의 결함 분석 및 통제 시스템 개발[☆]

Development of a defect analysis and control system based on CMMI

조 성 민* 한 혁 수**
Sung-Min, Cho Hyuk-Soo, Han

요 약

결함을 초기 단계에 발견하여 제거함으로써 품질이 좋은 소프트웨어를 만들 수 있다. 이를 위해 결함을 효과적으로 추적하고 관리하는 결함 추적 시스템(Defect Tracking System)의 도입이 필요하다. 기존의 결함 추적 시스템은 한꺼번에 많은 종류의 데이터를 모으으로써 이를 도입한 조직에서 수많은 데이터를 왜 모아야 하는지에 대해 인식이 부족하고 결함 관리 프로세스 없이 도구만 적용함으로써 결함 추적 시스템 도입에 어려움을 겪고 있다. 또한 CMMI 성숙도 수준 2, 3에서는 결함 관리 활동을 수행하는 프로세스 영역이 없기 때문에 CMMI 성숙도 수준 2, 3에 해당하는 조직이 결함 추적 시스템을 도입하는데 문제가 되고 있다. 이에 본 논문에서는 CMMI 기반의 조직이 기존의 결함 추적 시스템을 도입하였을 때 생기는 문제점을 해결하고 CMMI 각 성숙도마다 어떤 결함 데이터를 모아야 하는지, 어떻게 결함 관리를 해야 하는지에 대한 가이드라인을 제공하고자 한다. 그리고 검토 및 테스트를 통해 도출된 결함 데이터에 대하여 각 담당자 별로 업무를 할당, 추적할 수 있게 하며, 결함에 관한 각종 상태정보와 통계 정보 등을 제공하는 결함 분석 및 통제 시스템을 개발하고자 한다.

Abstract

As we detect defects and eliminate them in early stages, we can make better quality software. For doing this task, we need to use a defect tracking system which can effectively track and manage defects that give severe effects on software quality. Those existing defect tracking systems have some weaknesses as we apply them to organizations that use CMMI for process improvements. Major problems of those systems are that they require the organizations to collect many types of defect data at a time without providing the proper explanation and even without the support of defect management process. The organizations at CMMI maturity level 2 and 3 have problems for analyzing those defects because there is no specific process area at CMMI maturity level 2 and 3 which directly handles defect managing activities. This paper resolves those problems by developing a defect tracking system which offers methods of managing defects. And the system provides guidelines of which defects should be gathered for each CMMI maturity levels. The system also has functions to generate various status and statistic information on defects, and to assign defect data to the person in charge so that he or she track the defect to the closure

☞ Key Words : CMMI(Capability Maturity Model Integration), 성숙도 수준, 결함(Defect), 결함 추적 시스템(Defect Tracking System)

1. 서 론

개발된 소프트웨어 제품이 고객이 원하는 대로 작동하지 않는다면, 그 제품에 오류/장애/실패(통

칭하여 결함(Defects))가 많이 내포되어 있다는 의미가 된다. 따라서 이러한 결함을 발견하여 제거해야만 품질이 높은 소프트웨어를 만들 수 있다. 이를 위해 품질에 많은 영향을 끼치는 결함을 효과적으로 추적하고 관리할 수 있는 결함 추적 시스템(Defect Tracking System)의 도입이 필요하다. 기존의 결함 추적 시스템들은 많은 데이터를 한꺼번에 모으도록 요구하여, 해당 데이터를 왜 모아야 하는지에 대한 인식이 부족한 상태에서 데이터를 모으도록 유도함으로써 데이터를 제공하

* 정 회 원 : 상명대학교 일반대학원 컴퓨터과학과(석사)
upoi@smu.ac.kr

** 정 회 원 : 상명대학교 소프트웨어학부 교수
hshan@smu.ac.kr(교신저자)

[2006/12/14 투고 - 2006/12/19 심사 - 2007/01/04 완료]

