버그리포트의 메타필드 초기 재할당의 실증적 분석

민세원*, 김미수**, 이은석*
*성균관대학교 컴퓨터공학과
**성균관대학교 전자전기컴퓨터공학과

e-mail : minsw08@skku.edu

An Empirical Study of Meta Field Reassignment on New Bug Report

Sae-Won Min*, Mi-Soo Kim**, Eun-Seok Lee*
*Dept. of Computer Science, Sungkyunkwan University
** Dept. of Electrical and Computer Engineering, Sungkyunkwan University

요 약

소프트웨어 개발 및 유지보수 단계에서 발생한 문제들은 버그 추적 시스템을 통해 버그리포트로 등록되고 관리된다. 등록된 버그리포트를 기반으로 배정자는 해당 문제를 해결할 수 있는 개발자들을 배정하고, 배정된 개발자는 이를 해결한다. 그러나 버그리포트에서 제공되는 정보가 정확하지 않을 경우 문제 해결에 많은 시간이 소모된다. 본 논문에서는 Eclipse 오픈소스 프로젝트들에 대해 12 가지의 도메인으로 분류하여 총 395,967 개의 버그리포트에 대해 초기 정보의 불완전성을 분석한다. 이를 위해 초기 버그리포트에서 제공되는 정보 중 메타필드 내 정보에 초점을 맞춘다. 분석 결과 필드들이 도메인 별로 최소 6%, 평균 20%, 최대 33%가 최소 한 번 이상 변경되는 것을 확인하였으며, 프로젝트 도메인 별로 상이하게 변경되는 것을 확인할 수 있었다.

1. 서론

소프트웨어 개발 및 유지보수 단계에서 어떠한 문제가 발생하게 되면 개발자, 테스터나 유저들은 문제 해결을 위해 버그리포트를 작성하여 버그 추적 시스템에 등록한다. 버그리포트는 크게 텍스트 필드와 메타필드로 구성된다. 텍스트 필드는 리포터가 자연어로 작성하는 부분이고 메타 필드는 리포터가 선택한 버그의 속성을 설명하는 부분이다. 개발자들은 위 정보들을 통해 버그를 해결하기 때문에 버그리포트의 정보는 정확해야한다. 그러나 버그리포트를 등록할 때 리포터들의 최초 작성이 잘못된 경우가 많기 때문에 버그리포트 값들의 변경이 자주 일어난다[1,2]. 이는 결과적으로 버그리포트를 해결해야 하는 개발자를 제대로 배정하지 못하거나[2], 문제 해결 시간을 증가시키는 요인이 된다[3].

X.Xia 는 4 개의 오픈소스 프로젝트 내 190,558 개 버그 리포트를 대상으로 버그리포트들의 메타필드 값 재할당에 관하여 분석을 수행 하였다[1]. 위 논문에서는 설문을 통해 버그리포트의 메타필드가 변경 되는 이유는 크게 잘못된 정보를 올바른 값으로 재할당(Reassignment)하는 경우와 해당 리포트를 해결하기 위해 할당된 개발자가 주관적으로 변경하여 정보를 강화(Enhancement)하는 경우가 있음을 확인하였다. 실증적 분석을 통해 최소 1 번 이상 변경되는 버그리포트의 비율이 80% 이상인 것을 확인하였다. 그러나 위 분석 결과는 메타필드가 변경되는 모든 경우에 대한 결과로써, 초기에 등록되는 버그리포트의 불완전성을 설명하지 못한다.

본 논문은 메타필드가 변경되는 큰 두 가지 이유 중 개발자가 할당되기 전에 잘못된 정보들이 변경되는 경우(Reassignment)에 초점을 맞춰서 초기 버그리포트의 불완전성을 분석한다. 분석을 위해 기존 연구보다 더 많은 총 395,967 개의 버그리포트를 사용하였으며, 도메인 별로의 불완전성이 상이한 것을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구를 3 장에서는 분석 결과, 4 장에서는 결론을 설명한다.

2. 관련 연구

T. Zimmermann 는 설문을 통해 낮은 품질의 버그리포트 때문에 문제 해결에 소모되는 시간이 증가한다는 것을 확인하였다 [2]. 또한 버그 추적 시스템과 문제 해결 프로세스의 개선을 통해 높은 품질의 버그리포트를 제공받을 수 있는 버그 추적 시스템을 제안한다[4] P. J. Guo 는 실증적 분석을 통해 잘못된 정보를 갖는 버그리포트는 5 가지 이유로 인해 적합한 개발자를 배정하지 못하는 것을 확인하였다[5]. 이를 통해 잘못된 버그리포트가 궁극적으로 소프트웨어 개발 및 유지보수 비용을 증가시키는 것을 확인할 수 있었다.

