

# CMMI기반 프로세스 개선을 통한 장애 예방 연구

배장준\* 박승현\*\*

고려대학교 컴퓨터정보통신대학원\*

고려대학교 컴퓨터학과\*\*

e-mail: jjbae@korea.ac.kr\*, shpark@formal.korea.ac.kr\*\*

## A Study on Defect Prevention for Process Improvement based on CMMI

Jang-Jun Bae\* and SeungHyun Park\*\*

Graduate School of Computer and Information Technology, Korea University\*

Dept. of Computer Science and Engineering, Korea University\*\*

### 요 약

오늘날 많은 기업들이 납기 단축, 비용 절감 및 효율적인 프로세스 관리 등을 위해 소프트웨어 프로세스 모델 및 표준을 도입하고 있다. 이러한 표준들은 유지보수 과정에서 빈번하게 발생하는 장애를 혁신적으로 감소시켜 안정적 서비스 제공을 한다. 본 논문에서는 CMMI에서 요구되는 프로세스를 좀 더 자세히 체계화하여 CMMI 인증을 받은 금융회사의 장애분석을 통하여 장애의 원인들을 파악하고 이에 대한 개선된 프로세스를 제안하고자 한다.

### 1. 서론

오늘날 많은 기업들이 납기 단축, 비용 절감 및 효율적인 프로세스 관리 등을 위해 SPICE와 CMMI와 같은 소프트웨어 프로세스 모델 및 표준을 도입하고 있다. 이러한 모델은 기업의 현재 프로세스 상태에 대한 이해와 평가를 통해 Product 또는 서비스의 개발, 조직의 공정 및 관리능력을 향상시키기 위한 가이드를 제공한다.[1] 이러한 CMMI는 SW개발 및 시스템 공학을 개선하려는 기업에 새로운 관점과 자세를 요구한다.

품질은 비용, 일정과 함께 프로젝트의 성공을 결정하는 중요한 요소이다. 소프트웨어는 다양한 품질 관련 특성을 가지고 있는데, 품질을 위한 국제표준도 있다. 인도된 결함밀도, 즉 인도된 소프트웨어를 개발하는 것은 사람의 활동이므로 결함이 발생하는 것을 완벽히 막을 수는 없다. 고품질의 소프트웨어는 적극적으로 결함을 제거하려는 노력이 필요하다. 결함은 검토투와 테스트 같은 품질 통제 활동을 통해서 제거 할 수 있다.[2]

소프트웨어의 품질을 정량적으로 통제 할 수 있다

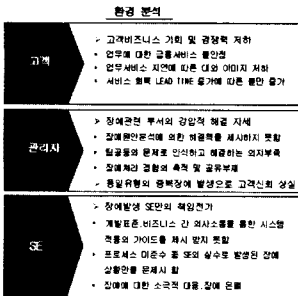
면 즉 정량적인 목표를 달성하기 위해서 정량적인 데이터를 바탕으로 소프트웨어 프로젝트를 통제한다면 측정기준 데이터를 분석할 때 이런 통제를 위해 개선되어야 한다.[3] 프로젝트의 품질 목표를 달성하기 위해 프로세스를 설정하면, 결함수준을 예측하여 품질을 정량적으로 조정할 수 있다. 이러한 프로세스 설정은 장애에 대한 통제와 향후 장애 발생 예방을 목적으로 한다.

