JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 3 : 683~688 Mar. 2015

CodeSonar를 이용한 지역 SW개발 업체의 결함 유형분석+

노정현 · 이종민 · 박유현*

Defect-Type Analysis of Regional SW Development Companies using CodeSonar

Jeong-Hyun Noh · Jong-Min Lee · Yoo-Hyun Park*

Department of Computer Software Engineering, Dong-Eui University, Busan 614-714, Korea

요 약

최근 소프트웨어의 결함을 탐지할 수 있는 다양한 종류의 동적분석 도구가 점점 더 많이 활용되고 있다. 하지만 업계에서 실제로 발생하는 결함에 대한 조사는 지금까지 거의 없었다. 본 논문에서는 C/C++, 자바 프로그램에서 결함을 찾아내는 툴인 CodeSonar가 찾아낸 결과를 분석하고자 한다. 분석결과 동남권 지역에서 가장 많이 발생하는 결함들을 다양한 방법으로 제시한다.

ABSTRACT

Recently, various static analysis tools for software defect detection are becoming widely used in practice. However, there is little public information of the most frequent defects in commercial areas until now. In this paper, we analyze the defects found by CodeSonar, a static analysis tool that finds defects in C/C++, Java programs. So we report the most frequent defects by various aspects in Dongnam area, Korea.

키워드 : 결함유형 분석, 정적분석, SW 테스팅, 소프트웨어공학

Key word : defect-type analysis, static analysis, SW testing, Software Engineering

접수일자 : 2015. 01. 20 심사완료일자 : 2015. 02. 10 게재확정일자 : 2015. 02. 23

+ 본 논문은 동남권SW품질역량센터 SW공학연구회 연구보고서 "최근 3년간 정적 코드 분석 결과를 통한 SW 품질 현황 분석과 개선 방안"과 동의대학교 석사학위 논문 "정적분석 도구를 이용한 지역 SW개발 업체의 결함 유형분석"의 내용을 요약하여 작성하였음

* **Corresponding Author** Yoo-Hyun Park(E-mail:yhpark@deu.ac.kr, Tel:+82-51-890-1737)
Department of Computer Software Engineering, Dong-Eui University, Busan 614-714, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.3.683 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 개발되고 있는 소프트웨어들은 과거에 비해 규모가 매우 커지고 있으며, 개발주기는 매우 빨라지고 있어 소프트웨어의 체계적인 품질 관리 기술의 필요성이 증가하고 있다. 고품질의 소프트웨어를 위한 방법으로는 다양한 인증을 획득, 최신 소프트웨어 개발 방법론이나 프로세스를 적용, 개발된 소프트웨어를 완벽한 테스트 프로세스와 테스팅을 통한 소프트웨어의 오류 제거가 대표적이다[1].

하지만, 중소기업의 경우 소프트웨어 개발을 적절하게 관리할 수 있는 체계적인 기법이나 절차 도입이 현실적으로 어려워 개발된 소프트웨어의 품질과 경쟁력 면에서 많은 문제점을 내포하고 있다. 이는 소프트웨어의 영속성 측면에서 유지보수를 어렵게 하는 요인이 되어 더욱더 경쟁력 확보가 어려워지고 있다.

2013년 소프트웨어공학센터에서 발간된 소프트웨어공학백서[2]를 참조하여 국내 소프트웨어 개발 기업의 2012년 소프트웨어 개발 수준을 살펴보면, 프로세스 수준점수, 소프트웨어품질인력 수준점수, 기술 수준점수, 소프트웨어공학 수준점수 모두 2011년도 대비 향상된 것으로 나타난다. 그러나 소프트웨어품질인력 수준점수는 간신히 60점을 넘어 다른 점수에 비하여 낮은데, 이것은 국내 소프트웨어 개발 업체들이 소프트웨어 개발 프로세스를 사내규정으로 정의하고 있지만, 해당 프로세스를 수행할 소프트웨어인력이 부족하거나 이를 위한 자동화된 시스템이 부족함을 의미한다[2]. 이는 소프트웨어 개발 업체의 개발자들이 소프트웨어품질 향상을 위한 최신 소프트웨어공학기술 적용을 위한 재교육 및 실무에 적용하려는 노력 부족과 함께 대학의 컴퓨터 관련 학과 교육과정에서 이와 관련된 최신 소프트웨어공학 기술 교육이 충분하지 않다는 점에서 그 원인을 찾아볼 수 있다.