☆ 본 연구는 상명대학교 소프트웨어·미디어 연구소의 지원으로 수행되었음

는 사람들이 신뢰성 있는 데이터의 제공을 힘들게 하는 문제점을 야기한다. 그리고 결함 관리 프로세스를 제공하지 않고 데이터 수집 기능만을 제공하기 때문에 CMMI 기반의 프로세스 개선을 수행하고 있는 조직들이 결함 추적 시스템을 도입하는데 문제가 되고 있다. 또한 CMMI는 성숙도 수준 2와 3에 직접적으로 결함 관리 활동을 수행하는 프로세스 영역을 가지고 있지 않다[12]. 하지만, 성숙도 수준 2에서부터 결함 데이터들은 테스트와 디버깅에서 중요한 역할을 하고 있다 [13]. 본 논문에서는 CMMI 기반으로 프로세스 개선 활동을 하는 조직들이 성숙도 수준 2에서부터 결함 데이터에 대한 관리를 시작하여, 성숙도가 높아질수록 데이터에 대한 많은 정보를 보유하도록 하기 위해 기존의 결함 추적 시스템을 도입하였을 때 생기는 문제점을 해결한 개선된 결함 분석 및 통제 시스템을 개발하였다. 이 도구를 통해 CMMI 각 성숙도마다 어떤 결함 데이터를 모아야 하는지, 어떻게 결함관리를 해야 하는지에 대한 지침을 제공하여 결함 방지에 기여하고자 한다.

2. CMMI

CMMI(Capability Maturity Model Integration)는 과거 소프트웨어 프로세스 개선 모델로써 사용되던 SW-CMM의 진화된 모델로서 SW-CMM V2.0 Draft C, EIA/IS 731, IPD-CMM V0.98을 기반으로 개발되었다. CMMI는 과거 SW-CMM이 소프트웨어와 관련된 프로세스 개선 활동만을 지원하던 취약점을 보완하여 소프트웨어, 시스템, 하드웨어, 서비스와 관련된 모든 프로세스 개선 활동을 지원한다[18]. CMMI의 성숙도 표현 방식은 CMM과 마찬가지로 5개 성숙도 수준으로 나누어지며 각 성숙도 레벨에는 해당 레벨에서 수행해야 하는 22개의 프로세스 영역(Process Area)을 제시하고 있다[14].

3. 결함

3.1 결함이란?

결함은 품질 관련 모델 및 전문가들에 의해 다양하게 정의 내려지고 있다. 결함은 오류/장애/실패를 포함하며 고객이 요구하고 의도한 것과 산출물을 다르게 만듦으로써 재작업을 요구하는 모든 원인으로 정의 된다[3]. 결함은 소프트웨어의 품질과 직접 관련이 있으므로 여러 의미에서 결함 데이터는 공수 데이터보다 더 중요하다[9]. 결함 데이터는 우선 프로젝트 관리를 위해 필요하다. 대규모 프로젝트는 수 천 개의 결함을 포함할 수 있는데, 이 결함은 각기 다른 사람들이 프로젝트의 각각 다른 단계에서 발견하게 된다. 흔히 프로세스에서 결함을 고치는 사람과 결함을 발견하거나 보고하는 사람은 서로 다르다[11].

3.2 결함 추적 시스템

결함 추적 시스템(Defect Tracking System)은 하나의 제품을 개발하는 과정에서 발생하는 프로그램의 결함을 관리할 수 있도록 도구 형태로 제공되는 것으로 결함 보고자와 결함 수정자와의 의사소통을 원활히 할 수 있도록 도와준다. 이러한 도구는 데이터베이스(Database)를 근간에 두고 결함을 데이터베이스에 레코드(Record) 형태로 입력(생성)/수정/삭제하면서 결함 상태에 따라 결함을 추적하는 시스템이다[3]. 여기서 결함(Defect)은 버그(Bug), 오류(Error), 해결 대상(Issue) 등과 혼용하여 사용하고 있다. 같은 맥락에서 결함 추적 시스템은 버그 추적 시스템(Bug Tracking System)이라고도 한다. 버그 추적 시스템은 Mantis, Bugzilla, Tracker, Debian Bug Tracking System, Open Track, Time Sheets for Networks(TSN), Journyx Time 등의 수 많은 공개/free 소프트웨어와 TestTrack Pro, Rubicon Tracker, Keystone 등의 여러 상용 소프트웨어가 있다[6][7][8][9]. 결함 추적 시스템의 주요 기능에는 결함의 키워드 검색과 테스터의 시스템 사용 권한 관리 등이 있다[2].