프로젝트 중에 수행되는 측정기준 분석은 일차적으로 프로젝트를 정확히 통제하기 위한 것이다. 어떤 분석은 결함의 특성을 이해하고, 나중에 이런 결함을 방지하는데 도움이 되는 결함 분석에 중점을 두어 보완할 수 있다. 어떤 형태의 결함예방이 조직의 성숙도 레벨에 관계없이 모든 조직 안에서 이루어진다. 어떤 의미에서 모든 표준 방법론, 규칙 등의 목표는 결함을 방지 하는 것이다. 그러나 실제 데이터를 사용할 때에는 결함데이터 분석을 통해 더욱 효과적으로 결함을 방지할 수 있다.[4] 본 논문에서는 CMMI 인증을 받은 금융회사의 장애분석을 통하여 장애의 원인들을 파악하고 이에 대한 개선된 프로세스

스를 제안하고자 한다. 2장에서는 관련연구와 문제점을 알아보고 3장에서는 개선된 프로세스를 제안하고 4장에서는 제안된 프로세스를 이행 후 개선된 효과에 대해 논의 하고 5장에서 결론 및 향후 연구 방향에 대해 논의하고자 한다.

## 2. 관련연구

### 2.1 장애 환경 분석



(그림1) 환경 분석

한 금융회사의 환경분석을 통해 고객의 입장에서 장애로 인하여 고객비즈니스 기회 저하 및 경쟁력 저하를 일으키고 있으며, IT서비스의 불안정, 업무 서비스 지연에 따른 대외 이미지를 저하시키고 있다. 관리자의 입장에서 장애원인분석에 의한 해결책을 제시하지 못하며, 장애를 인식하고 해결하려는 의지가 부족하다. SE의 입장에서 개발표준 및 프로세스를 인식하지 못하거나 가이드를 제시 받지 못하고 장애에 대한 소극적 대응과 장애를 은폐시키고 있다.

이에 대한 시사점으로 개발 프로젝트 및 유지보수 과정에서 빈번하게 발생하는 장애를 혁신적으로 감소시켜 안정적 서비스 제공의 기반 확보가 필요하다.

<표 1> CSR에 대한 정상적 건수와 장애건수

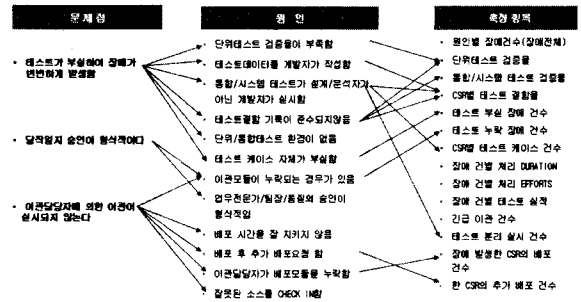
정상 CSR	총 장애건수	휴먼장애 건수	시스템장애 건수
200	15	12	3
233	10	8	2
248	17	12	5
186	8	7	1
194	12	8	4
242	21	15	6
<b>1303</b>	<b>83</b>	<b>62</b>	<b>21</b>

표에서 월별 CSR에 대한 정상적으로 처리 된 건수와 장애건수를 보여 준다.

고객에게 제공되는 서비스의 중단 및 그에 해당하는 피해를 유발하는 전체 장애 중 프로세스 미준수 등 SE의 실수로 발생한 휴먼 장애가 75%를 차지하는 것을 볼수 있다.

이슈 파악을 통해 문제점을 정의하여 원인을 도출하고 핵심요인을 찾아내기 위하여 수집 되어야 하는 데이터를 파악하였다.

<표 2> 장애에 미치는 상관관계 분석



이슈 파악을 통해 문제점을 정의하여 원인을 도출하고 핵심요인을 찾아내기 위하여 수집되어야 하는 데이터를 파악을 하였고 FACT검증을 통하여 원인들 이 장애에 미치는 상관관계를 분석해 핵심 요인을 도출하였다.

### 2.2 문제점

집계된 장애 자료를 업무유형별로 분석한 결과 휴먼장애의 70%가 CSR을 처리하는 과정에서 발생하는 것으로 조사 되었고 그 중 87%가 테스트와 이관 단계에서 발생하는 것으로 파악이 되었다.

