제20권 제2호 2014년 2월 ISSN 1229-7712

정보과학회논문지 : 컴퓨팅의 실제 및 레터

Journal of KIISE : Computing Practices and Letters

VOLUME 20, NUMBER 2, FEBRUARY 2014

레터 논문

한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

기존 UCP 기반 유스케이스 복잡도와 위험 영향도 상관관계 분석

(An Analysis on Correlations Between Use Case Complexity based on Existing UCP and Risk Impact)

김 보 연 † 손 현 승 †
(Bo Yeon Kim) (Hyun Seung Son)

김 영 철 ††
(R. Young Chul Kim)

요 약 기존의 리스크 기반 테스팅은 프로젝트, 프로세스, 제품 리스크등 전체에 초점이 맞추어져 있어 소프트웨어 개발에 최적화된 리스크 기반 테스팅 방법이 필요하다. 기존 논문에서 소프트웨어 개발에 리스크 기반 테스팅를 적용하고자 리스크 결정 매트릭스를 제안하였다[1]. 리스크 결정 매트릭스는 리스크 요구사항과 유스 케이스 간의 상관관계 점수와 리스크 점수를 이용하여 유스 케이스 기반의 리스크 영향도를 측정하고 이를 우선순위화 한다. 본 논문에서는 기존의 유스 케이스 점수(UCP)와 제안한 리스크 영향도와의 상관관계를 보이고자 한다. 상관관계를 통하여 유스 케이스 점수가 높을수록 리스크 영향도 값이 높다는 것을 알 수 있다. 이를 통하여 유스 케이스 단위의 리스크 영향도 우선순위화 방법이 소프트웨어 개발의 리스크 기반 테스팅에 적합성을 보인다. 제안한 이론의 적합성을 검증하기 위하여, 본 논문의 리스크 기반 테스팅 기법을 자동차 물품관리 시스템에 적용하여 유스 케이스 점수와 리스크 영향도의 상관관계성을 유추 확인하였다.

키워드: 리스크 기반 테스팅, 유스 케이스 점수, 리스크 영향도, 리스크 결정 매트릭스, 우선순위

Abstract As the existing risk-based tests are focused on only risks of project, process and product, so an optimized method for software development is necessary. In former paper, we proposed the Risk Decision Matrix for applying the risk-based testing to software development. The Risk Decision Matrix evaluate the Risk Impact(RI) based on the Correlation between Risk Requirement(RR) and Use Case(UC), and Risk Point(RP). Then it prioritize the values of Risk Impact. In this paper, we show the correlationship between the existing Use Case Point(UCP) and proposed Risk Impact. Through analysis of the correlationship, we find that the higher the UCP is, the higher the RI is. According to this result, it is shown that the proposed method of RI prioritization for use case is proper to the risk based testing for software development. For the example to verify the proposed theory, we apply it to the Car Appliances Management System and confirm our inference.

Keywords: risk based testing, use case point (UCP), risk impact, risk decision matrix, priority

1. 서 론

현재 대다수의 소프트웨어 팀들은 대응 리스크 전략에 의존하고 있다. 그래서 고객이 원하는 소프트웨어의 기대치가 높아짐에 따라 잠재적인 리스크들에 대한 관리가 필요하다. 이에 따라 리스크 기반의 테스팅에 대한 관심이 높아지고, 계속적으로 많은 연구되고 있다. 리스크 기반 테스팅은 리스크 노출 값을 측정하여 리스크의 실패 가능성을 줄이고자 한다. 또한 리스크 발생 가능성이 높은 부분에 대하여 선택과 집중을 통하여 테스팅을 수행할 수 있으며, 시간 및 자원의 한계로 인해 필수적으로 테스트 할 부분을 고려하여 테스팅을 실시 할 수 있다. 이는 리스크를 완벽하게 제거할 수 없기 때문에 완벽한 테스팅이 아닌 최적의 테스트 방안을 찾고자 함이다[2-4].

