통합사용자기반 소프트웨어결함 추적시스템 개발 연구

최성*, 한정란**
*남서울대학교 컴퓨터학과
**협성대학교 경영정보학과
e-mail : sstar@nsu.ac.kr

A Case Study on the Software Defect Tracking System for Integrated User based

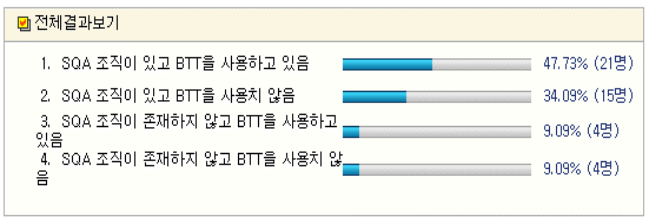
Sung Choi*, JungLan Han**
*Dept. of Computer Science, Namseoul University
**Dept. of Management Information System, Hyupsung University

요 약

본 연구에서는 기존의 소프트웨어 결함 추적 시스템에 대한 벤치마크를 수행하여 기능을 비교, 분석하고, 그 결과를 바탕으로 국내에서 최초로 웹기반 결함 추적 시스템을 개발하였다. 개발된 결함 추적 시스템에 대한 개발사례를 기술하고 해당 시스템이 테스팅 조직의 생산성 향상에 기여하는 정도를 연구결과로 증명하였다.

1. 개발배경

IT 산업의 발전에 따라 소프트웨어 품질의 중요성에 대한 인식이 많이 제고 있다. 하지만 국내 현실은 소프트웨어 품질을 향상시키기 위해 필수적인 테스팅 과정에 거의 투자가 되지 않고 있다. 현재 국내 소프트웨어 업계의 결함 추적 시스템 사용 현황을 조사한 결과는 아래 그림 1 과 같다. 48%정도가 사용하고 있는 것으로 나타나지만 설문 대상자의 30%정도가 대규모 SI 조직에 있는 것을 감안하면, 실제 사용하는 업체는 대략 전체 S/W 업체의 20~30% 정도가 되는 것으로 판단된다 [3].



(그림 1) 결함 추적 시스템 도입 현황

본 논문에서는 소프트웨어 QA 조직이나 개발 부서에서 결함 추적 시스템을 도입하는데 도움을 주기 위해 업계에서 많이 사용하고 있는 시스템에 대해 벤치마크를 수행하여 기능을 비교•분석하였다. 또한 벤치마킹 결과와 실무 경험을 바탕으로 국내 환경에 적합하면서 통합적인 사용자 관리가 가능한 시스템을 개발한 사례를 통하여 시험결과로 증명하였다.

2. 소프트웨어 결함 추적 시스템 개요

소프트웨어 결함 추적 시스템 (Software Defect Tracking System)은 하나의 제품을 개발하는 과정에서 발생되는 프로그램의 오류를 관리할 수 있도록 도구 형태로 제공되는 것으로 오류 보고자와 오류 수정자의 의사 소통(Communication)을 원활히 할 수 있도록 도와준다. 이 도구는 데이터베이스(Database)를 근간에 두고 결함을 데이터베이스에 레코드 (Record) 형태로 입력(생성)/수정/삭제하면서 결함 생명 주기 (Defect Lifecycle)에 따라 결함을 추적하는 결함 관리 시스템이다. 여기서 결함(Defect)은 버그 (Bug), 오류 (Error), 해결대상(Issue) 등과 혼용하여 사용하고 있다. 같은 맥락에서 결함 추적 시스템은 버그 트래킹 시스템 (Bug Tracking System)이라고도 한다. 결함 추적 시스템의 주요 기능에는 결함의 키워드 검색과 테

스터의 시스템 사용 권한 관리 등 이 있다[2].

2.1 결함 생명 주기

소프트웨어 결함 추적 시스템은 업계표준(de facto standard)인 결함 생명 주기 (Defect Lifecycle)에 근간을 두고 대부분 유사한 기능을 구현하고 있다. 결함생명주기는 아래 표 1 과 같다[1].