<표 1> 대상 프로젝트

Domain	# of Projects	# of Bug Reports
Birt	1	23,056
ECD	4	10,030
Eclipse	5	185,148
IoT	14	1,009
MyLyn	1	8,422
RT	20	36,212
Modeling	51	43,096
Science	5	355
SOA	7	2,043
Technology	54	37,453
Tools	25	46,552
Webtools	2	2,591
Total	**189**	**395,967**

X.Xia 는 필드가 최근 재할당 된 버그 리포트의 작성자와 개발자들에게 질의응답을 수행하여 버그리포트들의 재할당 원인을 크게 세 가지로 정리했다[1]. 첫째는 신규 버그리포트들의 작성 오류, 둘째는 버그 해결 진행 중의 수정 오류, 마지막은 관리자들의 일괄처리 오류라고 분류하였다. 그리고 버그리포트들의 재할당 비율, 필드 수, 필드 별 재할당 횟수, 재할당 시기, 해결 소요 시간, 리포터들의 경력과의 재할당 상관관계에 대한 실증적인 분석을 수행하였다. 또한 재할당 여부를 예측하는 기술을 제안하였다[6].

그러나 기존 연구는 단순한 통계 수치에 대해서 설명하고 있을 뿐, 해당 결과가 갖는 의미나 필드가 어떻게 변화하는 지에 대해서는 고려하지 않는다. 재할당 원인을 세 가지로 구분하고 있지만 실증적 연구에서는 원인과는 무관한 재할당 횟수와 시기만으로 분석한다. 또한 오픈 소스 프로젝트 4 가지로만 구분한 통계이기 때문에, 해당 프로젝트 내의 버그리포트들의 공통 양상으로 보기에는 모호한 부분이 있다. 본 논문에서는 초기 버그리포트의 불완전성을 분석하기 위해 기존 논문보다 더 많은 버그리포트를 사용하여 실증적 분석을 수행한다.

3. 실증적 분석 결과

3.1 대상 프로젝트

본 논문에서는 실증적 분석을 위해 Eclipse 기업 내 오픈소스 프로젝트들을 대상으로 공개된 14 개 도메인, 261 개 프로젝트의 510,699 개의 버그리포트를 수집하였다. 그 중 초기 재할당을 위한 분석만을 수행하기 위해 초기 재할당이력이 없는 프로젝트들은 제거한다. 최종적으로 총 12 개 도메인, 189 개 프로젝트의 395,967 개의 리포트를 대상으로만 분석하였다. 표 1 은 사용된 데이터를 정리한 표이다.

버그리포트들은 버그에 대한 텍스트 필드와 메타 필드를 갖는다. 표 2 는 본 논문의 분석 대상이 되는 버그리포트가 갖는 메타필드를 설명한다. 본 논문에서는 초기 재할당에 대한 분석을 위해 개발자가 할당되기 전까지의 이력만 추출하므로 Assignee, Status 항목은 제외한 7 개 필드에 대한 결과만을 분석하였다.

<표 2> 버그리포트의 주요 메타 필드

Field	Description
OS	문제가 발생한 운영체제
Hardware	문제가 발생한 하드웨어 및 OS 정보
Product	문제가 발생한 소프트웨어 상품
Version	문제가 발생한 소프트웨어 상품의 버전
Component	문제가 발생한 소프트웨어 내 컴포넌트
Severity	문제의 심각도
Priority	문제 해결 우선순위
Status	문제 해결 진행 상태
Assignee	문제를 해결하기위해 배정된 개발자

3.2 연구 질문

초기 버그리포트의 재할당의 실증적 분석을 위해 아래와 같은 3 가지 연구 질문을 설정한다. 아래 질문들은 X.Xia 의 연구 질문[1]을 확장하여 설정한 본 논문의 연구 질문이다..

RQ1. 초기 버그리포트의 어떤 필드가 높은 비율로 재할당되는가?
RQ2. 초기 버그리포트의 필드 별 재할당이 얼마나 많이 발생하는가?
RQ3. 재할당까지 얼마나 많은 시간이 소모되는가?

