

결함 데이터의 요구사항 분석단계 활용을 통한 품질 향상 방안

고경희^o, 이은석

성균관 대학교 정보통신 공학부 전기전자컴퓨터 공학과

경기도 수원시 장안구 천천동 300, 440-746

e-mail : kkh1004@skku.edu, eslee@ece.skku.ac.kr

A Study on the quality improvement method in the requirement analysis phase using defect datum

Kyounghee Ko ^o, Eunseok Lee

300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746

Sungkyunkwan University

[요약]

SQA 엔지니어는 품질을 보증하기 위해 전 단계 특히 개발 초기부터 관여하여 궁극적으로는 장차 발생할 품질 관련 비용을 줄여야 한다. 대체적으로 SQA 엔지니어는 테스트 단계에서 확인(validation)하는데 노력과 자원을 많이 소비하고 있는데 불행히도 결함은 늦게 발견되면 늦게 발견될수록 고치는데 비용이 급격하게 증가한다. 본 논문에서는 이러한 문제에 대한 대응 방안으로 SQA 엔지니어가 테스트 단계 이전 특히 결함 개선이 쉬운 요구사항분석단계에 결함 발견 활동을 효율적으로 수행할 수 있게 하는 “요구사항 품질 체크리스트 추출 프로세스”를 제안하였다. 과거의 결함 데이터를 지표 추출이나 관리용도로 사용하는데 그치지 않고 결함데이터의 원인을 분류하고 의미를 부여함으로써 새로운 요구사항과 추적된 결함들과의 연관성을 도출하여 요구사항 품질 체크리스트를 작성하고 요구사항 분석에 적용하여 요구사항 분석단계 발견 결함밀도를 높이고 테스트 단계 이후에 발견되는 요구사항 결함을 줄이는 효과를 확인할 수 있었다.

[필요성]

요구사항 분석과 관련하여 발생하는 문제는 다음과 같다. 첫째 요구사항 문서에 대한 인스펙션(Inspection)을 수행하였음에도 불구하고 SQA가 제품 또는 시스템을 이관 받아 테스트를 수행하고 나서 발견되는 결함들을 살펴보면 원인이 요구사항 오류인 결함이 존재한다. 둘째 요구사항에 대한 관련자들의 동의가 이뤄지지 않는 경우가 많다. 즉 기능을 요구한 사람과 개발하는 사람간의 의견일치가 되지 않아 개발된 시스템이 사용자에게 도움이 되지 않는 경우이다. 셋째 요구사항을 정의하는 사람들 중 상당수는 요구사항을 생략하여 기술하거나 애매모호하게 기술하고 있다. 마지막으로 변하는 사회 환경이나 규범 규정 등에 대한 고민 없이 개발 될으로써 나중에 규율 또는 사회 환경 변화에 따른 요구사항을 충족시키는데 생각보다 많은 비용이 발생한다.

[제안사항]

제안시스템의 구성은 그림 1과 같다. 이 구성을 만족시키는 활동들은 그림 2와 같다.

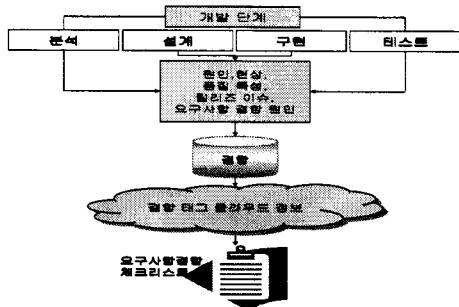


그림 1 제안 시스템 개요

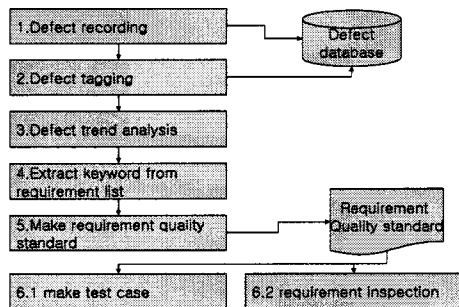


그림 2 제안 프로세스

주요 활동인 결함 태깅 작업을 상세화 하면 그림 3과 같다. 결함에 대해 분야별로 관련 정보를 자유롭게 입력하도록 한다. 이 논문에서는 결함을 세 측면에서 분석 즉 태깅 하였다. 요구사항 결함 원인(Requirement error), 관련 품질특성(Quality attribute), 해당 릴리즈 이슈(Release issue) 측면에서 분석하였다.