기존의 리스크 기반 테스팅은 조직, 프로젝트, 프로세스, 제품 리스크등 전체에 초점이 맞추어져 있다. 리스크 기반 테스팅의 리스크 식별 및 분석 단계에서 리스크 결정 매트릭스를 적용하고자 한다[5,6]. 리스크 결정 매트릭스는 경험적 리스크 분석방법을 기초로 한다. 제안된 리스크 결정 매트릭스는 리스크 요구사항과 유스 케이스 간의 상관관계 점수와 리스크 점수를 이용하여 유스 케이스 기반의 리스크 영향도를 측정하고 이를 우

· 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업원천기술개발사업[10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발]과 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2013R1A1A2011601)
· 이 논문은 2013 한국컴퓨터종합학술대회에서 '유스 케이스 복잡도와 리스크 영향도 상관관계성 연구'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 홍익대학교 소프트웨어공학연구실
yeon@selab.hongik.ac.kr
son@selab.hongik.ac.kr
†† 정 회 원 : 홍익대학교 컴퓨터정보통신공학과 교수
bob@hongik.ac.kr
(Corresponding author임)

논문접수 : 2013년 8월 16일
심사완료 : 2013년 11월 13일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제20권 제2호(2014.2)

선순위화 한다. 이를 통하여 얻은 리스크 영향도를 유스 케이스 복잡도와 비교분석 한다. 본 논문에서는 유스 케이스 점수 계산방법을 통해 리스크 영향도와 유스 케이스 점수의 유사성을 찾아 리스크 결정 매트릭스의 리스크 영향도가 소프트웨어 개발의 리스크 기반 테스팅에 적합 또는 유사성을 확인한다. 기존에 연구되어진 리스크 관련 자료들을 종합하여 요구사항으로부터 리스크 식별 및 분석을 위한 프로세스 메커니즘을 제안하여 테스트 단계로 확장하고자 한다.

본 논문은 총 5장으로 구성되어 있다. 2장에서는 요구사항 분석을 통한 리스크 추출 및 우선순위화에 대하여 기술하고, 3장에서는 유스 케이스 점수와 리스크 영향도 상관관계를 분석하고, 4장에서는 결론 및 향후 연구 과제에 대해서 기술한다.

2. 관련 연구

리스크 기반의 테스팅이 중요한 이유는 소프트웨어 개발이 본질적 위험에 대하여 불확실하기 때문이다. 정해진 시간 안에 소프트웨어를 철저하게 테스트 할 수 없기 때문에, 선택적으로 테스트를 해야 한다. 테스트의 효율성을 극대화하기 위해 위험이 높은 부분에 집중하여 테스팅 함으로써 제한된 시간과 자원으로 좋은 품질을 생산해 낼 수 있다[2-4,7].

2.1 리스크 요구사항 추출

소프트웨어 개발에서 중요한 요소 중 하나는 정확한 요구사항을 추출하는 것이다. 요구사항은 문제를 해결하거나 목적을 달성하기 위하여 사용자 또는 고객이 필요로 하는 조건 또는 기능을 말하며, 사용자 및 개발자의 관점을 모두 포함한다. 소프트웨어의 규모가 커지면서 요구공학의 중요성은 나날이 커지고 있다. 또한, 요구사항을 정확히 반영함에 따라 소프트웨어의 품질차이도 매우 크다. 본 논문에서는 현재 시스템 상에서 발생할 수 있는 리스크에 대한 리스트를 리스크 요구사항이라고 정의한다. 리스크 요구사항은 기존의 Corkburn이 제시한 목표지향의 유스 케이스 방법을 확장한 목표지향 요구사항 기반 테스팅(GORe: Goal-Oriented Requirement Based Testing)을 적용하여 식별한다. 리스크 요구사항을 식별하기 위하여 먼저, 고객의 요구사항을 정의한 후 사용자별 목표를 도출하고 요구사항 정의 문서를 작성한다. 목표는 기능적과 비기능적 요소를 포함하며, Cockborn의 목표지향 유스 케이스 방법을 도입하여 도출된 유스 케이스 다이어그램을 통해 목표를 도출하여 잠재되어있는 리스크들을 식별해낸다. 그 다음, 유스 케이스 명세서를 작성한다. 유스 케이스 별로 식별된 리스크와 유스 케이스 명세서를 통하여 각각의 유스 케이스별로 리스크 요구사항을 식별한다[8].