3.3 기존 결함 추적 시스템의 문제점

결함 데이터를 처음 모으거나 결함 및 측정에 대한 이해가 약한 조직은 기존 결함 추적 시스템에서 요구하는 데이터 들을 한꺼번에 모을 수 없다. 시스템 도입에 따라 많은 데이터를 한꺼번에 모음으로써, 해당 데이터를 왜 모아야 하는지에 대한 인식이 부족하고 결함 관리 프로세스 없이 시스템만 도입함으로써 결함 관련 활동을 조직에 정착시키지 못했다. 또한 결함 추적 시스템을 통해 나온 결함 데이터를 어떻게 분석해야 이후 프로젝트에 도움이 될 수 있을지에 대한 가이드라인도 없다. 이에 본 논문은 이러한 문제점을 해결하고자 한다.

4. CMMI 기반의 결함 관리

CMMI에서는 결함에 관한 구체적인 내용을 언급하는 프로세스 영역이 성숙도 수준 5의 CAR(Causal Analysis and Resolution)라는 프로세스 영역에 존재한다[12]. CAR 프로세스 영역은 결함 예방과 관련된 활동이다. 이전 성숙도에 해당하는 조직이 성숙도 수준 5에 도달할 때까지 결함 제거 및 예방 활동을 수행하지 않는 것은 아니다[13]. 성숙도 수준 3까지는 결함 관련 활동이 결함 제거 및 관리에 초점이 맞춰져 있다. 또한 결함 제거 및 관리가 여러 프로세스 영역을 통해 복합적으로 이루어진다. 본 논문에서 제안하고 있는 범위는 성숙도 수준 3까지의 범위로 한정한다. 이를 위해 각 성숙도 수준 별로 결함 관리상에 어떠한 차이점이 있는지를 파악해야 한다.

4.1 성숙도 수준 2의 결함 관리

성숙도 수준 2인 조직은 기본적인 프로젝트 관리 프로세스들을 갖고 있으며, 또한 그것에 따라서 업무가 수행된다[12]. 성숙도 수준 2의 조직이 수행하는 결함 관리의 가장 큰 활동인 ‘발견’과

‘제거’는 CMMI의 검증 및 확인(Verification and Validation) 프로세스 영역에서 수행한다. 하지만 검증 및 확인 프로세스 영역은 성숙도 수준 2에는 존재하지 않고 성숙도 수준 3에 존재한다. 그렇기 때문에 검증 및 확인 영역의 주요 활동인 동료 검토(Peer Review)를 수행하는 조직도 있지만, 성숙도 수준 3의 목적(Goal)을 만족시킬 만한 성숙도를 갖고 있지 못하며, 결함 발견 및 제거 활동은 주로 테스트를 통해 이루어지게 된다[13]. 결함을 관리하는 수준 또한 프로젝트 수준에서만 머무르게 된다. 따라서 성숙도 수준 2의 결함 관리 목표는 조기에 결함을 발견하여 제거하고, 지금까지 발견된 결함에서 배우며, 프로젝트의 남은 기간 중에 결함을 예방하려 하는 것이다[10]. 그리고 발견된 결함을 관리하는 활동은 다른 프로세스 영역에서 품질 활동의 일부로 진행하게 된다.

4.2 성숙도 수준 3의 결함 관리

성숙도 수준 3인 조직의 특징은 해당 조직에서 따라야 하는 조직 차원의 표준 프로세스를 보유하고 있는 것이다. 이러한 조직 표준 프로세스는 해당 조직에서 프로젝트를 수행할 경우 프로젝트 특성에 따라 적절하게 조정하여 사용된다[12].

이러한 성숙도 수준은 훨씬 구체적이고 정교한 프로세스를 제공하기 때문에 성숙도 수준 2와 차별화 된다[13]. 조직은 이러한 구체적인 프로세스를 이용해 측정치를 수집할 수 있으며, 또한 프로세스들 간의 관계를 정확하게 이해할 수 있다. 이러한 특성에 비추어 성숙도 수준 3에 해당하는 검증 및 확인 프로세스 영역을 수행하게 된다. 결함 발견 및 제거 활동은 공식적인 검토와 테스트를 통해 이루어진다. 결함을 관리하는 수준은 조직 차원에서 이루어지며, 결함 관리 활동을 하며 문제점이 드러난 프로세스에 대해 개선 활동이 일어난다. 조직 차원의 결함 분석은 조직 전체를 위한 체크리스트, 프로세스, 또는 교육 훈련 등의 확장으로 연결될 수 있다[13].