본 논문은 2011년부터 2013년까지 3년간 동의대 부산IT융합부품연구소에서 수행한 정적 코드 분석 결과를 바탕으로 주로 발생하는 결함유형을 분석하였다.

이를 통하여 지역 중소기업에서 자주 발견되는 결함들을 추출하고 이들을 우선적으로 해결함으로써 전체적인 결함발생률을 감소시킬 수 있다는 것을 보인다.

Ⅱ. 관 련 연 구

2.1. 소프트웨어품질 관련 표준 동향

소프트웨어 품질은 크게 소프트웨어 제품(product) 품질과 소프트웨어 프로세스(process) 품질로 구분할 수 있다. 소프트웨어 제품 품질은 소프트웨어 고유의 목적과 고객의 요구사항을 만족하는지 여부로 알 수 있으며, ISO/IEC 9126 표준[3]는 소프트웨어 제품 품질을 위한 좋은 참고 모델이다. ISO/IEC 9126 소프트웨어 품질 모델은 현재 ISO/IEC 25010:2011[4]로 대체되어 있다. 프로세스 품질에 관한 표준은 CMMI, ISO 9001 등이 있다.

2.2. CWE(Common Weakness Enumeration)

CWE는 소프트웨어 보안 및 품질 강화를 위해 개발 시에 참고할 수 있도록 전 세계 소프트웨어 취약점을 표준화한 목록이다[5]. 미국국방성 산하 MITRE에서 관리하고 있으며 코드, 디자인 또는 시스템구조에서 발견되는 결함들에 대하여 발견하거나 토론할 수 있으며 다양한 개발언어를 지원하고 있다. 그리고 소프트웨어 및 응용프로그램 보안, 코드분석, CWE호환성이 보장되는 많은 테스팅도구들을 안내해주고 있으며, 결함유형별 예제와 해결방안도 제시하고 있다. 다른 여러 나라에서도 CWE를 참고하여 각 나라별 상황에 맞는 가이드를 제작하여 배포하고 있는데, 우리나라에서도 시큐어코딩_가이드[6] 및 소프트웨어 개발 보안 가이드[6]를 제작하여 배포하고 있다. 본 논문에서 분석 할 결함을 검출하기 위해 사용된 정적분석 도구인 Code Sonar도 CWE호환성을 보장 받아 검출 된 결함에 대한 정보를 쉽게 취득 할 수 있다.

2.3. 정적 테스팅

소프트웨어 품질을 향상시키기 위한 테스팅은 다양한 방법이 존재하며 이를 분류하는 기준도 여러 가지 존재 한다. 기본적으로 정적 테스팅과 동적 테스팅으로 분류한다. 정적 테스팅은 크게 도구를 사용하지 않는 검토(review) 절차와 도구를 사용하는 정적 분석(static analysis) 또는 정적 코드 분석(static code analysis)으로 분류할 수 있다.

정적 코드 분석은 동적 테스팅을 실행하기 전에 결함을 발견할 수 있어 수정 비용 측면에서 유리하다[7]. 프

로그램 복잡도 등의 품질 척도(metric)를 계산하여 의심스러운 코드와 설계를 미리 발견할 수 있으며, 동적 테스팅에서 발견하기 어려운 결함을 발견 할 수 있다. 또한 소프트웨어 모델에서의 의존성과 불일치성을 발견할 수 있으며, 코드와 설계의 유지보수성을 향상시킨다.

정적 분석 도구에는 간단한 오류를 발견하고 제거할 수 있는 CppCheck, PMD, FindBugs와 같은 오픈소스 도구와 배포전 소프트웨어의 품질을 최대한 향상 시킬 수 있도록 도와주는 CodeSonar, QAC/QAC++ 등의 상용툴이 있다. 본 논문에서 분석하는 정적 테스팅 결과는 다양한 언어와 환경에서 사용하기 편하고 여러 국제 표준을 지원하는 CodeSonar의 테스팅 결과를 이용하였다.