<표 1> 결함 생명 주기 (Defect Lifecycle)

단계	내용	용어	주체
1	결함 발견	Open	QA 테스터
2	결함 분배	Assign	관리자
3	결함 검토	Dev Review	개발자
4	결함 수정 또는 불인정	Fix or Decline	개발자
5	결함 처리 후 QA 로 전달	Assign back	개발자
6	결함 처리결과 검토	QA Review	QA 테스터, 관리자
7	결함 수정 확인 및 종료	Verify, Close	QA 테스터, 관리자
8	결함 다시 보고 및 재분배	Re-open, Re-assign	QA 테스터, 관리자

이러한 결함 생명 주기에 영향을 주는 부서는 크게 결함 보고 부서와 결함 수정 부서이다. 결함 수정 부서는 보고된 기능 오류를 검토하고 해결하는 작업을 수행한다. 반면, 결함 보고 부서는 결함 발견 및 보고(Open), 개발자에 의해 해결된 결함 수정 확인(Verify), 수정되지 않은 결함 다시 보고(Reopen), 수정된 결함 종료(Close) 등의 작업을 수행한다.

여기서 "결함 확인 (Verify)"은 발견된 결함이 개발자에 의해 보고된 오류가 수정되어, 수정 사항이 적용된 새로운 제품(build)에서 확인했을 때 문제가 수정되었음을 확인한 상태를 의미한다.

"결함 종료(Close)"는 제품 테스트를 종료하는 최종 단계에서만 표시할 수 있는 오류의 상태이다. 테스트 최종 제품(build)에서 지금까지 보고된 오류 중 수정된 오류 즉, "결함 확인(Verify)" 상태의 오류를 재현해보고 기존에 수정된 결함이 모든 운영시스템(OS)에서 발생하지 않고 시스템에 다른 문제를 일으키지 않는지 확인한 후 이상이 없을 경우에만 오류의 상태를 "종료"로 변경할 수 있다.

이는 회귀 테스트 (Regression Test) 과정에서 예기치 못한 결함이 발생하여 그 결과 종료한 결함을 다시 "Re-open" 해야 하는 경우가 많기 때문이다. 이렇게 함으로서 개발자나 형상 관리 담당자의 파일 관리에 있어서 잘못된 파일(오류 수정 이전의 파일)이나 오류를 수정하면서 다른 오류를 발생하게 하는 실수를 감지해 낼 수 있도록 한다.

이외에도 "연기(Defer)" 상태가 존재하며, 이는 경영진과 개발/QA 조직 모두가 중요하지 않은 문제이므로 또는 현재 해결할 수 있는 방안이 없어서 다음으로 수정을 미룰 때를 의미한다. "중복 (Duplicate)"은 이미 생성되어 있는 결함을 반복하여 생성하였을 때를 의미한다.

이 밖에도 "정보 부족 (NMI – Need More Information)", "결함 아님 (NAB – Not A Bug)", "재현 불가(NREP – Not Reproducible), "수정 계획 없음 (NPTF – No Plan To Fix)", "사용자 오류(User Error) 등의 상태가 존재한다[2].

2.2 결함 리포트

결함 추적 시스템은 테스트 중간 또는 종료 시점에서 결함 리포트를 자동적으로 출력해 준다. 결함 리포트에는 기본적으로 아래의 내용(Bug Components)이 포함되며, 시스템 사용자의 요구에 따라 다양한 결함 리포트를 출력할 수 있다. 해당 항목은 시스템을 사용하여 테스트를 진행하는 동안 사용자 인터페이스를 통해 입력되어야 하는 내용이다[2].

1. 결함번호 (Bug Number)
2. 요약 기술 (Short Description)
3. 제품/프로젝트 이름 (Product/Project Name)
4. Product Version
5. Build Number
6. Fixed Build Number
7. Steps to reproduce
8. Comments
9. Author
10. Test Environment(O/S, Web Browser)
11. Category
12. Date(Submitted/Updated/Target)
13. Developer Assigned
14. Developer Resolution
15. QA Assigned
16. State
17. Priority
18. Severity
19. History
20. Attachment

– Miscellaneous/Custom

3. 결함 추적 시스템 개발

결함 추적 시스템의 일반적인 기능과 상용제품 벤치마크 결과, 결함 생명 주기, 결함 리포트 등을 참고하고, 베타 테스트 관리를 위해 운영하던 시스템을 기반으로 국내 최초로 제품을 개발하였다.

내부 테스터, 고객기업의 테스터, 외부 사용자 테스터를 관리하고, 테스트 프로젝트를 관리하기 위한 시스템을 개선해 온 경험을 바탕으로 상용 제품을 기획하게 되었으며 그 결과물이 소프트웨어 결함 추적 시스템(Software Defect Tracking System)이다.

