

소프트웨어 신뢰도 측정에서 고장률 비교

정혜정

평택대학교 데이터정보학과 교수

Comparison of Failure Rates in Measuring Software Reliability

Jung Hye Jung

Professor, Dept of Data Information & Statistics, Pyeong-Taek University

요약 본 연구는 소프트웨어 품질 특성인 기능적합성, 신뢰성, 사용성, 이식성, 유지보수성, 성능효율성, 보안성, 호환성 중에서 신뢰성에 대한 평가 방안을 연구하였다. 소프트웨어 품질 측정에 있어 신뢰성의 정량적 평가 방안을 제시한다. 본 연구는 소프트웨어 품질 특성 중 하나인 신뢰성 평가 중 성숙성에 포함되는 고장률을 측정하는 방법에 대해서 소개하고 고장 데이터의 형태에 따라 고장률이 어떻게 변화하는지에 대해 실험데이터를 가지고 연구한 내용이다. 소프트웨어 테스트를 중심으로 매일의 고장 수를 중심으로 소프트웨어 신뢰성 성장 모델에 적용하여 고장률을 측정하여 고장 데이터의 형태에 따라 비교 분석하였다. 6일간의 테스트를 통해서 발견된 고장 시간을 중심으로 고장률을 측정하고 측정결과를 이용해서 국제표준 ISO/IEC 25023에서 제시하고 있는 고장률과 비교 하였으며 데이터 형태에 따라 적용방안을 검토하였다.

주제어 : 소프트웨어고장 데이터, 신뢰도 측정, 소프트웨어 신뢰도 성장모형, 고장률, 포아송분포

Abstract This research studied the evaluation of reliability among the software quality characteristics: suitability, reliability, usability, portability, maintainability, performance efficiency, security, and compatibility. It proposes a quantitative evaluation of reliability in the measurement of software quality. This study introduces a method for measuring the failure rate included in maturity during reliability evaluation, which is one of the characteristics of software quality, and is a study with experimental data on how the failure rate changes depending on the form of failure data. Focusing on software testing, the failure rate was measured and compared according to the type of failure data by applying it to the software reliability growth model, focusing on the number of failures per day. The failure rate was measured around the failure time found through the 6-day test, and the failure rate was compared with the failure rate proposed by the international standard ISO/IEC 25023 using the measurement results, and the application was reviewed according to the data type.

Key Words : Software failure data, Reliability measurement, Software reliability growth model, Failure rate, Poisson distribution

1. 서론

국내에서는 소프트웨어 품질 향상을 위해서 많은 연구를 진행하고 있다. 최근에는 AI 제품 출시로 인하여 제품을 정확하게 평가하기 위한 방안 마련을 진행하고 있으며, 제품에 대한 품질 평가를 제도화해서 운영 하

기 위한 방안을 마련하고 있다. 우리 생활이 소프트웨어에 대한 의존도가 높아지면서 소프트웨어 품질에 대한 관심도 높아지고 있으며, 특히 소프트웨어 중 안전과 관련된 소프트웨어 제품에 대해서 품질 향상을 위한 방안으로 시험기관을 통해서 공인인증을 받도록 의

*Corresponding Author : Jung Hye Jung(jhjung@ptu.ac.kr)

Received April 11, 2022

Accepted May 20, 2022

Revised April 30, 2022

Published May 28, 2022

무화 하고 있다. 코로나로 인해서 소프트웨어에 대한 의존도가 높아졌으며 우리 생활도 소프트웨어의 영향으로 많은 변화가 일어나고 있다.

본 연구는 소프트웨어 품질 평가에 있어서 신뢰성에 대한 평가항목 중 고장률에 대한 측정 결과를 제시하고 정확히 측정할 수 있는 측정방안을 제시하려 한다. 최근 소프트웨어는 융합기술을 이용한 융합 소프트웨어가 많아지고 있으며 다양한 공개 소프트웨어를 이용해서 사용자 측면의 편의를 고려해서 개발 되어지고 있다. 국제표준모델 ISO/ IEC 25023에서는 소프트웨어 품질 평가를 위해서 8가지 품질 특성을 제시하고 있다. 8가지 품질 특성은 기능적합성, 사용성, 신뢰성, 효율성, 이식성, 성능효율성, 보안성, 호환성을 들 수 있으며 각각의 품질특성에 따른 부특성과 품질 평가를 위한 메트릭(metric)이 제시되어져 있다.

