가치 및 위험 기반의 SW 테스트 우선순위 방법론

이정우*, 이동현**, 김능회**, 인호,**
*고려대학교 컴퓨터정보통신대학원 소프트웨어공학과전자계산학과
**고려대학교 정보통신대학 컴퓨터 · 통신공학부
e-mail :iammario@hanafos.com, tellmehenry@gmail.com, nunghoi@korea.ac.kr, hoh_in@korea.ac.kr

Value & Risk-based SW Test Prioritization Method

Jung-Woo Lee*, Dong-Hyun Lee**, Neung-Hoe,Kim**, Hoh Peter In**
*Dept. of Software Engineering, Graduate School of Computer Information & Communication, Korea University
** Dept. of Computer & Communication, College of Information & Communication, Korea University

요 약

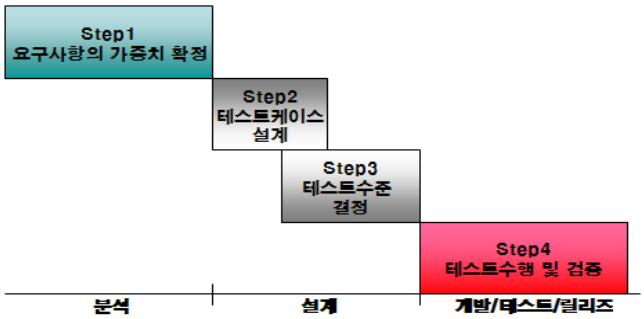
기존에 위험기반 테스팅(Risk-based Testing) 방법은 널리 알려져 있다. 또한 가치기반의 소프트웨어 공학(Value-based SW engineering)이 고전적인 방법론과 더불어 대두되고 있다. 요구공학을 기반으로 하면서 소프트웨어의 생산절차를 가치(Value)를 기반으로 구성하고 있다. 본 논문에서는 위험기반 테스팅 기법을 Value-based Testing 과 접목하여, 테스트케이스에 우선순위를 부여하고, 우선순위별로 적절한 테스트 기법을 도출, 적용하여 그 효과를 검증하고자 한다. 검증의 방법은 적용 전/후의 품질비용을 기준으로 한다.

1. 서론

대부분의 SI 프로젝트에서 어플리케이션 테스트의 중요성은 강조되고 있다. 많은 연구에서 전체 개발비용의 30~50% 정도의 비용을 테스트에 지불하고 있다는 결과를 보여주고 있다[1]. 그러나 국내 프로젝트의 경우는 현실적으로 한정된 자원으로 많은 양의 테스트를 요구하고 있다. 이러한 이유로 적용하게 된 것이 바로 위험기반 테스팅 기법이다. 위험도가 높은 테스트케이스를 도출하여 테스트를 집중하고 ROI 를 높이는 기법이다.

본 논문에서는 위험기반 테스팅과 함께 최근 대두되고 있는 가치기반 소프트웨어 공학기법을 적용하여 일반적으로 경험기반(Exploratory) 테스트를 수행한 경우와 품질비용을 비교하여 증명하고자 한다.

2. Value & Risk-based SW Testing Process



(그림 1) Value & Risk-based testing process

(그림 1)은 본 논문에서 제시하는 아이디어를 도식화 한 것이다. 요구사항 별로 가치와 위험을 부여하여 우선순위를 선정하고, 테스트케이스를 설계하여 그 우선순위에 따라 테스트수준을 결정한다. 그리고 테스트를 수행한 결과를 검증하는 절차로 구성을 했다.

본 방법론에서는 테스트케이스를 구성함에 있어서 가치와 위험을 기준으로 하여 우선순위를 부여하고 그에 따라 적용 테스트 기법을 차별화 하는 것을 기본으로 한다. 가치 및 위험의 가중치는 고객이 제시한 요구사항을 정리하는 분석 시점에 부여를 한다. 요구사항 별로 부여가 되고, 실제로 테스트케이스를 차별화 하는 것은 테스트케이스를 개발하는 단계인 설계단계이다.

가치라는 것은 고객이 제시한 요구사항이 구현되었을 때 그 업무를 수행함으로써 기대할 수 있는 미래에 창출가능한 것을 말한다. 기존에 없던 프로세스는 신규로 창출 가능한 가치가 될 것이고, 기존에 있던 것이 개선된 경우에는 추가적으로 발생하는 가치만큼이 여기서 말하는 가치가 될 수 있을 것이다. 이는 순수하게 고객이 판단할 수 있는 부분이다. 설계된 테스트케이스에 각각의 가치가 기록이 될 것이다.

위험 역시 발생가능성과 그 영향력을 계산하여 위험 노출도를 산출한다. 위험기반의 테스팅은 기존에 널리 알려진 방법을 그대로 적용한다.

이렇게 산출된 가치와 위험을 metric 으로 표현을 한 이후 차별화 된 테스트 기법을 적용하도록 한다.

차별화된 테스트 기법은 테스트 완료조건, 테스트 인력, 자동화된 회귀 테스트(regression test) 적용여부 등으로 각각의 완료기준 및 품질수준을 정하여 테스트를 적용한다.