3.1 시스템 디자인

먼저, 결함 추적 시스템의 사용자 환경과 서버 환경은 다음과 같다.

[사용자 환경]

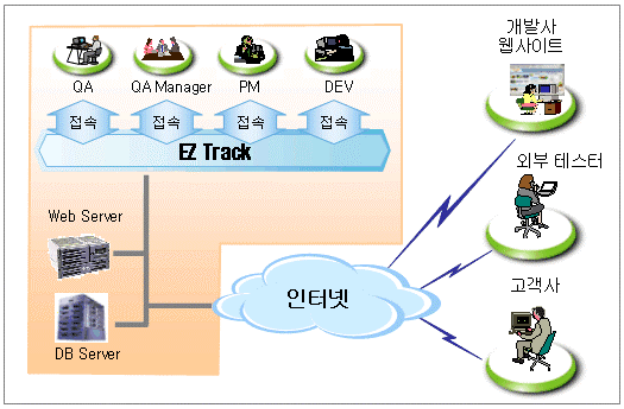
- ✓ CPU: Intel Pentium Processor(혹은 그 이상)
- ✓ OS: Microsoft Windows 계열

✓ 메모리: 32 MB 이상
✓ Browser: Explorer 5.0 이상

[서버 환경]

✓ OS : Linux
✓ DB: MySql

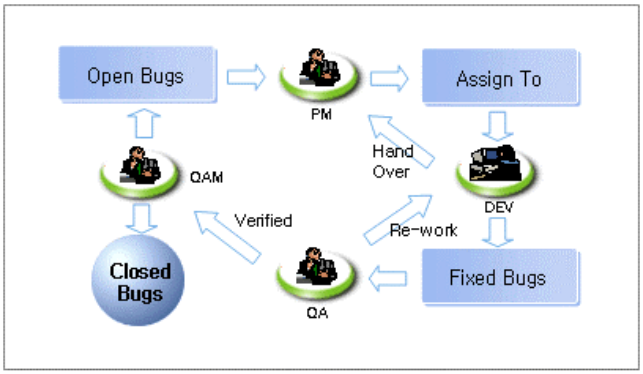
소프트웨어결함추적시스템(Software Defect Tracking System)은 결함 발생 가능성을 조기에 예측하여 개발 기간과 비용을 줄이고, 짧은 시간에 소프트웨어의 품질을 효율적으로 향상시키는 것은 물론, 개발팀과 QA 팀간에 빠르고 정확한 피드백 및 긴밀한 커뮤니케이션을 할 수 있도록 개발된 시스템이다. 시스템 구성 도는 아래 그림 2 와 같이, QA 팀은 내부 테스트를 관리하면서 외부 테스터를 인터넷을 통해 관리하고, 자사 웹사이트 및 고객 사로부터 반응도 관리하는 것을 고려하였다.



(그림 2) 시스템 구성도

시스템 설계는 QA 관리자가 내부 테스터와 개발자를 관리하면서 내부 테스트 프로젝트를 진행하고, 외부 사용자와 고객사의 테스터는 내부 테스트 프로젝트와 분리하여 관리하는 사용 시나리오(Usage Scenario)를 기본으로 하였다.

기본적인 업무 흐름 (Work Flow)은 결함 생명주기를 그대로 반영하여 아래 그림 3 과 같은 흐름을 따랐다. 이에 대한 설명은 "2.1 결함생명주기" 부분을 참고하기 바란다.



(그림 3) 업무 흐름도

3.2 결함추적시스템업무 흐름

결함추적시스템은 사용 시나리오와 업무 흐름도를 중심으로 "오류 보고 -> 오류 배분 -> 오류 처리 -> 오류 확인 -> 종료" 등의 처리 과정으로 구현 하였다.

[오류 보고] 보고서 작성 시 보고자의 컴퓨터 환경을 자동으로 분석하여 제공 함으로서 잘못된 시스템 정보의 입력으로 인한 오류 해결 실패율을 낮추도록 하였으며, 쉽고 간편한 화면 캡쳐 시스템을 통해 리포트 작성에 소요되는 시간을 최소화 하였고, 오류가 발생한 화면을 첨부 파일이 아닌 보고서 자체에 바로 출력하여 담당자가 오류에 대한 이해를 하는데 용이하도록 디자인 되었다.