Ⅲ. CodeSonar를 이용한 지역 SW개발 업체의 결함 유형분석

본 장에서는 2011 ~ 2013년간 동남권 지역 소프트웨어 개발 업체에서 CodeSonar를 이용하여 검출한 결함유형을 분석 하고자 한다. CodeSonar에서 검출 할 수 있는 결함은 경고(Major), 권고(Minor), 치명(Critical) 3개의 레벨로 나눌 수 있다. 경고(Major) 결함은 실행시 시스템 동작에 많은 부작용을 미치기 때문에 반드시 제거해 주어야 하는 결함이고, 권고(minor) 결함은 실행시 부작용으로 인한 시스템 동작에 많은 영향을 미치지 않지만, 가능하면 제거하는 것이 좋은 결함이다. 또한, 치명적(critical) 결함은 시스템 동작에 직접적으로 영향을 미치므로 경고 결함과 함께 반드시 제거해 주어야 하는 결함이다.

3년간 수행한 테스팅의 결과 치명적 결함은 검출되지 않았고, 경고 경함과 권고 결함이 다수 나타났으나 본 논문에서는 경고 경함만을 분석하고자 한다. 향후에는 권고(Minor) 결함 및 결함밀도 자료를 추가하여 전체적인 결함 유형을 분석한다면 더욱 정확한 분석이 될 것으로 판단된다. 본 논문에서 기술하는 “결함”은 “경고결함”을 의미한다.

또한, 분석에 필요한 추가정보로는 연도별 테스팅 수행 업체, 업체 별 직원 수이며, 정확한 분석을 위하여 필요한 결함 밀도(결함개수/KLOC)에 대한 정보는 업체의 보안상의 이유로 보유하고 있지 않다. 결함 밀도(결함 개수/KLOC)에 대한 자료가 부족하기 때문에 이를 객관적으로 비교하기는 어렵다는 점을 감안하고 판단하여야 할 것이다.

3.1. 참여 업체 현황

지난 3년간 정적 분석을 의뢰한 업체 현황은 <표 1>, <표 2>과 같다.

표 1. 연도별 및 분야별 참여 업체 수 단위:개
Table. 1 Year number of participating companies and the number of participating companies by size

연 도	업체 수	규 모	업체 수
2011	9	소규모(10명 이하)	20
2012	15	중규모(50명 이하)	9
2013	12	대규모(50명 초과)	7
합계	36	합계	36

표 2. 분야별 참여 업체 수 단위:개
Table. 2 The number of participating companies by field

분 야	업체 수	분 야	업체 수
일반 소프트웨어	12	선박	6
영상	3	자동차	5
건축·토목	4	국방	5
의료	1		
합계	36		

3.2. 결함 유형 분석

3.2.1. 연도별 분석 결과

<표 3>은 2011년부터 2013년까지 수행된 정적 코드 분석 결과를 누적 빈도순으로 정리한 것이며 3년간 검출된 총 21개의 결함 중 거의 검출 되지 않는 “No Space For Null Terminator” 같은 결함 7가지는 제외하였다.

연도별로 정적 코드 분석에 참여한 업체 수와 업종이 달라서 이들 결함에 대한 비교 평가에 한계가 있을 수 있지만, 매년 비슷한 유형의 결함이 많이 발생하고 있다는 것을 알 수 있다. 가장 많이 발견된 결함은 결함 A로 21.27%를 차지하였으며, 메모리 관련 결함에 속하는 결함 H, J, K를 모두 포함하면 전체의 30.83%를 차지하고 있음을 알 수 있다. 이는 할당된 메모리를 잘못 사용하는 데서 기인함을 알 수 있다. 경고 결함에 대한 자세한 설명은 [8]에 자세히 기술하고 있다.

3.2.2. 업체 유형별 분석

<표 4>는 3년간 정적 코드 분석을 의뢰한 업체의 분야별 결함 개수를 누적한 것이다. <표 2>에서는 업체들을 7개로 분류하였지만, 이를 유사 분야 3개로 묶어 비교 분석 하였다. 기존 소프트웨어분류 중 의료 분야는 1개의 업체만이 분석에 참여 하여 업체별 분석에서 제외하였다.