3.3 분석 결과

A. RQ1: 버그리포트들의 필드 별 재할당 비율

X.Xia 의 분석 결과에서는 필드 별 재할당 비율은 Assignee 의 재할당 비율이 약 51-75%로 가장 높고, 그 외 필드는 약 20% 미만의 비율을 보인다. 본 논문은 초기 버그리포트 분석을 위해 Assignee 변경과 개발자가 변경시킨 필드 외의 순수한 초기 재할당된 필드만 사용하여 분석을 수행한다. 해당 필드들은 12 개의 도메인 별로 구분하여 전체 버그리포트에 대한 필드 별 재할당 비율을 분석하였다.

표 3 은 초기 버그리포트들에 대한 필드 별 재할당 비율을 설명한다. 도메인 별로 재할당이 자주 일어난 필드를 분석한 결과, IoT 와 Modeling 도메인에서는 Priority, Version, Severity 필드, RT 와 Eclipse 에서는 Priority, Severity 필드, Birt 와 Science 는 Severity 필드의 재할당이 잦았다. ECD 와 Webtools 도메인은 Component, Product 필드, Tools 도메인은 Component, Version 필드, SOA 도메인에서는 Version 필드가 재할당 비율이 가장 높았다. 결과적으로 초기 리포트들의 재할당은 전체적으로 Severity 와 Priority 필드에서 가장 많이 발생하였다.

분석 결과를 통해 모든 도메인에서 필드의 변경 비율이 대체로 10% 미만인 것을 확인하였다. 그러나 IoT, SOA 도메인의 Version 필드, MyLyn 도메인의 Priority 와 Severity 필드, Webtools 도메인의 Component, Product, Version 필드는 10% 이상의 비율로 재할당 되었다. 특히 MyLyn 도메인의 Priority 필드는 약 23%로 다른 필드들에 비해 자주 재할당되었음을 알 수 있다. 즉 도메인 별로 자주 재할당 되는 필드에서 조금씩 차이를 보이는 것을 확인할 수 있었다.

<표 3> 버그리포트들의 필드 별 재할당 비율 분석 결과
(Underline: 가장 높은 비율, Bold: 10% 이상의 비율)

Domain	OS	Priority	Hardware	Version	Component	Severity	Product	Total
Birt	0.3%	1.8%	0.1%	1.7%	2.0%	**2.9%**	0.1%	23,056
ECD	0.7%	1.1%	0.3%	3.1%	**4.9%**	2.9%	4.3%	10,030
Eclipse	1.8%	3.9%	1.1%	2.4%	1.4%	**4.2%**	0.9%	185,148
IoT	1.2%	5.8%	1.0%	**10.9%**	4.2%	4.3%	0.0%	1,009
Modeling	3.4%	6.4%	3.2%	**8.3%**	5.4%	6.7%	1.8%	43,096
MyLyn	1.0%	**22.6%**	0.4%	1.8%	3.0%	**11.4%**	0.3%	8,422
RT	2.6%	**6.3%**	2.1%	4.9%	2.4%	6.0%	0.6%	36,212
Science	0.9%	0.9%	0.9%	0.0%	0.3%	**1.1%**	0.3%	355
SOA	0.9%	2.5%	0.3%	**10.8%**	3.0%	3.6%	0.2%	2,043
Technology	3.5%	3.4%	2.8%	4.5%	2.7%	**5.4%**	1.6%	37,453
Tools	1.9%	4.4%	1.1%	6.3%	7.3%	5.4%	2.0%	46,552
Webtools	1.5%	0.9%	1.5%	**10.3%**	**12.9%**	5.9%	**11.4%**	2,591
Total	**2.1%**	**4.6%**	**1.5%**	**4.0%**	**3.0%**	**4.9%**	**1.3%**	**395,967**

초기 버그 리포트의 재할당이 높은 비율로 나타난 필드는 Severity 와 Priority 로 전체 버그리포트 중 각각 4.9%, 4.6%로 비율로 재할당 되었다. 이는 리포터가 버그리포트를 작성할 때 Severity 와 Priority 를 가장 잘못 등록하는 것을 의미한다. 추가적으로 도메인 별로 재할당이 잦은 필드가 상이한 것을 확인하였다.

B. RQ2: 버그리포트의 재할당 횟수

표 4 는 버그리포트가 Assignee 가 할당되기 전까지 필드의 재할당 빈도를 도메인 별로 분석한 결과이다. 기존 연구에서는 문제 해결을 위해 버그리포트 별로 약 1~2 번의 재할당을 거친다는 결과를 보여준다[1]. 이에 근거로 재할당 횟수를 0,1,2,3 번이상으로 나누어 분석을 수행한다.