[오류 배분] 오류 배분에 대한 과정에서의 문제점은 오류를 배분하는 관리자(PM 또는 책임개발자)가 그 역할을 수행하지 못하는 상황이 발생하는 경우 배분 과정에서 정체 된다는 점이다.

이를 해결하기 위해 관련 담당자(부책임자)에게 배분 권한을 설정할 수 있게 하여 담당자가 역할을 대신 처리할 수 있게 함으로써 프로세스상의 문제점을 보완할 수 있도록 지원한다.

[오류 처리] 오류 처리 과정은 프로젝트 관리자나 책임 개발자가 현재의 진행상태를 명확히 판단할 수 있도록 하여야 한다. 이를 위해 담당자가 현재 오류 상태를 다양하게 설정할 수 있도록 지원 하며, 이는 시험 프로젝트 진행상태를 보여주기 위한 기초 자료가 되어 통계로서 보여진다. 추가적으로는 이러한 상태 설정을 통해 해당 담당자는 오류에 대한 명확한 책임을 가지게 된다.

서로 다른 팀 조직인 QA 팀과 개발팀간의 원활한 커뮤니케이션을 위해 각각의 처리 과정에서 발생하는 모든 기록으로 남게 된다. 발생된 주요내역은 다음 과정의 담당자에게 자동적으로 이메일을 통해 통지되어 처리 과정에서 걸리는 담당자 별 커뮤니케이션 시간을 최소화 하였다.

[오류 확인] 위의 과정을 통해 수정된 오류는 QA 팀에 의해 오류 수정 확인 단계를 거친다. 이상이 없을 경우 "Verified" 상태가 되며 수정 되지 않거나 문제점이 있을 경우 "Re-Work"를 통해 개발자에게 다시 전달되어 재 수정 하도록 하였다.

[종료] QA 에 의해 "Verified"된 오류는 프로젝트 종료 시점에 다시 한번 재 확인 작업을 거쳐 최종적으로 이상이 없을 경우 "Closed"된다. 이러한 과정을 통해 오류를 효율적으로 관리할 수 있어 높은 품질의 소프트웨어를 생산할 수 있게 된다.

4 개발 시스템의 강점 및 기여사항

내부 테스터, 고객기업의 테스터/개발자, 대규모 외부 사용자 테스터 등을 체계적으로 관리하는 기능을 다년간의 실제 테스트 적용 과정을 거쳐 개발하였다. 따라서, 이 결함 추적 시스템은 사용자 통합 관리 기능이 우수하다. 개발사 홈페이지의 버그리포트 게시판을 통해 보고되는 오류 및 제안 사항을 시스템으로

통합하여 관리가 가능해 짐으로써 사용자 의견을 체계적으로 관리하고 제품에 적용이 가능하게 되었다.

기업내부 결함 관리 프로세스에 따라 버그 상태를 생략하거나 반드시 지켜야 할 부분을 제시한다는 장점이 있다. 예를 들어, 최종 출시될 제품(Build)으로 기존에 보고되어 수정된 오류를 다시 한번 모두 재현해 볼 수 있도록 버그 상태를 수정(Fix)과 종료(Close) 단계 사이에 확인(Verify)과 같은 단계가 필요한데 이를 시스템적으로 구현하고 있다. 즉, 결함 추적 시스템을 사용할 경우 최종 출시될 제품에서도 수정된 오류가 적용 되었음을 확인했을 때만 비로서 하나의 오류가 종료되도록 시스템을 구현하고 있다. 또한 기업의 필요에 따라 오류 보고서 작성시 곧바로 "결함 배분(Assign)"이 가능할 수 있도록 하는 등 프로세스의 변경에 따른 조정이 가능하도록 하는 인터페이스를 제공한다.

오류 생명 주기에 따라 오류를 작성하고 관리하는데 소요되는 평균 시간을 분단위로 계산하면 아래 표 2 와 같다.