2.2 유스 케이스 기반의 리스크 결정 매트릭스

소프트웨어 개발 전단계에 리스크 기반의 테스팅을 실시함으로써, 모든 리스크들을 다 피할 수 있는 것은 아니지만 리스크의 통제가 가능하고 효과적으로 대처할 수 있다. 기존의 리스크 기반의 테스팅과는 달리 소프트웨어 개발에 최적화된 리스크 기반의 테스팅을 위해 리스크 결정 매트릭스를 제안하였다. 이는 리스크 기반 테스팅의 리스크 분석, 측정단계에 해당한다. 제안한 리스크 영향도 측정방법은 가치혁신 요구공학(VIRE:Value-Innovative Requirments)과 Goal지향 요구사항 기반 테스팅(GORe: Goal-Oriented Requirement Based Testing)을 리스크 결정 매트릭스로 확장하고 이를 이용하여 리스크 영향도를 우선순위화하였다[1].

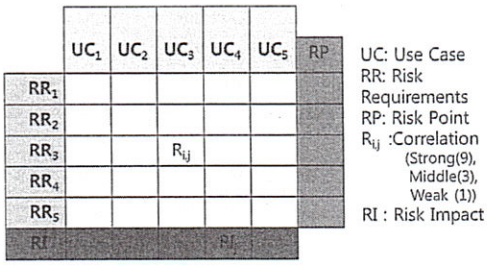

그림 1 리스크 결정 매트릭스[1]
Fig. 1 Risk Decision Matrix

식별된 리스크 요구사항을 통하여 각 리스크의 유스 케이스와의 관련정도를 분석하고, 리스크의 우선순위(Priority)와 리스크 포인트(Risk Point, RP)를 측정한다. 리스크 포인트는 위험 없음, 위험정도가 낮음, 위험정도가 보통, 위험정도가 높음, 위험정도가 매우 높음으로 1부터 5까지의 점수로 측정하고, 이를 리스크 지향 리스크 결정 매트릭스에 적용한다. 그림 1은 리스크 결정 매트릭스이다. 가로축은 요구사항 추출 과정 중에 식별된 유스 케이스이고, 세로축은 각각의 유스 케이스에서 식별된 리스크 요구사항이다. 리스크 요구사항와 유스 케이스간의 연관관계정도를 상, 중, 하로 구별하여 9점,3점,1점으로 측정하고 이를 리스크 결정 매트릭스에 적용한다. 리스크 결정 매트릭스는 다음 식을 계산하여 리스크 영향도(Risk Impact)를 계산할 수 있다. 리스크 영향도 수식은 다음과 같다[1,5].

$$RI = (RP_1 \times R_{i,j}) + (RP_2 \times R_{i,j}) + \cdots + (RP_i \times R_{i,j}) = \sum_{i,j=1}^{n} (RP_i \times R_{i,j})$$

이처럼 리스크 결정 매트릭스를 통하여 리스크 영향도를 계산함으로써 각각의 유스 케이스에 대한 리스크 우선순위를 측정하여 리스크 기반 테스팅에 적용한다.

2.3 유스케이스 점수 기법(UCP)

Karner가 최초로 제안한 유스 케이스 점수 기법은

제품의 유스 케이스 개수, 크기, 복잡도를 평가하여 소프트웨어의 크기를 측정하는 방법이다[9-11]. 기술적 복잡도 인자(Technical Complexity Factor, TCF) 및 환경인자(Environmental Factor, EF)을 통하여 유스 케이스 점수를 계산한다. 먼저 액터의 분류에 따른 규모를 계산한다. 액터의 형태에 따라 단순, 평균, 복잡으로 나누며 가중치를 1, 2, 3으로 둔다. 조정 전 액터 가중치(Unadjusted Actor Weight, UAW)의 계산식은 Σ(# of Actors × WF)이다. 그 다음, 유스 케이스는 트랜잭션 수를 통해서 단순, 평균, 복잡으로 나누어 측정한다. 3개 이하의 트랜잭션은 단순으로 분류하여 가중치를 5로 두고, 4개에서 7개의 트랜잭션은 평균으로 분류하고 가중치를 10으로 두었다. 마지막으로 7개 이상의 트랜잭션은 복잡으로 분류하고 가중치를 15로 두었다. 조정 전 유스 케이스 가중치(Unadjusted Use Case Weights, UUCW)의 계산식은 Σ(# of Use Cases × WF)이다. 기술적 복잡도 인자(TCF)는 총 13개 항목으로 이루어져 있고, 환경인자(EF)는 총 8개 항목으로 이루어져 있다. 이 인자들에 0부터 5까지의 값을 할당하고, 그 값에 -1부터 2까지의 가중치를 곱하여 기술적 복잡도 인자 TCF = 0.6 + (0.01 × TFactor)와 환경 인자 EF = 1.4 + (-0.03 × EFactor)를 계산한다. 조정 인자를 반영하여 최종적으로 유스 케이스 점수를 계산한다. 다음은 유스 케이스 점수 계산식이다.(UUCP=UAW+UUCW).