4.3 현재의 문제점

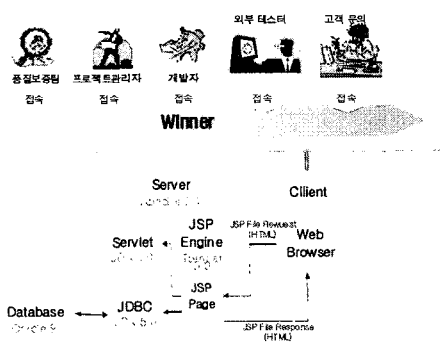
본 장에서 살펴본 것처럼 CMMI 성숙도 수준 2 또는 3에 해당하는 조직은 따로 결함을 관리할 수 있는 특정 프로세스 영역이 존재하지 않으며, 여러 프로세스 영역에 걸쳐 결함 관리가 복합적으로 일어나고 있다. 또한 CMMI 기반의 조직들이 기존 결함 추적 시스템을 도입하여 결함 관리를 할 경우, 각 성숙도 수준마다 어떤 결함 데이터를 모아야 하는지, 어떻게 결함 관리를 해야 하는지에 대한 가이드라인이 없어 심사나 프로세스 개선 활동 시 많은 혼선이 가중되고 있다.

5. CMMI 기반의 결함 분석 및 통제 시스템 개발

CMMI 기반의 조직에서 만든 제품의 개발 중이나 인수 후의 결함 데이터를 수집, 추적, 분석할 수 있도록 하는 도구로 이름은 품질에서 승리한다는 의미로 Winner라 칭하였다.

5.1 Winner 개발 환경 및 시스템 구성

본 시스템의 개발 환경 및 시스템 구성은 아래 그림 1과 같다.



(그림 1) Winner 개발 환경 및 시스템 구성도

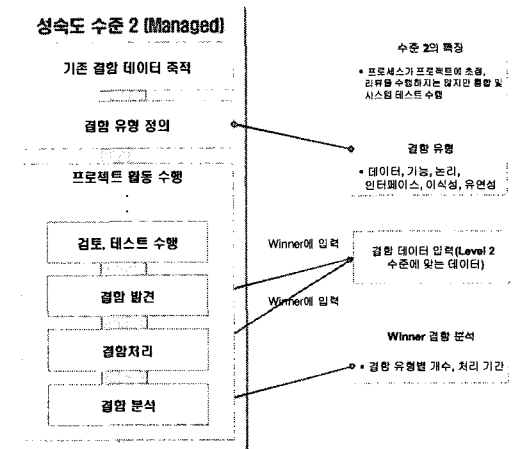
5.2 Winner의 주요 기능

Winner는 기존의 결함 추적 시스템의 기능 면에서는 유사하나, 운영 측면에서 CMMI 성숙도 수준별로 기능을 설계하였다.

1. CMMI 성숙도 수준별 결함 데이터 관리
2. 조직에 맞는 새로운 결함 유형 정의 가능
3. 결함의 상태, 현재 처리 담당자(역할) 할당
4. 결함 유형 분석, 결함의 유입 단계 분석
5. 결함의 발견 단계 정보 관리
6. 단계별, Impact 별 결함 개수 및 공수에 대한 통계 그래프 제공
7. 사용자 역할에 따른 권한 관리 기능
8. 사용자에게 메일 전송

5.3 Winner를 통한 성숙도 수준 2의 결함관리 활동

본 절에서는 4.1에서 언급했던 성숙도 수준 2에 해당하는 결함 관리 활동의 특징에 맞추어 Winner의 결함 관리 활동과 결함 데이터를 제시하였다. CMMI 성숙도 수준 2에 해당하는 조직이 Winner를 활용하여 결함 관리 활동 시 그림 2의 순서에 따라 작업하게 된다.



(그림 2) 성숙도 수준 2의 결함 관리 활동

Winner 에 입력되는 성숙도 수준 2의 결함 관련 데이터는 표 1에 정리된 내용과 같다.