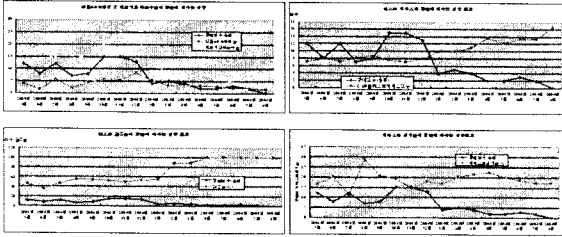
테스트에 의한 빈번한 장애발생 유형으로는 테스트 데이터를 개발자가 작성을 하거나, 통합테스트 시스템테스트를 개발자가 담당하는 경우이며 또한 테스트 결함 기록이 준수되지 않음을 알수 있다. 이관담당자에 의한 이관이 일어나지 않음으로 배포시간이 지켜지지 않으며 이관담당자가 배포 모듈을 누락하여 장애가 발생을 하고 있다.

SE들의 전체 업무를 인터뷰 및 장애 자료 등, 장애 현상을 파악한 결과 대부분의 휴먼 장애는 CSR처리 단계의 테스트와 이관에 집중되었고, Data보정 작업에 대한 개선안은 즉시 해결할 수 있는 것으로 결론을 내렸다.



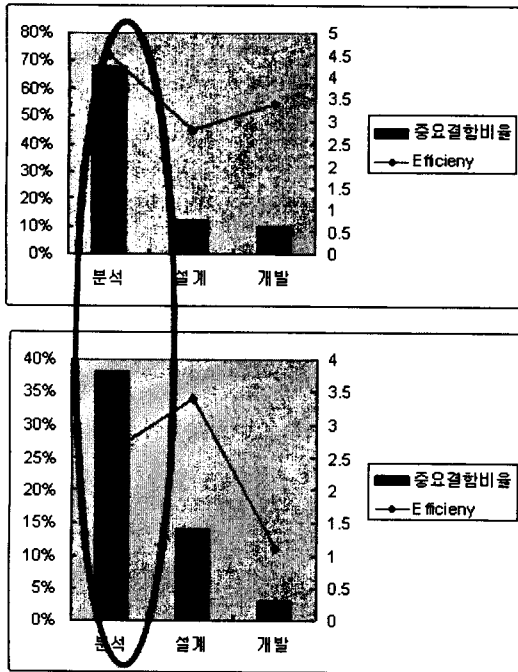
장애발생건수의 감소됨을 알 수 있다.

### 4.3 Fix Inspection 도입으로 인한 효과 분석



(그림6) Fix Inspection 효과분석

Formal Inspection에 소요되는 Effort로 인해 Inspection 건수가 점차 감소하므로 Fix Inspection을 도입하여 적시에 효과적인 검토를 수행할 수 있었다.



(그림7) Fix Inspection 효과분석

중요결함비율과 Efficiency가 개선됨을 볼 수 있다.

### 5. 결론 및 향후 연구과제

이상으로 장애에 대한 환경 분석 및 근본적인 원인을 분석하고 장애감소 및 장애예방에 관한 프로세스 개선 제안을 하였다. 장애처리 비용은 장애처리 시 발생하는 품질비용과 고객기회비용을 의미한다. 프로세스 개선을 통하여 장애비용을 현저히 줄일 수 있다. 향후에도 꾸준한 프로세스 개선 활동과 충분한 내재화를 통한 활동이 요구될 것이다.

향 후 과제로 본 논문에서는 세 가지 프로세스에 관한 개선안을 적용 하였는데 이 외에도 장애에 미치는 요소들에 대해서도 연구해 보고자 한다.

### 참고문헌

- [1] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3rd Ed. McGraw Hill
- [2] 송태국 구현사 예를 통한 CMM이해
- [3] International Standards Organization Information Technology Software Product Evaluation Quality Charateristics and Guidelines for Their Use ISO/IEC IS 9126,Geueoa,1991
- [4]R, Grady, Practical Software Metrics for Project Management and Process Improvement. Prentice Hall,1992l