<표 2> 오류 작성/관리에 소요되는 평균 시간

(단위 : 분)

시간 소요 내역	Spread sheet	결함추적 시스템	Mantis
오류 내용 작성	5.0	5.0	6.0
PC 사양정보 입력	0.5	-	0.1
화면 캡처 후 입력	3.0	0.5	1.0
오류 보고 문서 작성 및 전송	20.0	5.5	8.0
오류 확인/수정 후 통보	10.0	-	-
오류 정보 업데이트	10.0	2.0	1.3
오류 내용 (파일) 관리	30.0	-	-
입력 실수 검토, 추적, 발견, 처리	30.0	3.0	2.0
오류 내용 검색	5.0	1.0	1.0
오류 내용 추출	5.0	1.0	2.0
보고서 작성	30.0	20.0	30.0

여기서, 오류 내용 작성, PC 사양 정보 입력, 화면 캡쳐 후 입력 등은 각각의 오류 1 개 당 소요되는 시간이다. 반면, 오류 보고 문서 작성 및 전송, 오류 정보 업데이트, 오류 수정 사항 확인 후 처리, 오류 내용 검색, 오류 내용 추출 등은 여러 개의 오류를 묶어서 처리할 수 있는 형태이다. 또한 오류 내용 (파일) 관리, 입력 실수 검토/추적/발견/처리, 보고서 작성 등은 하나의 테스트 프로젝트에서 1 회만 작업하면 처리 가능하다.

하나의 테스트 프로젝트에서 결함 추적 시스템을 사용할 경우, 트래킹 시스템을 사용하지 않을 경우보다 56% 이상의 시간 절감 효과를 볼 수 있고, Mantis 라는 시스템을 사용하는 것에 비해서는 12 %의 시간 절감 효과를 거둘 수 있다.

이 밖에도, 사용자의 요구에 따라 시스템 변경(Customization)이 가능하며, 시스템의 사후 관리를 원하는 시기에 체계적으로 받을 수 있다. 또한 리포트가 다양하며 그래프로 표현되어 다른 상용 제품에 비해 사용 성이 높으며 리포트 포맷의 변경이 가능하다. 즉, 한번 발견된 오류의 관리를 체계적이고 효율적으로 할 수 있도록 보고된 오류에 대한 분석 그래프를 날짜 단위로 표시함으로써 제품 개발 기간 동안에 발생된 오류의 증감을 한눈에 파악할 수 있었다

5. 결론

현재 개발된 결함 추적 시스템은 외국의 제품들과 기능과 성능 면에서 대동소이하다. 그러므로 이 시스템을 도입하면서 고려해야 할 기준은 "결함 관리 프로세스를 효율적으로 관리할 수 있도록 지원하는 시스템인가"라는 측면과, "테스터/개발자가 얼마나 쉽고 빠른 시간 내에 결함을 입력/검토할 수 있는가"의 측면, "오류 보고서 작성을 지원하기 위해 필요한 정보를 쉽게 한눈에 파악할 수 있도록 작성하는 기능을 제공하는가"라는 측면으로 볼 수 있다. 이 세 가지 부류의 기능이 제품의 우수성을 결정한다. 해당 조건을 만족하는 우수한 제품은 테스트 시간을 줄여주고, 관리를 체계적이고 효율적으로 할 수 있게 해주어 고품질의 소프트웨어 제품을 적시에 출시하는데 결정적인 역할을 할 것으로 사료된다.

참고문헌

[1] SW 시험을 위한 평가모델, SW 시험센타(TTA), 2001.11.17

[2] 김재웅, 신석규, 소프트웨어 벤치마킹현황, 한국정보처리학회지, 2005.3 월호소프트웨어 품질 및 시험특집

[3] 황석영,양해술 클래스계층구조 재구성과 품질평가방법, 한국정보처리학회지, 2005.3 월호특집

[4] 노성운, 최성, 미육군 교육용 AA 온라인 게임품질 모형에 의한 버그시험평가, 2005 년 3 월호한국정보처리학회지 특집

[5] 배현섭, 소프트웨어시험단계별 자동화지원도구, 한국정보처리학회지, 2005.3 월호특집

[6] Bruce Durbin, "Q/A Defect Tracking Process", www.stickyminds.com, 2001

[7] Mitch Allen, "Bug Tracking Basics - A beginner's guide to reporting and tracking defects", STQE, pp.20-24, May/June 2002

[8] STEN, no.739 국내 SQA(SW 품질보증)조직 현황 및 BTT(Bug Tracking Tool) 도입현황 파악, sten.cyworld.com, 2003.12

[9] www.stickyminds.com/defecttracking.asp, "Stickyminds.com – Defect Tracking-Tools", 2003.12

[10] www.web-based-software.com/tracking, "Software Reviews - Bug Tracking", 2003.12