$$UCP = UUCP \times TCF \times EF \quad [9\text{-}11]$$

3. 유스케이스 복잡도와 리스크 영향도의 상관관계

기존 논문의 자동차 물품관리 시스템의 사례연구[1,5,6]를 재사용하여 유스 케이스 점수와 리스크 영향도 간의 상관관계를 보여주고자 한다. 그림 2는 사례연구의 유스 케이스 다이어그램이다.

3.1 기존 UCP 기반 유스 케이스 복잡도 점수 계산

유스 케이스 점수는 유스 케이스 설계로부터 얻은 기술적 복잡도 인자와 환경 인자를 통하여 계산된다.

먼저 유스 케이스별로 자동차 물품관리 시스템의 액터인 사용자와 관리자, 판매자의 액터의 가중치를 계산하여 액터의 분류에 따른 규모를 계산한다. 또한, 트랜잭션의 수에 따라 유스 케이스 가중치를 계산하여 유스 케이스 분류에 따른 규모를 계산한다. 유스 케이스에 대한 기술적 복잡도와 관련된 인자 13개항목과 환경인자 8개 항목에 점수를 할당하여 기술적 복잡도 인자 및 환경 인자를 계산한다. 마지막으로 계산된 조정인자를 반영하여 유스 케이스 점수를 자동으로 계산해준다. 유스 케이스 점수는 기존에 조정 전 액터 가중치와 조정 전 유스 케이스 가중치를 합한 값이며, 기술적 복잡도 인자

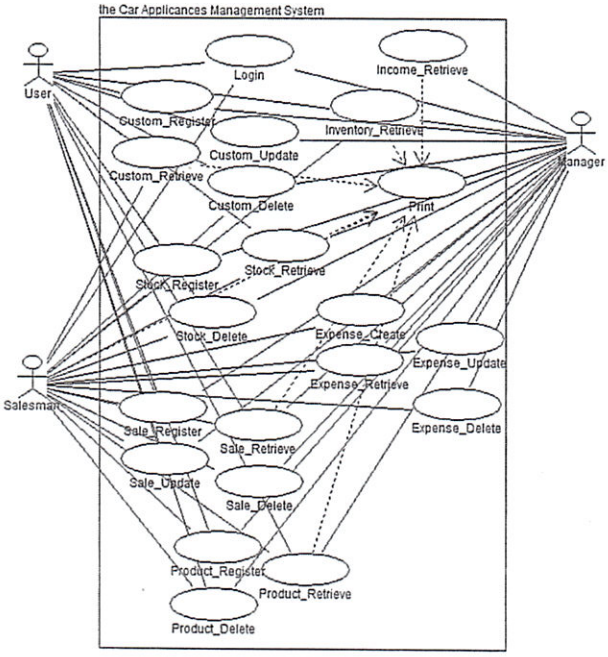

그림 2 자동차 물품관리 시스템의 유스 케이스 다이어그램

Fig. 2 Use Case Diagram of the Car Appliances Management System

와 환경인자를 통하여 계산된다. 유스 케이스 점수를 통하여 유스케이스 별로 유스 케이스 복잡도를 계산할 수 있었으며, 이를 우선순위화 하였다. 다음은 유스 케이스 복잡도 계산 자동화 도구에 적용사례를 적용하여 얻은 결과 값이다. 기존의 유스 케이스 점수를 계산하는 순서에 따라 입력값을 넣으면 UUCP, TCF, EF값이 계산되며 최종적으로 UCP 값을 얻을 수 있다.

