

# ISO/IEC 14971의 리스크 평가 기반 테스트케이스 가중치 산정 기법에 관한 연구

최민지, 송기원\*

건양대학교, \*건양대학교

22806507@konyang.ac.kr, \*kiwonsong@konyang.ac.kr

## A Study on the Risk Evaluation-Based Testcase Weight Calculation Techniques of ISO/IEC 14971

Choi Min-Ji, Song Ki-Won\*

Konyang Univ., \*Konyang Univ.

### 요약

본 논문은 소프트웨어의 품질을 보장하기 위하여 소프트웨어 테스트를 통해 결함을 찾아 개선하기 위하여 테스트 기법에 대하여 연구한다. ISO/IEC 14971을 기반으로 위험 발생 가능성과 위험 영향도의 등급을 5등급으로 세분화하여 위험 노출도 값을 계산하고 위험 노출도를 확률/영향 매트릭스로 나타내어 3가지 영역으로 나눈 뒤 우선순위를 결정하는 방법에 대하여 제안한다.

### I. 서론

소프트웨어 결함이란 응용 프로그램의 예상 동작을 실제 동작과 일치시키지 않음으로써 응용 프로그램의 정상적인 흐름을 제한하는 응용 프로그램의 결함 또는 오류를 말한다[1]. 소프트웨어의 결함은 시스템의 오작동까지 불러올 수 있다. 이때, 오작동이란 결함의 실행으로 예상과 다른 결과가 출력되는 것으로 오작동은 결함에 의해 발생하는 것이 맞지만 결함이 있다고 해서 오작동이 반드시 발생하는 것은 아니다. 소프트웨어의 결함을 찾기 위해서는 소프트웨어의 품질 관리가 필수적이다. 소프트웨어 품질이란 소프트웨어가 요구사항을 만족하며 표준에 준수하는 소프트웨어의 목시적인 필요 및 기능 또는 특성을 만족시키는 능력과 관련된 소프트웨어의 특징 및 특성 전체를 말한다[2]. 이는 제품이 고객에게 제공되었을 때 고객의 요구사항을 만족하고 이용에 불편함이 없는 정도라고 할 수 있다. 품질 관리를 할 때는 국제표준화기구(ISO; International Organization for Standardization)에서 제정한 품질 표준인 ISO/IEC 20510:2011의 품질 모델에 정의된 제품 품질 모델을 따른다[3]. 노출되어 있지 않은 숨어있는 결함을 찾기 위해서는 소프트웨어 테스트를 필수적으로 하게 되는데 이는 소프트웨어를 작동시키는 일련의 절차와 행위를 말한다[4]. 프로그램을 실행하여 오류 발견을 목적으로 품질을 평가하는 과정이며 개발된 소프트웨어의 문제와 결함을 식별하고 품질을 개선하기 위한 일련의 활동을 의미한다. 테스트를 진행하기 위해서는 기법을 선택하여야 하는데 테스트 기법을 선정할 때는 리스크 레벨이나 유형, 테스트 레벨, 개발 생명주기 등의 다양한 고려사항이 존재한다[5]. 따라서, 본 논문에서는 ISO/IEC 14971의 리스크 평가를 기반으로 하는 테스트케이스 가중치 산정 기법에 관하여 연구한다.

### II. 본론

그림 1은 위험 분석 수행 절차를 나타낸 것이다. 위험 요소를 식별하고 위험 노출도를 산정한 뒤 위험 평가를 진행한다.

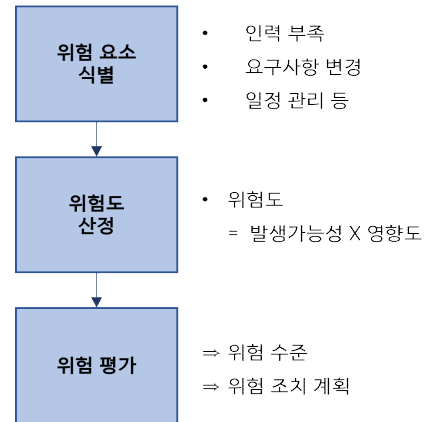


그림 1. 위험 분석 수행 절차

본 논문에서는 ISO/IEC 14971의 리스크 평가를 기반으로 하여 위험 발생 가능성과 위험 영향도로 구분한 뒤 5등급으로 세분화하였다. 위험 발생 가능성은 위험이 발생할 수 있는 가능 정도를 의미하며 위험 영향도는 위험이 발생하면 프로젝트의 성공에 영향을 주는 정도를 평가하는 것이다.

표 1. 위험 발생 가능성 등급

등급	설명
1	발생 가능성이 전혀 없음
2	30% 미만 발생할 가능성이 있음
3	30% 미만 발생
4	80% 미만 발생
5	80% 이상 발생

표 2. 위험 영향도 등급

등급	설명
1	시스템 손상 없음
2	5% 미만 시스템 손상
3	5% 미만 시스템 손상/다른 방안 존재
4	20% 미만 시스템 손상
5	20% 이상 시스템 손상

표 1은 위험 발생 가능성 등급을 나타낸 것이고 표 2는 위험 영향도 등급을 나타낸 것이다. 이 두 가지 항목을 활용하여 위험 노출도를 계산한다. 위험 노출도는 다음 계산식을 통해 도출한다.

$$\text{위험 노출도} = \text{위험 발생 가능성} \times \text{위험 영향도} \quad \text{수식 1}$$

위험 노출도는 표 3과 같이 매트릭스를 통해 도출할 수 있다. 이 매트릭스는 확률/영향 매트릭스로 위험 허용 기준을 나타내는데 위험을 다음과 같이 3가지 영역으로 구분한다.