모든 도메인에서 약 80%정도의 리포트들은 재할당 없이 개발자가 배정 되었으며, 약 14%정도는 최소 1 번의 재할당을 거친다는 것을 알 수 있다. 그러나 IoT, Modeling, MyLyn 도메인에서는 약 30% 이상의 버그리포트가 최소 1 회 이상 재할당이 발생하였고, WebTools 도메인은 약 8% 이상의 버그리포트가 3 회 이상의 재할당이 발생한다. 즉 전체적으로는 재할당이 발생하지 않을 수 있으나, 도메인 별로 재할당 빈도가 차이가 나는 것을 확인할 수 있었다.

초기 버그리포트 중 약 20%는 최소 1 회 이상의 재할당이 발생하는 것을 확인하였다. 즉 약 1/4 개의 초기 버그리포트가 불완전하다는 것을 설명한다. 추가적으로 도메인 별로 재할당되는 빈도가 상이한 것을 확인하였다.

C. RQ3: 버그리포트 필드 별 재할당 기간

버그리포트가 등록되고 필드 별로 처음으로 재할당될 때까지 소요되는 평균 소요기간을 도메인 별로 분석한다. 분석을 통해 필드 필드가 처음 재할당 되기까지 걸린 평균 일수를 비교한다. 표 5 는 비교 결과를 설명한다.

<표 4> 버그리포트의 재할당 필드 수

Domain	0	1	2	3	total
Birt	89%	9%	1%	0%	23,056
ECD	89%	6%	4%	2%	10,030
Eclipse	83%	13%	3%	1%	185,148
IoT	72%	23%	4%	1%	1,009
MyLyn	67%	22%	6%	3%	43,096
RT	67%	22%	9%	1%	8,422
Modeling	80%	15%	4%	1%	36,212
Science	94%	4%	2%	0%	355
SOA	78%	18%	4%	1%	2,043
Technology	81%	12%	4%	2%	37,453
Tools	76%	17%	4%	2%	46,552
Webtools	78%	8%	4%	9%	2,591
Total	80%	14%	4%	1%	395,967

X.Xia 의 논문에서의 전체 재할당에 대한 결과는 평균적으로 Component, Product 가 가장 오랜 시간이 소요되고 Priority 는 가장 빨리 재할당된다고 분석했다[1]. 그러나 초기 버그리포트의 분석 결과 대부분의 도메인에서 Component, Version, Priority 필드의 순으로 오랜 시간이 소요되고, 오히려 오랜 시간이 걸릴 것이라 분석된 Product 는 그렇지 않은 것을 확인할 수 있었다. 또한 전체 버그리포트에서 Hardware 필드가 가장 빠르게 재할당 되는 것을 확인한 것을 통하여 기존 논문의 결과들이 초기 버그리포트에 대해서는 동일하지 않다는 것을 확인하였다.

도메인 별로 분석한 결과, Modeling, Technology, Tools 도메인에서 최대 소요기간은 모든 필드가 비슷했지만, 표 5 에서 확인할 수 있듯 평균적으로는 Component 가 다른 필드에 비해 재할당되는 데 소요된 시간이 가장 오래 걸림을 알 수 있다.

초기 버그리포트의 재할당까지 Component 필드가 가장 오랜 시간이 소요되고, Version, Priority 필드도 비교적 오랜 시간이 소모된다. 반대로 가장 적은 시간이 걸리는 Product 필드가 가장 빨리 재할당 된다.

<표 5> 도메인 별 필드 평균 재할당 기간 (일)
(Underline: 가장 늦은 재할당이 일어난 필드, Italic: 가장 빠른 재할당이 일어난 필드)

Domain	OS	Priority	Hardware	Version	Component	Severity	Product	Average
Birt	80	**229**	87	198	115	119	***51***	**126**
ECD	125	***120***	173	**269**	140	143	128	**157**
Eclipse	**1,652**	1,246	1,837	730	674	1,215	***427***	**1,112**
IoT	81	215	***72***	**362**	191	174	-	**156**
Modeling	3,328	6,828	***1,968***	7,496	**10,040**	6,623	8,520	**6,400**
MyLyn	147	**250**	***66***	205	194	245	70	**168**
RT	1,496	2,000	1,284	2,553	**2,913**	2,444	***565***	**1,894**
Science	-	20	***17***	-	**69**	34	**69**	**30**
SOA	***135***	830	140	1,063	**1,863**	444	1,535	**859**
Technology	4,015	***3,511***	3,675	5,419	**9,558**	4,288	5,813	**5,183**
Tools	3,042	3,189	***2,628***	3,506	**6,086**	2,952	3,083	**3,498**
Webtools	***56***	108	58	491	561	213	**615**	**300**
Total	**1,180**	**1,546**	***1,000***	**1,858**	**2,700**	**1,575**	**1,740**	