표 1 유스케이스 점수 계산

Table 1 Calculated Use Case Point

No.	Use Case	UUCP	TCF	EF	UCP
UseCase 1	Login	10	0.705	1.145	7.05
UseCase 2	Custom_Register	12	0.74	1.145	8.88
UseCase 3	Custom_Update	12	0.705	1.145	8.46
UseCase 4	Custom_Retrieve	12	0.7	1.115	8.4
UseCase 5	Custom_Delete	12	0.68	1.145	8.16
UseCase 6	Stock_Register	12	0.695	1.175	8.34
UseCase 7	Stock_Retrieve	14	0.7	1.1	9.8
UseCase 8	Stock_Delete	12	0.67	1.175	8.04
UseCase 9	Sale_Register	12	0.655	1.175	7.86
UseCase 10	Sale_Retrieve	12	0.695	1.1	8.34
UseCase 11	Sale_Update	12	0.705	1.175	8.46
UseCase 12	Sale_Delete	12	0.655	1.175	7.86
UseCase 13	Product_Register	12	0.67	1.175	8.04
UseCase 14	Product_Retrieve	12	0.685	1.1	8.22
UseCase 15	Product_Delete	12	0.665	1.175	7.98
UseCase 16	Inventory_Retrieve	12	0.665	1.1	7.98
UseCase 17	Income_Retrieve	17	0.725	1.01	12.325
UseCase 18	Expense_Create	17	0.725	1.16	12.325
UseCase 19	Expense_Update	17	0.815	1.16	13.855
UseCase 20	Expense_Retrieve	17	0.735	1.16	12.495
UseCase 21	Expense_Delete	12	0.715	1.16	8.58
UseCase 22	Print	11	0.64	1.4	7.04

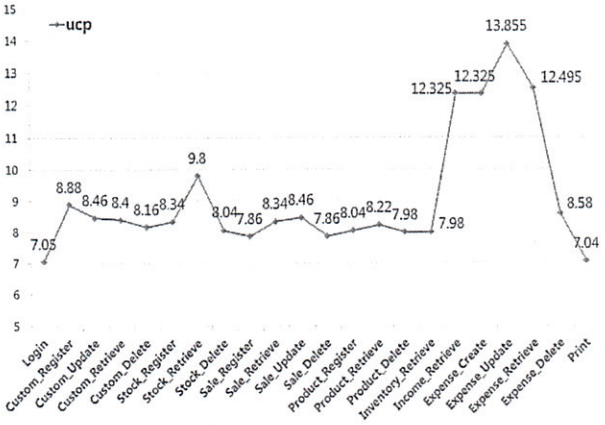

그림 3 유스 케이스 점수 결과 그래프
Fig. 3 Chart for Use Case Point

표 1은 유스 케이스 점수 계산 자동화 도구를 통하여 얻은 UCP 결과 값이며, 유스 케이스 점수를 이용하여 얻은 결과 값을 통하여 소프트웨어의 크기와 복잡도를 예측할 수 있다. 그림 3은 유스 케이스 점수를 유스 케이스별로 도식화 한 것이다. '지출 수정'의 유스 케이스가 가장 높은 값을, '인쇄' 유스 케이스가 가장 낮은 값을 보여준다.

3.2 제안된 리스크 결정 매트릭스 계산

리스크 영향도를 우선순위화해주는 리스크 결정 매트릭스의 도구로 계산한다. 식별된 리스크 요구사항별로 우선순위와 리스크 포인트를 측정한다. 그 다음, 각 리스크와 유스 케이스 간의 관련 정도를 분석 후, 리스크 지향 리스크 결정 매트릭스에 적용한다. 이를 통하여 리스크 영향도(RI) 값을 계산할 수 있다. 리스크 영향도는 리스크 결정 매트릭스 수식을 통하여 계산되었다.