표 3. 위험 노출도 매트릭스

영향도 \ 발생 가능성	손상 없음(1)	5% 미만 손상(2)	5% 미만 손상/방안 있음(3)	20% 미만 손상(4)	20% 이상 손상(5)
발생가능성 전혀 없음(1)	1	2	3	4	5
30% 미만 발생가능성 있음(2)	2	4	6	8	10
30% 미만 발생(3)	3	6	9	12	15
80% 미만 발생(4)	4	8	12	16	20
80% 이상 발생(5)	5	10	15	20	25

- ① Red zone : 허용할 수 없는 영역[위험 노출도 15 이상]
- ② Yellow zone : 합리적으로 실행 가능한 영역[위험 노출도 5-12 사이]
- ③ Green zone : 허용 가능한 영역[위험 노출도 4 이하]

위험 노출도는 준비된 각각의 테스트케이스 식별번호에 따라 위험 발생 가능성과 위험 영향도의 곱으로 구성한다. 표 4는 ‘메타버스 환경의 언어 교육’ 서비스의 시나리오별로 추출한 테스트케이스에 각각의 위험 발생 가능성과 위험 영향도, 위험 노출도를 산정한 것을 일부 나타낸 것이다. “시작 버튼을 눌렀을 때 시스템이 정상적으로 작동하는지 확인”의 항목을 보면 일부 기능이나 환경에 따라서 장애 발생 가능성이 30% 미만 있으므로 위험 발생 가능성을 3점 부여하였고, 동작하지 않을 시 시스템의 손상을 가져오기 때문에 위험 영향도를 5점을 부여하였다. 따라서 위험 노출도는 15점을 부여받게 된다. “학생의 아바타 기능이 on/off 되는지 확인”의 항목에서는 일부 기능이나 환경에서 30% 미만의 장애 발생 가능성이 있으므로 3점을 부여하였고 위험 영향도는 20% 미만의 시스템 손상을 야기할 수 있으므로 4점을 부여하였다. 따라서 위험 노출도는 12점을 부여받게 된다.

표 4. 위험 노출도 산정 일부

번호	테스트케이스	위험 발생 가능성	위험 영향도	위험 노출도
1	시작 버튼을 눌렀을 때 시스템이 정상적으로 작동하는지 확인	3	5	15
2	학생의 카메라 기능이 on/off 되는지 확인	3	1	3
3	학생의 아바타 기능이 on/off 되는지 확인	3	4	12
4	학생의 목소리 스타일을 변경할 수 있는지 확인	3	2	6

이러한 위험 노출도 값을 얼마나 강도 높게 해당 요구사항을 테스트해야 하는지 나타내며 위험 노출도를 이용하여 테스트의 우선순위를 결정할 수 있다. 위험 노출도가 높을수록 소프트웨어가 위험을 가할 확률이 높은 것이기 때문에 우선순위가 높은 것으로 분류한다.

### III. 결론

본 논문에서는 소프트웨어의 품질 개선을 최종 목적으로 결함을 찾기 위해 소프트웨어 테스트를 진행할 때의 테스트 기법에 관한 연구를 진행하였다. ISO/IEC 14971을 기반으로 위험 발생 가능성과 위험 영향도 등급을 5등급으로 세분화하여 위험 노출도 값을 산정하는 방법과 위험 노출도를 위해 확률/영향 매트릭스로 나타내어 3가지 영역으로 구분한 뒤 우선순위를 결정하는 방법에 대하여 제안한다. 사례로 ‘메타버스 환경의 언어교육’ 서비스의 시나리오별 테스트케이스에 적용했으며 위험 노출도를 산정한 일부를 나타내었다. 이를 통해 향후 스모크 테스트에 적용하여 테스터들에게 안정성 있고 효율성을 보장하며 테스트를 진행할 수 있도록 도움을 줄 수 있다.

### ACKNOWLEDGMENT

이 논문은 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 ‘바이오융복합기술 전문인력 양성사업’의 지원을 받아 수행된 연구임(No. P0017805).

### 참 고 문 헌

- [1] Radjenović, D., Heričko, M., Torkar, R., & Živković, A. (2013). Software fault prediction metrics: A systematic literature review. *Information and software technology*, 55(8), 1397-1418.
- [2] Korea Creative Content Agency. Understanding Software Quality. [www.kocca.kr](http://www.kocca.kr)
- [3] Telecommunications Technology Association. Software Test Specialist (csts) Guide.
- [4] Span, J. (1999). Software testing. *Dependable Embedded Systems*, 5(2006), 1.
- [5] Han Sang-hyuk, Jeong Jeong-soo, Jin Sung-il, & Kim Young-guk. (2010). Software test case dynamic management technique with weights. *Journal of the Society of Information Processing*, Vol. 17-D, (6), 423-430.