4. 결론 및 향후 연구

본 논문은 개발자가 할당되기 전에 잘못된 정보들이 변경되는 초기 재할당(Reassignment)의 경우에 초점을 맞춰 초기 버그리포트의 불완전성을 분석하였다. 분석을 위해 사용한 버그리포트는 395,967 개 였지만 프로젝트 별로 분석하기에는 각 프로젝트 별로 유의미한 값을 갖기에 부족한 개수를 가진 것이 대부분이었다. 이를 해결하기 위해 도메인 별로 분석을 수행하였다. 분석을 위해 3 가지 연구 질문을 설정하여, 필드 별 재할당 비율, 재할당 횟수, 재할당 기간을 분석하였다.

1) 필드 별 재할당 비율 분석 결과, 전체적으로 도메인 별로 차이는 보였지만 Priority 필드가 약 1~23%, Severity 필드가 약 1-11%, 평균적으로 각각 4.9%, 4.6% 의 재할당되는 것을 확인하여 다른 필드 대비 자주 재할당되는 것을 확인하였다.

2) 재할당 횟수를 분석한 결과 전체 버그리포트 중 약 20%가 초기에 불완전한 정보를 포함하고 있다는 것을 확인하였다.

3) 필드 별로 초기 재할당에 소모되는 시간을 분석한 결과 평균적으로 Component, Version, Hardware 필드 순으로 오랜 시간 소모되며, Product 필드는 가장 빠른 시간 내에 재할당 되는 것을 확인하였다.

즉 버그리포트의 초기 재할당 분석을 통해 새롭게 등록되는 버그리포트가 불완전할 수 있음을 파악할 수 있었다. 또한 모든 연구 질문들에서 도메인 별로 필드 별 재할당 특성이 상이한 것을 확인할 수 있었다.

분석 결과를 통해 확인된 초기 버그리포트의 불완전성은 버그와 관련된 문제를 해결에 소모되는 시간을 증가시킬 것이다. 그렇기 때문에 버그 리포트의 메타필드가 올바르게 제공되는 것이 중요하다. 즉 버그리포트들의 최초에 잘못 작성된 메타필드들을 바로 올바른 값으로 변경시킬 수 있다면 문제 해결에 소요되는 시간을 상당히 줄일 수 있을 것이다.

향후 본 연구팀은 논문의 분석 결과를 활용하여, 문제 해결 소요 시간을 단축시키기 위해 초기 메타필드를 강화시키기 위한 기술을 개발할 예정이다. 도메인 별로 파악된 재할당 패턴을 적용시킨다면 불완전한 초기 버그리포트들을 바로 수정할 수 있을 것이다.

Acknowledge

이 논문은 과학기술정보통신부 및 정보통신 기술진흥센터의 SW 중심대학 지원사업(No.2015-0-00914), 2016 년 정부(교육부)의 재원으로 한국연구재단-이공학 개인기초연구지원 사업(No. 2016R1D1A1B03934610), 2017 년도 정부재원(과학기술정보통신부 여대학(원)생 공학연구팀제 지원사업)으로 과학기술정보통신부, 한국연구재단과 한국여성과학기술인지원센터의 지원을 받아 연구되었습니다.

참고문헌

[1] Xia X, et al. “An empirical study of bug report field reassignment.” In Proceedings of the Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering, pp. 174-183, IEEE, 2014

[2] T. Zimmermann, et al. “What makes a good bug report?” IEEE Transactions on Software Engineering, Vol. 36, No. 5, pp. 618–643, 2010

[3] A. Lamkanfi and S. Demeyer. “Predicting reassignments of bug reports an exploratory investigation.” In Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering (CSMR), pp. 327–330, IEEE, 2013

[4] T. Zimmermann, et al. “Improving bug tracking systems.” In Proceedings of the 2009 31st International Conference on Software Engineering-Companion, pp. 247–250, IEEE, 2009.

[5] P. J. Guo, et al. “Not my bug! and other reasons for software bug report reassignments.” In Proceedings of the ACM 2011 conference on Computer supported cooperative work, pp. 395–404. ACM, 2011

[6] Xia, Xin, et al. “Automated bug report field reassignment and refinement prediction.” IEEE Transactions on Reliability, Vol. 65, No. 3, pp. 1094-1113, 2016