표 3. 연도별 결함 비교 단위:건수(%)
Table. 3 Year-on-year comparison of the major defect

결함유형	2011	2012	2013	합계
Null Pointer Dereference(A)	140 (28.5)	165 (18.0)	93 (20.1)	398 (21.27)
Redundant Condition(B)	76 (15.4)	168 (18.3)	56 (12.1)	300 (16.03)
Uninitialized Variable(C)	47 (9.6)	151 (16.5)	32 (6.9)	230 (12.29)
Buffer Overrun(D)	53 (10.8)	97 (10.6)	66 (14.3)	216 (11.54)
Integer Overflow of Allocation Size(E)	36 (7.3)	108 (11.8)	54 (11.7)	198 (10.58)
Cast Alters Value(F)	53 (10.8)	44 (4.8)	14 (3.0)	111 (5.93)
Dangerous Function Cast(G)	64 (13.0)	39 (4.3)	7 (1.5)	110 (5.88)
Use After Free(H)	-	21 (2.3)	81 (17.5)	102 (5.45)
Format String(I)	-	44 (4.8)	25 (5.4)	69 (3.69)
Leak(J)	4 (0.8)	19 (2.1)	16 (3.5)	39 (2.08)
Free Null Pointer(K)	3 (0.6)	23 (2.5)	12 (2.6)	38 (2.03)
Unreasonable Size Argument(L)	6 (1.2)	11 (1.2)	6 (1.3)	23 (1.23)
Buffer Underrun(M)	6 (1.2)	13 (1.4)	-	19 (1.02)
Type Mismatch(N)	4 (0.8)	13 (1.4)	1 (0.2)	18 (0.96)
합계	492	916	463	1871

<표 4>에서와 같이 일반 분야 업체의 결함은 다른 분야 업체에 비하여 많은 결함이 발견되었다. 이에 대한 프로젝트 성격, 팀원의 소프트웨어 개발 성숙도 등 자료가 부족하여 그 원인을 분석할 수 없지만, 이미 업체 품질 규정을 준수하고 있는 자동차, 선박, 국방 등 특화된 다른 분야 업체와 비교하여 소프트웨어 품질 향상이 더욱 필요한 것으로 판단된다.

표 4. 분야별 결함 비교 단위:건수(%)
Table. 4 Major defect sectoral comparison

분야	업체수	2011	2012	2013	합계
일반 영상	15 (42.8)	214 (37.9)	284 (32.9)	147 (47.5)	645 (37.1)
건축 선박	10 (28.5)	285 (50.5)	346 (40.1)	97 (31.3)	728 (41.9)
자동차 국방	10 (28.5)	65 (11.5)	232 (26.9)	65 (20)	362 (20.8)
합계	35	564	862	309	1735

<그림 1> ~ <그림 3>은 세부적인 발생결함을 업체별로 나타낸 그래프이다. 테스팅을 수행한 업체수가 분야별로 다르기 때문에 분야별로 발생한 총 결함수와 각 세부결함의 비율로 정규화로 일반화 하였다.

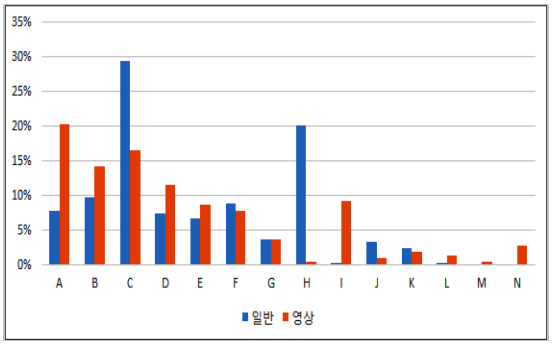

그림 1. 일반/영상 분야 결함유형 정규화
Fig. 1 General/Visual field defect type normalization

<그림 1>는 일반 소프트웨어분야 12개 업체와 영상분야 3개 업체의 결함유형을 나타내고 있다. <그림 1>에서 일반 소프트웨어 분야의 경우 결함A가 전체 결함 중 약 20%로, 영상 분야의 경우 결함 C가 전체의 약 34%로 가장 많이 발생하고 있다는 것을 알 수 있다.