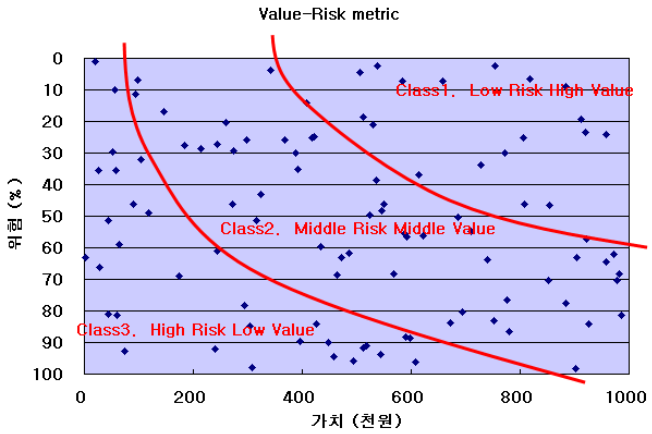
3. Value & Risk-based SW Testing Process 적용결과

3.1 적용절차

본 논문에서 제시한 아이디어를 적용한 결과는 인건비 기준으로 품질비용(Cost of Quality)을 측정하여 각 품질비용 영역 (예방, 평가, 내부실패, 외부실패)으로 구분하고 기존의 방법대로 테스트한 프로젝트의 평균치와 비교하여 Value & Risk based SW Testing 의 성과를 증명해 보이고자 한다. 적용절차는 다음과 같다.

Step1 (요구사항의 가중치 확정): 분석단계에 고객의 요구사항을 정리한 목록에 가치 및 위험 필드를 추가하여, 가치 부분은 고객이, 위험 부분은 개발사업자가 작성을 한다.

Step2 (테스트케이스 설계): 설계 단계에 테스트 케이스를 작성할 때 가치와 위험을 최종 확정하여 기록을 하고, 가치, 위험을 축으로 하는 metric을 작성한다. 테스트 우선순위를 정하기 위한 작업이다. 아래 (그림 2)에서 도출된 테스트 케이스를 대상으로 가치와 위험을 축으로 한 metric 을 만든다.



(그림 2) Value-Risk metric

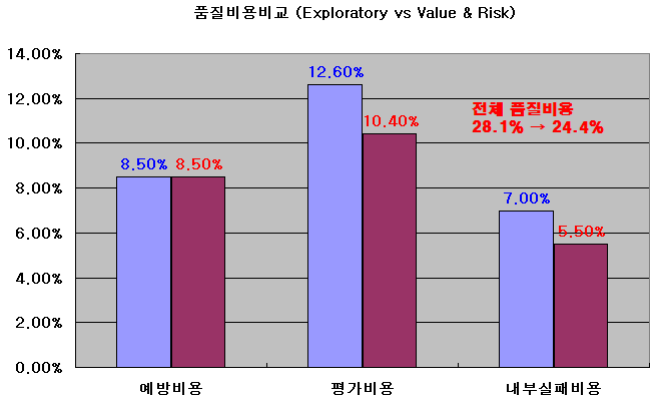
Step3 (테스트 수준 결정): 정해진 metric 을 위험과 가치를 기준으로 세 개의 class 로 분류하여, 각 class 별로 적용할 테스트 방법을 차별화 한다. 해당 class 에 대해서는 동일한 수준의 테스트를 수행하도록 한다. [3]

<표 1> Class 별 테스트 수준

Class	설계기법	완료조건	테스트인력배치
Class1 Low Risk High Value	usecase testing usecase integration testing exploratory testing pair-wise testing exreme testing	TC 100% 수행 및 95% 성공 Regression Test (5 times)	test manager 통제 외부 전문테스트 투입
Class2 Middle Risk Middle Value	usecase testing state transition testing (1-swithch) boundary value analysis equivalance class partitioning	TC 100% 수행 및 90% 성공 Regression Test (3 times)	test manager 통제 외부 테스트 인력 투입
Class3 High Risk Low Value	usecase testing state transition testing (01-swithch) equivalance class partitioning	TC 100% 수행 및 85% 성공 No regression Test	test manager 통제 내부 개발인력이 테스트 수행

Step4 (테스트 수행 및 검증): 정해진 테스트를 절차에 맞게 수행한 이후 프로젝트의 종료 이후에 검증을 위한 품질비용을 측정한다. 실패비용을 산정하기 위해 Release 이후에 측정하는 것으로 한다.

3.2 적용결과



(그림 3). 품질비용 비교

(그림 3)은 '03~'09 년까지 측정한 Exploratory 기법을 적용하여 테스트한 315 개 프로젝트의 품질비용 평균과 본 논문에서 제시한 방법론을 적용한 프로젝트의 품질비용을 비교한 것이다. 적용 결과에서 볼 수 있듯이 각 class 별로 차별화된 테스트 기법을 적용한 후 전체 품질비용이 28.1%에서 24.4%로 줄어들었다. 금액으로 환산한다면, 인건비 100 억의 프로젝트라면, 약 3,700 만원에 해당하는 금액이다.

고객의 가치는 실패비용에 영향을 줄 수 있는 부분이다. 요구사항 중 가치가 높으면서 위험도가 낮은 것에 테스트를 집중하여 테스트 ROI 를 높이고, 전체적인 품질비용을 저감함으로써 개발사업자의 원가 절감 및 고객의 만족도 제고에 공헌할 수 있는 모델이다.

향후 class 를 세분화하는 기법과 class 별 테스트 수준을 추가적으로 구분할 수 있는 효율적인 모델에 대한 연구가 필요하다.

Acknowledgement
"본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2009-(C1090-0903-0004))

참고문헌

[1] Stefan Biffl, Barry Boehm. "Value-Based Software Engineering"Linda H. Rosenberg, Ph. D "Risk-based object oriented testing"

[2] Linda H. Rosenberg, Ph. D "Risk-based object oriented testing"

[3] Rex Black. "Managing the Testing Process"