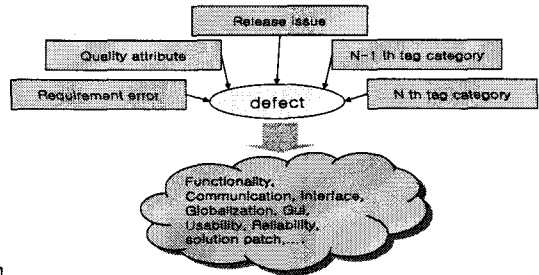


그림 3 결함 태깅

[적용 결과]

결함에 태그를 적용한 결과를 태그 클라우드로 전환하여 보면 그림 4와 같다.

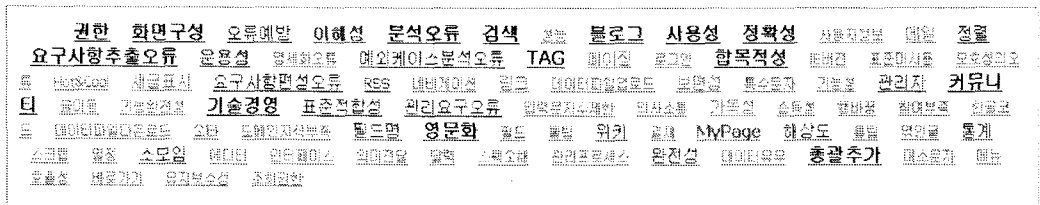


그림 4 결함 태그 클라우드

위 태그 정보를 가지고 새로운 요구사항에 관련된 결함 정보를 추출하여 요구사항 체크리스트로 작성하고 이 체크리스트를 적용하지 않고 인스펙션을 실시하였을 때와 체크리스트를 적용하고 인스펙션을 수행하였을 때를 비교한 결과 체크리스트를 적용하였을 때 더 많은 결함을 발견할 수 있었다. 그림 5에서 확인하면 적용해본 시스템 S1과 S2의 시스템 특성에 따른 결함 발견 밀도의 차이는 있지만 모두 결함 발견 밀도가 증가하였다. 또한 체크리스트를 적용한 인스펙션을 실시한 후 미리 발견된 결함과 테스트를 수행하여 발견된 요구사항 결함을 비교하여 예방 가능 했던 결함수를 추출한 결과 표1에서 보는 바와 같이 줄어들었다.

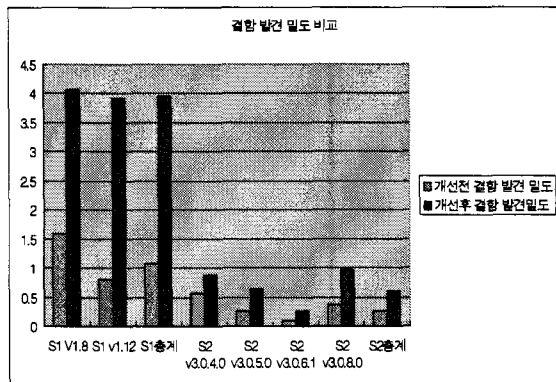


그림 5 개선 전후 발견 결함 밀도

표 1 테스트 단계 예방 가능한 요구사항 결함 수

구분	개선 전 테스트 단계 발견된 요구사항 결함 수	예방 가능한 요구사항 결함 수	개선 후 테스트 단계 발견된 요구사항 결함 수
S1 v1.8	43	10	33
S1 v1.12	29	27	2
S2 v3.0.4.0	1	1	0
S2 v3.0.5.0	1	0	1
S2 v3.0.6.1	0	0	0

[결론]

개선 시스템은 요구사항 결함을 분류하고 원인을 분석하므로 SW프로세스 개선에 활용할 수 있고 과거에는 특별히 재사용하지 않던 결함정보를 재사용 하게 하고 같은 실수를 반복하는 것을 예방하여 결함을 고치는데 드는 비용을 줄일 수 있다. 또한 결함관리에 Web2.0의 개념인 태그를 도입하여 유연한 결함 관리 방안을 제시하였다.