최근들어 AI 제품이 많아지면서 AI 제품에 대해서 한국표준협회에서는 AI+(에이아이 플러스) 인증 제도를 도입해서, 현장시험과 제품시험을 통해서 제품의 품질 평가를 진행하고 있다. 국내에서는 2000년부터 국내의 소프트웨어 품질 제도 정착화를 위해서 GS(Good Software) 품질 인증 제도를 만들어서 일반 소프트웨어의 품질 평가를 진행하였다. 소프트웨어 품질 평가는 초기에 ISO/IEC 9126-2를 적용하여 품질을 평가하였으나, 현재는 국제표준의 변화로 인해 ISO/IEC 25023을 기반으로 소프트웨어 품질 평가를 진행하고 있다[1].

ISO/IEC25023의 국제표준 문서인 SQuaRE (Software Quality Requirement and Evaluation) 시리즈는 8가지 품질 특성을 기반으로 하여 소프트웨어 품질 평가를 위한 방안을 제시하고 있다.[2] 국내의 소프트웨어 품질 평가를 위해서 사용되고 있는 ISO /IEC 25000 시리즈는 소프트웨어 품질 평가를 위한 기반으로 사용되고 있다[3]. ISO/IEC2502n 시리즈는 소프트웨어 품질 평가 메트릭을 제시한 평가문서로써, ISO/IEC2502n 문서는 ISO/IEC 25021, ISO/IEC 25022, ISO/IEC 25023, ISO/IEC 25024로 구성되어져 있다[4].

각 문서의 내용에서 ISO/IEC 25021에서는 품질 측정 요소에 대한 내용과 소프트웨어 품질 측정을 위한 형식을 제공하고 있다. ISO/IEC 25022에서는 품질 특성에 대한 측정 방법 및 품질 측정 요소를 제시하고 있다. ISO/IEC 25023에서는 시스템 및 소프트웨어의 품

질을 측정하기 위한 방법 및 품질 측정 요소에 대한 평가 방안을 제시하고 있다. ISO/IEC 25024에서는 데이터에 대한 품질 측정 방법을 제시한 문서로써 현재 데이터의 중요성이 인식되면서 데이터 품질 평가에 활용되어지고 있다.

본 연구에서는 국제표준을 기반으로 소프트웨어 신뢰성의 측정에 있어 고장률 측정 방법을 소프트웨어 신뢰성 성장 모델에 적용시켜 측정하려 한다. 특히 본 연구는 소프트웨어 시험과정에서 구한 고장시간과 관련된 데이터 특성을 고려한 신뢰성의 고장률 측정을 제안한 것이다.

본 연구는 ISO/IEC 25023에서 제시하고 있는 다양한 평가 메트릭 중에서도 신뢰성에 대한 평가 항목의 정량적 측정 방안에 대해서 제시하려 한다. 2장에서는 국제표준에서 제시하고 있는 ISO/IEC 25023의 소프트웨어 품질 평가 모델에 대해서 소개한다. 3장에서는 소프트웨어 품질 평가 메트릭 중 신뢰성의 고장률 측정 메트릭을 소개하고, 소프트웨어 신뢰성 성장 모델을 제시하여 시험기관에서 수집한 고장에 관련된 고장 발생 시 고장 시간에 대한 시험 데이터를 이용해서 고장률 측정의 결과를 비교하고 4장에서는 결론을 내려고 한다.

2. 국제표준문서 ISO/IEC 25023

2.1 ISO/IEC 25023 구성

소프트웨어 품질의 사전적 의미로는 소프트웨어 이해 당사자 간에 명시적 요구를 만족하는 정도를 의미한다고 할 수 있다. 시스템 및 소프트웨어의 품질 평가를 위해서 국제표준 ISO/IEC 25023에서는 제품에 대한 품질 특성과 부특성을 통해서 부특성에 따른 소프트웨어의 품질측정 메트릭을 제시하고 있다. 국제표준 ISO/IEC 25023에서 제시하고 있는 품질 메트릭은 제품의 품질을 정량적으로 측정하기 위한 방안제시가 되어 있다. 8가지 품질 특성에 따라서 86개의 품질 메트릭을 제시하고 있어 제시된 형태에 따라 소프트웨어의 품질을 측정한다.