(표 1) 성숙도 수준 2의 결함 데이터

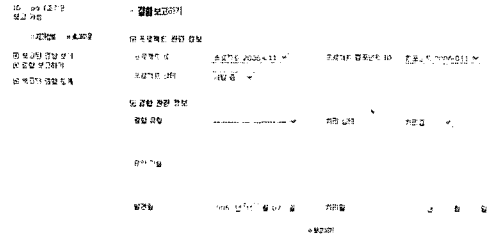
번호	데이터	데이터 설명	입력/분석
1	Product ID	결함이 발견된 프로젝트 또는 서비스	입력
2	Product Component ID	결함이 발견된 프로젝트 밑에 하위 레벨	입력
3	프로젝트 상태	프로젝트가 개발 중/인도 후 인지를 표시	입력
4	결함 유형	결함 분류	입력
5	요약 기술	결함이 어떠한 과정을 거쳐 일어났는지 자세한 설명	입력
6	처리 상태	결함 처리가 진행되는 상태	입력
7	발견일	결함을 발견한 날짜	입력
8	처리일	결함을 처리한 날짜	입력
9	담당자	결함을 처리하는 담당자	입력
10	인도 후 결함수	인도 후 발견된 전체 결함 개수	분석
11	결함 유형별 개수	결함 DB에 입력된 유형별 결함 수	분석
12	결함처리기간	발견일부터 처리일까지 공수	분석

입력 데이터는 결함 보고자 및 담당자가 Winner에 입력하는 데이터를 말하며, 분석 데이터는 프로젝트 관리자 및 경영진이 볼 수 있는 데이터를 의미한다. “결함 유형” 데이터의 경우에는 PSP(Personal Software Process)에서 제공하는 결함 유형[16]이나 IEEE Std 1044 (Classification for software anomalies)에서 제공하는 결함 유형[15] 중 조직에 적합한 유형으로 선택할 수 있게 했으며, 기존 결함 유형에 새로운 유형을 정의할 수도 있고, 기존 모델의 결함 유형을 사용하고 싶지 않은 경우 새로운 결함 유형을 만들 수도 있다.

5.3.1 Winner 성숙도 수준 2의 사용자 입력 화면

표 1에 제시된 성숙도 수준 2의 결함 데이터에

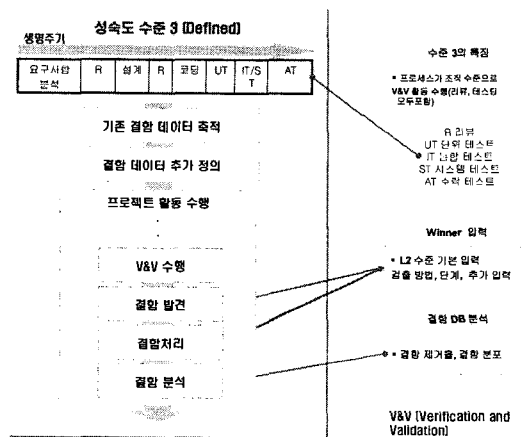
맞추어 결함 데이터를 입력 받는다.



(그림 3) 성숙도 수준 2의 사용자 입력 화면

5.4 Winner를 통한 성숙도 수준 3의 결함관리 활동

본 절에서는 4.1에서 언급했던 성숙도 수준 3에 해당하는 결함 관리 활동의 특징에 맞추어 Winner의 결함 관리 활동과 결함 데이터를 제시하였다. CMMI 성숙도 수준 3에 해당하는 조직이 Winner를 활용하여 결함 관리 활동 시 그림 4의 순서에 따라 작업하게 된다.



(그림 4) 성숙도 수준 3의 결함 관리 활동

Winner를 통해 입력/분석 되는 성숙도 수준 3의 결함 관련 데이터는 표 2에 정리하였다.

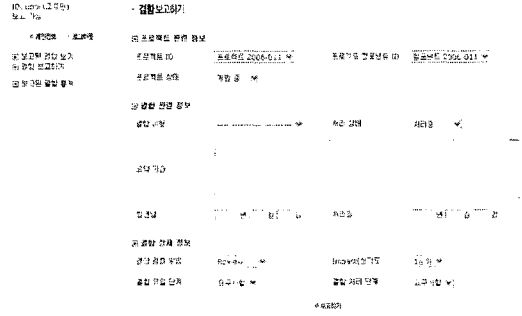
(표 2) 성숙도 수준 3의 결함 데이터

번호	데이터	데이터 설명	입력 / 분석
1	결함 검출 방법	Review or Inspection or Test	입력
2	결함 유입 단계	요구사항/설계/코딩 /테스팅	입력
3	결함 제거 단계	요구사항/설계/코딩 /테스팅	입력
4	Impact	결함의 영향력 (4단계)	입력
5	결함 주입률	전체 결함수/ 크기(FP, LOC)	분석
6	Impact별 결함수	Impact에 따라 결함 수를 분석	분석
7	검출 방법 별 결함수	검출방법에 따라 결함 수 분석	분석
8	결함 분산	단계별에 결함 수	분석