표 2는 유스 케이스 별로 계산된 리스크 영향도 값을 나타내며, 유스 케이스단위로 리스크 우선순위화를 할 수 있다. 그림 4는 유스 케이스 별로 리스크 영향도 값을 도식화 한 것이다. '지출 수정'이 가장 높은 리스크 영향도 값을, '인쇄'기능이 가장 낮은 리스크 영향도 값을 보여준다. 가장 높은 리스크 영향도 값을 가지고 있는 유스 케이스단위가 결국 복잡성도 높다는 점이다. 고로 더욱 주위깊게 테스팅을 진행하여야 하는 것이다.

3.3 UCP and RI 상관관계 분석

표 3은 리스크 결정 매트릭스 자동화도구와 유스 케이스 점수 계산기를 통하여 얻은 두 값의 상관관계를 비교분석한 것이다. 유스 케이스 점수 계산기를 통하여 유스 케이스 점수를 구하였고, 리스크 결정 매트릭스 자동화 도구를 통하여 리스크 영향도 값을 구하였다.

그림 5는 표 3에서 비교한 값을 그래프로 표현한 것이다. 두 그래프의 형태와 유스 케이스 별 값의 분포를 통하여 유스 케이스 점수와 리스크 영향도의 상관관계

표 2 리스크 결정 매트릭스
Table 2 Risk Decision Matrix

	UC_1	UC_2	UC_3	UC_4	UC_5	UC_6	UC_7	UC_8	UC_9	UC_{10}	UC_{11}	UC_{12}	UC_{13}	UC_{14}	UC_{15}	UC_{16}	UC_{17}	UC_{18}	UC_{19}	UC_{20}	UC_{21}	UC_{22}	RP
RR_1	9											9					9	9	9	9	9		5
RR_2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	5
RR_3		9				3			3			9					3						1
RR_4		9	9	9	9																		4
RR_5		9	9	9	9	9	9	9	9	9	9		9	9	9	9							3
RR_6										9									9				2
RR_7										9									9				2
RR_8													9	9	9								3
RR_9							9			9										9		1	2
RR_{10}						9	9	9															4
RR_{11}						9																	1
RR_{12}			3	3			3	1		3	3	1		3	1	3	3			3	1	9	3
RR_{13}																		9	9	9	9		4
RR_{14}									9	9	9	9											1
RI	90	117	135	135	108	120	135	111	84	108	126	102	108	108	102	81	117	129	162	153	129	74	

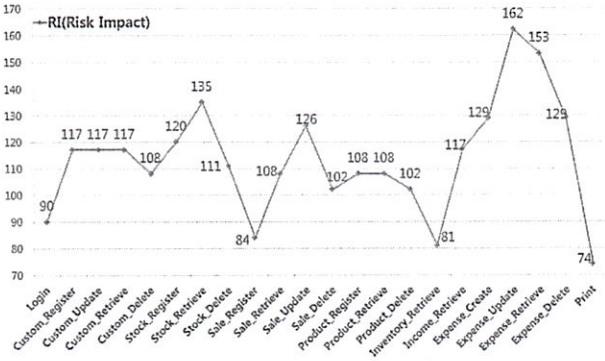

그림 4 리스크 영향도 결과 그래프
Fig. 4 Chart for Risk Impact

에 대한 그래프를 나타낸다. 유스 케이스 점수를 통하여 유스 케이스의 복잡도의 우선순위를 구하였으며, 리스크 영향도를 우선순위 화하였다. 이는 유스 케이스 별로 리스크 영향도 값과 유스 케이스 점수를 각각 순위화를 비교한 그래프를 통하여 유스 케이스 점수(유스 케이스 복잡도)와 리스크 영향도의 유사성을 찾을 수 있었다.