일반 소프트웨어 분야의 경우는 <표 3>에서 정리한 것과 같이 결함 A, B, C가 전체 결함의 TOP3를 차지하고 있지만, 영상 분야의 경우, 결함 A~F가 많이 발생하며, 특히 결함 H가 많이 발생하였으나 이는 논문에서 분석한 바에 따르면 테스트 수행 업체수가 적어 보편적인 결과로 받아들이기 힘든 값으로 판단된다.

<그림 2>는 건축토목 분야 4개 업체 선박 분야 6개 업체의 결함개수를 정규화 하여 나타낸 그래프이다. 건축토목 및 선박 분야의 오류유형도 전반적으로 <표 3>의 순서와 유사하게 결함 A-F가 대부분의 결함을 차지하고 있으며, 선박 분야가 건축토목 분야보다 상대적으로 결함개수가 많이 나온 것을 알 수 있다.

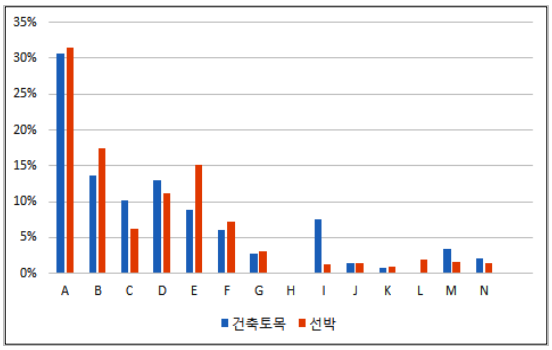

그림 2. 건축토목분야와 선박분야 결함유형 정규화
Fig. 2 Architecture and civil engineering field and vessels field defect type normalization

<그림 3>은 자동차분야 5개 업체와 2013년에만 테스팅한 국방 분야 5개 업체의 결함을 나타낸 그래프이다. 결함은 다른 분야와 마찬가지로 결함 A-F가 대체적으로 많았다.

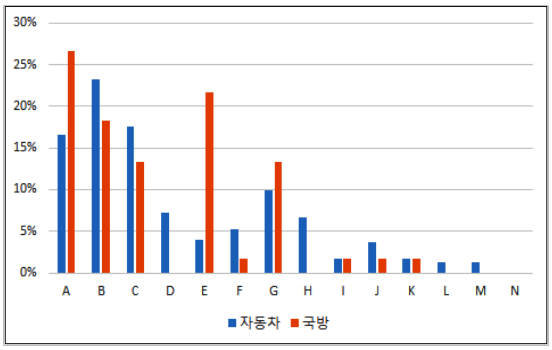

그림 3. 자동차분야와 국방 분야 결함 유형 정규화
Fig. 3 The automotive field and defense field defect type normalization

자동차 분야와 국방 분야의 결함이 다른 분야에 비해 적었는데, 그 이유는 아직 초창기이긴 하지만, 국방관련 품질 기준인 무기체계 소프트웨어 개발 및 관리 지침, 내장형 소프트웨어 획득 및 관리 실무 지침서, 무기체계 소프트웨어 코딩 규칙, 소프트웨어 신뢰성 시험평가 기준, 소프트웨어 기술 문서 작성 방법과 자동차 분야의 ISO26262 등 분야별 자체 소프트웨어 품질 기준을 가지고 이를 제도화 해 나가고 있기 때문인 것으로 판단된다.

3.2.3. 업체 규모별 분석

본 절에서는 참여한 업체의 규모를 소규모(10명 이하), 중규모(50명 이하), 대규모(50명 초과)로 분류하며 분석한다. <표 3>에서 정리한 바와 같이 조사된 소규모 업체는 20개, 중규모 업체는 9개, 대규모 업체는 7개이다. <그림 4>의 그래프는 소규모 20개 업체의 결함유형을 나타내고 있다. 결함 A-F가 많이 발생하며 이는 순서가 다를 뿐 <표 3>의 누적 결함과 같은 유형을 보이고 있다.