표 2에서 제시한 데이터는 표 1에서 제시한 성숙도 수준 2의 데이터 이외에 앞서 4.1에서 언급했던 성숙도 수준 3의 특징을 반영하여 이전 수준 보다 추가된 데이터이다. 또한 성숙도 수준 3에서는 조직에 표준 프로세스가 정립되기 때문에 개발 단계 중 어느 부분에서 결함이 유입되고 제거되는지를 파악할 수 있으며, 검증 및 확인 활동을 수행하기 때문에 매 단계마다 공식적인 검토 활동을 통해서도 결함을 발견 및 제거할 수 있다. 이러한 데이터를 근거로 조직 프로세스 개선 활동에 도움을 줄 수 있다. 이로 인해 조직에서 표 2에서 제시한 1, 2, 3, 5, 7, 8 번과 같은 데이터를 모을 수 있다. 그리고 성숙도 수준 3에서는 성숙도 수준 2에서 모았던 결함 유형 및 처리 기간, 인도 후 결함 수 등을 근거로 어떤 유형의 결함이 프로젝트에 많은 영향을 주었는지 파악이 가능하다. 이를 통해 프로젝트 관리자는 이러한 유형의 결함을 미리 파악함으로써 결함을 담당자에게 할당할 때 프로젝트에 미치는 영향력(Impact)을 우선순위로 두어 할당 할 수 있다. 개발자는 영향력이 높은 결함이 할당될수록 우선 순위를 두고 처리해야 한다.

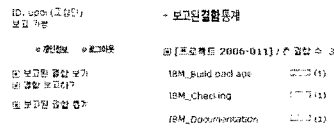
5.4.1 Winner 성숙도 수준 3의 사용자 입력 화면

표 2에 제시된 성숙도 수준 3의 결함 데이터에 맞추어 결함 데이터를 입력 받는다.



(그림 5) 성숙도 수준 3의 사용자 입력 화면

5.5 Winner 주요 분석 정보



(그림 6) 결함 유형 분석

그림 6은 주요 분석 정보 중 결함 유형별 통계 그래프에 대한 화면이다. 이를 통해 프로젝트 관리자는 해당 프로젝트에 어느 유형의 결함이 많이 발생했는지 파악이 가능하다. 프로젝트 관리자는 이러한 통계 자료를 근거로 이후 유사 프로젝트 진행 시 이러한 유형의 결함에 대비, 개발자에게 부족한 기술 교육이나, 공수의 조정, 장비 추가 등 사전에 프로젝트의 위험을 낮출 수 있다.

6. 결론 및 향후 연구 방향

본 논문에서 제안한 Winner 및 가이드라인을 통해 프로젝트 관리자는 다음과 같은 효과를 볼 수 있다.

1. 결함 데이터를 통해 개발자 개인 별 취약점을 파악하고 이를 프로젝트 관리 및 계획에 반영할 수 있다.
2. Impact가 큰 결함의 재발생을 막기 위해 예방 차원의 조치를 취할 수 있다.
3. 프로젝트 관리자는 결함을 제거하는 방법이 어떤 것이 효율적인지에 대한 정보를 정량적으로 파악할 수 있다.

조직 차원에서는 다음과 같은 효과를 볼 수 있다.

1. 조직이 CMMI 성숙도 수준 별로 결함 데이터 수집이 가능하다.
2. 사용하지 않을 쓸모없는 데이터를 수집하지 않는다.
3. 조직 차원에서는 어느 단계, 어느 프로세스에 결함이 많이 유입되는지 파악함으로써 조직의 표준 프로세스 개선 정보에 도움을 줄 수 있다.