유스 케이스 점수를 측정하면 소프트웨어의 크기와 복잡성을 구할 수 있으며, 유스 케이스 점수가 높을수록 소프트웨어의 크기와 복잡성이 크고 리스크 발생 확률이 높아진다. 자동차 물품관리 시스템에 유스 케이스 점

표 3 유스 케이스 점수와 리스크 영향도 상관관계 분석
Table 3 Analysis of correlation between UCP and RI

No.	UseCase	UCP	Priority
UseCase 1	Login	7.05	21
UseCase 2	Custom_Register	8.88	6
UseCase 3	Custom_Update	8.46	8
UseCase 4	Custom_Retrieve	8.4	10
UseCase 5	Custom_Delete	8.16	14
UseCase 6	Stock_Register	8.34	11
UseCase 7	Stock_Retrieve	9.8	5
UseCase 8	Stock_Delete	8.04	15
UseCase 9	Sale_Register	7.86	19
UseCase 10	Sale_Retrieve	8.34	11
UseCase 11	Sale_Update	8.46	8
UseCase 12	Sale_Delete	7.86	19
UseCase 13	Product_Register	8.04	15
UseCase 14	Product_Retrieve	8.22	13
UseCase 15	Product_Delete	7.98	17
UseCase 16	Inventory_Retriev	7.98	17
UseCase 17	Income_Retrieve	12.325	3
UseCase 18	Expense_Create	12.325	3
UseCase 19	Expense_Update	13.855	1
UseCase 20	Expense_Retrieve	12.495	2
UseCase 21	Expense_Delete	8.58	7
UseCase 22	Print	7.04	22

No.	UseCase	RI	Priority
UseCase 1	Login	90	19
UseCase 2	Custom_Register	117	8
UseCase 3	Custom_Update	117	8
UseCase 4	Custom_Retrieve	117	8
UseCase 5	Custom_Delete	108	13
UseCase 6	Stock_Register	120	7
UseCase 7	Stock_Retrieve	135	3
UseCase 8	Stock_Delete	111	12
UseCase 9	Sale_Register	84	20
UseCase 10	Sale_Retrieve	108	13
UseCase 11	Sale_Update	126	6
UseCase 12	Sale_Delete	102	17
UseCase 13	Product_Register	108	13
UseCase 14	Product_Retrieve	108	13
UseCase 15	Product_Delete	102	17
UseCase 16	Inventory_Retrieve	81	21
UseCase 17	Income_Retrieve	117	8
UseCase 18	Expense_Create	129	4
UseCase 19	Expense_Update	162	1
UseCase 20	Expense_Retrieve	153	2
UseCase 21	Expense_Delete	129	4
UseCase 22	Print	74	22

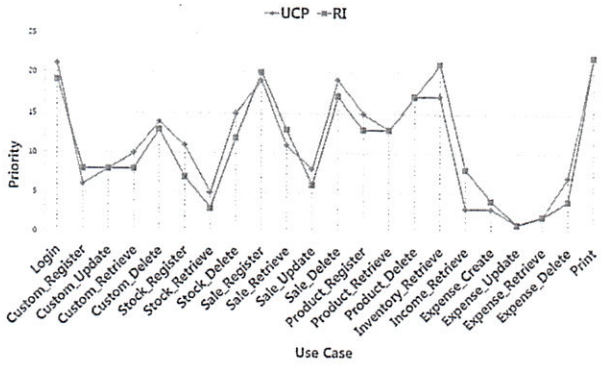

그림 5 UCP와 RI 상관관계 그래프
Fig. 5 Chart for UCP and RI

수를 이용한 테스팅과 제안한 리스크 영향도를 이용한 테스팅을 적용하여 실험한 결과에서, 리스크 영향도가 높을수록 유스 케이스 점수도 높게 나타난다는 것을 확인하였다. 이는 리스크 영향도가 높을수록 소프트웨어의 크기와 복잡성이 커지고, 리스크 발생 확률이 증가한다는 것을 의미한다.

리스크 영향도를 측정하여 리스크 영향도가 높은 것은 우선적으로 주어진 시간과 자원을 할당하여 강력한 테스팅을 진행하고, 리스크 영향도가 낮은 부분은 시간이 허용되는 범위 내에서 테스팅 정도를 정하여 진행한다면 효율적인 리스크 기반 테스팅을 수행할 수 있다. 리스크 영향도를 식별함으로써 어떤 리스크를 감소시킬지 결정하는데 도움이 될 수 있으며, 리스크의 불확실성을 감소시킬 수 있다.