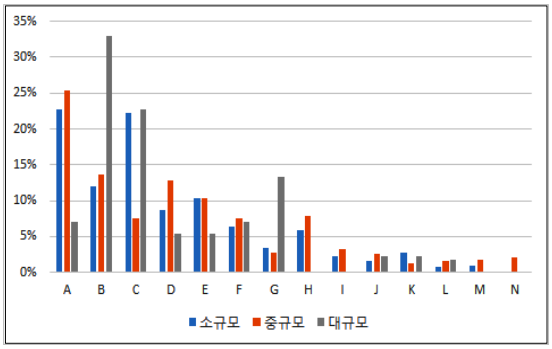

그림 4. 규모별 결함 유형 정규화
Fig. 4 Defects normalization by various company scale

Ⅳ. 결 론

본 논문에서는 2011년~2013년까지의 3년간 정적분석 도구인 CodeSonar로 검출된 결함에 대하여 여러 각도에서 분석하였다. 주로 경고 결함에 관하여 분석을 하였으며 총 21개 유형의 경고 결함이 나타났다. 이중 가장 빈번히 발생하는 상위 5개 유형의 경고 결함은 Null Pointer Dereference, Redundant Condition, Uninitialized Variable, Buffer Overrun, Integer Overflow of Allocation Size이며, 이들에 대해서만 완벽히 제거할 수 있더라도 전체 결함의 평균 70%이상을 줄일 수 있다는 것을 밝혀

내었다.

본 논문에서 밝혀진 주요 결함들을 대상으로 소프트웨어 개발자에게 홍보 및 결함 제거에 관한 교육을 수행한다면 좋은 성과를 이룰 것으로 기대된다.

감사의 글

이 논문은 2014학년도 동의대학교 교내연구비에 의해 연구되었음(과제번호 2014AA274)

REFERENCES

[1] Bo Kyung Park, Ha Eun Kwon, Hyun Seung Son, Young Soo Kim, Sang-Eun Lee, Young Chul Kim, "A Case Study on Improving SW Quality through Software Visualization", *Journal of KIISE*, Vol. 41, No. 11, pp. 935-942, 2014. 11

[2] Software Engineering White Book, NIPA, 2013.

[3] ISO/IEC 9126-1:2001 Software engineering – Product Quality – Part 1:Quality Model.

[4] ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.

[5] CWE - Common Weakness Enumeration, Available: http://cwe.mitre.org

[6] Software Development Secure Coding Guide, Available: http://www.mospa.go.kr/frt/bbs/type001/commonSelectBoardArticle.do?bbsId=BBSMSTR_000000000045&nttId=34430

[7] Certified Tester Foundation Level Syllabus, International SoftwareTesting Qualification Board (ISTQB), 2011.

[8] Jeong-Hyun Noh, "Defect-Type Analysis of Regional SW Development Companies using a Static Analysis Tool", Mater's thesis, DongEui Unversity Graduate School, 2015.2

노정현(Jeong-Hyun Noh)

2013년 동의대학교 컴퓨터소프트웨어공학과 공학사
2015년 동의대학교 컴퓨터소프트웨어공학과 공학석사
※관심분야 : 소프트웨어 테스팅, 클라우드 시스템, 앱 프로그래밍, 소프트웨어공학

이종민(Jong-Min Lee)

1992년 2월 경북대학교 컴퓨터공학과 공학사
1994년 2월 KAIST 전산학과 공학석사
2000년 8월 KAIST 전자전산학과 공학박사
1997년 9월 ~ 2002년 2월 삼성전자 무선사업부 책임연구원
2005년 2월 ~ 2006년 2월 Research associate, University of California at Santa Cruz
2012년 1월 ~ 2013년 2월 Visiting Scholar, The University of Alabama
2002년 3월 ~ 현재 동의대학교 컴퓨터소프트웨어공학과 교수
※관심분야 : 네트워크 프로토콜, 병렬컴퓨팅, 애드혹 라우팅

박유현(Yoo-Hyun Park)

1996, 1998, 2008년 부산대학교 전자계산학과 이학사, 이학석사, 이학박사
2000년 한국국방연구원(KIDA) 연구원
2001년 ~ 2009년 한국전자통신연구원(ETRI) 선임연구원
2012년 ~ 2014년 동의대학교 부산IT융합부품연구소 부소장
2009년 ~ 현재 동의대학교 컴퓨터소프트웨어공학과 부교수
※관심분야 : 클라우드 시스템, 빅데이터, 인터넷시스템, 소프트웨어 품질, IT 융합 서비스