Winner는 다양한 통계 기능을 통해 각종 결함 데이터 분석 활동을 지원하고, 조직의 결함 데이터를 CMMI 성숙도 수준에 따라 관리할 수 있도록 하였으며, 각 수준마다 필요한 결함 관리에 대한 가이드라인도 제시하였다. 이를 통해 CMMI 기반의 조직이 품질을 정량적으로 관리할 수 있는 토대를 마련하였다.

또한 Winner는 CMMI 성숙도 수준 3을 획득하는데 필요한 PAL(Process Asset Library)과 더불어 활용될 수 있다. 그러나 Winner는 CMMI 성숙도 수준 3까지의 조직에 적합하게 개발되어 결함 제거 위주의 활동을 수행하는 프로세스에 초점을 맞추었다. CMMI 성숙도 수준 4, 5의 결함 관리 활동을 하기 위해서는 결함 예방에 관한 연구가 선행되어야 한다. 이를 위해 결함 예측 모델 및 결함 원인들 간의 상관관계 분석이 이루어져야 한다. 또한 Winner를 통해 할당 받은 결함을 수정한 후에는 형상 관리 툴과 연동하여 바로 제품 빌드를 가능하게 하는 것이 필요하다. 이를 통해

CMMI의 모든 성숙도에 걸쳐 결함 관리 활동을 수행할 수 있다.

참고 문헌

- [1] Bruce Durbin, "Q/A Defect Tracking Process", www.stickyminds.com, 2001.
- [2] N. E. Fenton and S.L. Pfleeger. Software Metrics, a Rigorous and Practica Approach, second edition. International Thomson Computer Press, 1996.
- [3] Mitch Allen, "Bug Tracking Basics - A beginner's guide to reporting and tracking defects", STQE, pp.20-24, May/June 2002.
- [4] 안유환, 김신애, 허희정, "Bug Tracking 시스템을 활용한 테스트 및 결함관리," 제8회 SW 품질관리 심포지움 논문집, pp.457-464, 2003.
- [5] 권원일, "통합 사용자 기반 소프트웨어 결함 추적 시스템," KCSE, 2004.
- [6] <http://linas.org/linux/pm.html>
- [7] <http://testingfaqs.org/t-track.html>
- [8] <http://www.mantisbt.org/>
- [9] Matthew P. Barnson, "The Bugzilla Guide 2.16.3Release, <http://www.bugzilla.org/docs216/html/>,"April 2003.
- [10] Pankaj Jalote, 송태국, 이비즈온 SEOG 공역, 구현 사례를 통한 CMM 이해, Person Education Korea, 2002.
- [11] Wohlin,Runeson, «Defect Content Estimations from Re-view Data,"Proceedings International Conference on Software Engineering ICSE, pp.400-409, 1998.
- [12] Mary Beth Chrissis, et al., "CMMI guidelines for process integration and product improvement", Addison-Wesley Professional, 2003.
- [13] 이민재, 박남직, "CMMI의 이해," (주)피어슨에듀케이션코리아, 2006.

- [14] Mike Phillips, "CMMI Version 1.2 and Beyond a tutorial", USA SEPG, 2006.
- [15] Brehmer, C., Carl, J. R., "Incorporating IEEE std 1044 into Your Anomaly Tracking process," Crosstalk, no. 40, pp.9-16, Jan. 1993.
- [16] Humphrey, W.S., Introduction to the Personal Software Process, Addison Wesley, 1999.

● 저자 소개 ●



조 성 민(Sung-Min, Cho)

2003 상명대학교 소프트웨어학과 졸업(학사)
2007 상명대학교 일반대학원 컴퓨터과학과(석사)
관심분야: 소프트웨어 프로세스, 소프트웨어 품질, CMMI
E-mail: upoi@smu.ac.kr



한 혁 수(Hyuk-Soo, Han)

1985 서울대학교 계산통계학과 (학사)
1987 서울대학교 계산통계학과 (석사)
1992 Univ. of South Florida 전산학과 (공학박사)
1993 ~ 현재 상명대학교 소프트웨어 대학 소프트웨어 학부 교수
2003. 1 ~ 2004. 2 한국소프트웨어진흥원 소프트웨어공학센터 소장
관심분야 : 소프트웨어 프로세스, 소프트웨어 품질, 소프트웨어 아키텍처, 소프트웨어 사용성 평가 등
E-mail: hshan@smu.ac.kr