4. 결론 및 향후 연구

본 논문은 소프트웨어 개발에 최적화된 리스크 식별 및 분석에 대한 새로운 프로세스 메커니즘을 제안한다. 기존의 리스크 기반 테스팅은 프로젝트, 프로세스, 제품 리스크에만 초점이 맞추어져 있어 소프트웨어 개발에 최적화 되어있지 않았기 때문이다. 제안한 메커니즘에서는 유스 케이스 단위로 소프트웨어 개발에 관련된 리스크 요구사항을 식별하고, 이를 리스크 결정 매트릭스에 적용하였다. 이를 통해 얻은 리스크 영향도 값과 유스 케이스 점수를 비교분석을 하여 유스 케이스 점수가 높을수록 리스크 영향도도 높게 나타난다는 것을 보였다. 따라서 우리가 제안한 리스크 영향도 값을 측정함으로써 유스 케이스 단위의 리스크 우선순위화가 가능함을 확인하였다. 이는 리스크 결정 매트릭스를 요구공학을 통한 시스템 개발방법에 적용하는 것이 적합한지를 나타내는 것이며, 리스크 영향도가 리스크 식별 및 분석에 사용가능 함을 보여주는 것이다. 리스크 영향도 값을 측정하여 리스크 영향도가 높은 것은 반드시 테스트해야 하는 부분에 해당하기 때문에 우선적으로 주어진 시간과 자원을 할당하여 강력한 테스팅을 진행해야한다. 반면에, 리스크 영향도가 낮은 부분은 시간이 허용되는 범위 내에서 테스팅 정도를 정하여 리스크 기반의 테스팅을 진행한다면 효율적인 리스크 기반 테스팅을 수행할 수 있다. 리스크 영향도의 우선순위를 통하여 감소시켜야 하는 리스크에 대한 우선순위를 얻을 수 있으며, 리스크에 대한 불확실성을 감소시킬 수 있다. 리스크 기반 테스팅은 완벽한 테스팅이 아니기 때문에, 경험 기반의 테스팅 기법을 사용하는 것이 많은 리스크를 발견하는데 도움이 된다. 하지만, 경험자의 수준에 따라 달라질 수 있으며, 가중치의 기준이 미흡하다는 단점이 있다. 향후에는 더 정확한 기준 확립을 위하여 유스 케이스 단위에서 리스크 요구사항을 추출하는 방법 기준을 제안하고자 한다.

References

[1] BoYeon Kim, JaeSeung Kim, BoKyung Park, HyunSeung Son, Robert YoungChul Kim, Woo Yeol Kim "A Risk Extraction and Prioritization through Requirements Analysis based on Use Case Approach," *Proc. of the 40th KIPS Fall Conference*, vol.19, no.2, pp.1519-1522, Nov. 2012.

[2] Boehm, B.W., "Software Risk Management: Principles and Practices," *Software, IEEE*, vol.8, no.1, pp.32-41, Jan. 1991.

[3] Exploring risk-based testing and its implications Felix Redmill, Redmill Consultancy, 22 Onslow Gardens, London N10 3JU, U.K.

[4] Stale Amland, "Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study," *Journal of Systems and software*, vol.53, pp.287-295, 2000.

[5] Boyeon Kim, R.YoungChul Kim, Jae H. Kim "Automatic Computing Tool Development for Risk Decision Matrix Based on Use Case," *Proc. of the KISM Spring Conference*, vol.2, no.1, pp.350-352, May 2013.

[6] Boyeon Kim, R.YoungChul Kim, "A Study on Correlationship Between Use Case Complexity and Riskness (Risk Impact)," *Proc. of the KCC 2013*, pp.577-579, Jun. 2013.

[7] Gerrard P and Thompson N., Risk-Based E-Business Testing, Artech House, Norwood, 2002.

[8] BoKyung Park, "A Study On Requirements Extraction & Prioritization Based on Goal Oriented Requirement Based Testing," *Master's Thesis*, HongikUniv., 2012.

[9] C.R. Synmons, Software Sizing and Estimationg MK II FPA. Wiley-Inter-science, 1991.

[10] SunKyung Lee, "Software effort estimation based on use case transaction," Master's Thesis, KAIST 2010.

[11] [Online]. Available: http://en.wikipedia.org/wiki/Use_Case_Points