ISO/IEC 25023에서 제시하고 있는 품질 특성으로는 기능적합성(Functional Suitability), 성능효율성(Performance Efficiency), 호환성(Compatibility), 사용성(Usability), 신뢰성(Reliability), 보안성(Security), 유지보수성(Maintainability), 이식성(Portability)이

다. 특히 ISO/ IEC 25023 국제표준 문서에서만 제시되어져 있는 보안성과 상호운영성에 대한 평가는 현대 소프트웨어의 특징이 잘 반영된 사례로 보여진다.

ISO/IEC 25023에서 제품이나 시스템의 명세 된 기능이 제대로 작동하는지를 평가하기 위한 기능적합성에 대한 평가가 제시되어져 있으며, 기능적합성에 대한 평가는 기능완전성과 기능정확성과 기능적질성을 부특성으로 해서 평가항목이 제시되어져 있다. 성능효율성은 각 TASK에 대해서 반응시간과 반환시간 등을 측정하는 시간반응성과 프로세스 사용률, 메모리 사용률 등을 측정하는 자원사용률과 동시 접속자를 고려한 용량측정이라는 부특성으로 구성되어져 있다. 호환성은 공존성과 상호운영성을 중심으로 평가하도록 제시되어져 있어 데이터 포맷과 관련된 내용과 타 소프트웨어 사용에 있어 공존성을 평가할 수 있다. 사용성은 적절한 인지성, 학습성, 운용성, 오류방지성, 인터페이스 심미성, 접근성을 평가하도록 부특성이 제시되어져 있어, 사용자 관점에서 제품을 사용하는데 있어 편의성을 제공하는 정도를 측정하도록 되어 있다. 신뢰성은 성숙성, 가용성, 결함 허용성, 복구성을 부특성으로 하여 소프트웨어 품질을 측정하도록 되어 있으며 소프트웨어를 사용하면서 발생한 고장을 중심으로 신뢰성을 평가하도록 제시되어져 있다. 보안성은 기밀성, 무결성, 부인방지, 책임성, 신원인증성 등의 부특성을 기반으로 제품에 대한 품질을 평가하도록 제시되어져 있다. 유지보수성은 모듈성, 재사용성, 분석성, 변경성, 시험성 등의 부특성을 기반으로 제품의 기능이 업데이트 되었을 때 기본적인 사항들을 평가하도록 제시되어져 있다. 이식성은 적응성, 설치성, 대치성을 통해서 제품의 설치와 관련된 내용을 평가하도록 구성되어져 있다.

ISO/IEC 25023에서 제시되어 있는 8개 품질 특성에 따른 86개의 평가 메트릭은 정보통신고시에서 제시하고 있는 소프트웨어 품질평가의 모델을 만들기 위한 기초자료로 활용되어지고 있다. 현재 GS(Good Software)인증을 담당하고 있는 TTA(한국정보통신기술협회), KTL(한국산업기술시험원), KTC(한국기계전기전자시험연구원), KTR(한국화학융합시험연구원), 동의대학교 산학협력단의 5개 기관에서는 정보통신고시를 기반으로 한 소프트웨어 품질 평가 메트릭을 구성해서 품질에 대한 평가를 실시하고 있다.

2.2. 품질 평가 메트릭

본 연구를 위해서 국제표준에서 제시하고 있는 ISO/IEC 25023의 품질평가를 위한 평가 메트릭 구성을 살펴보면 Table 1과 같다.

Table 1. Quality Metrics

ISO/IEC 25023	Functional Suitability
	Performance Efficiency
	Compatibility
	Usability
	Security
	Maintainability
	Portability
	Reliability

Table 1에서 제시한 8가지 품질 특성은 품질특성에 따라 부특성으로 분류되며 기능성의 부특성은 Table 2와 같이 분류된다.

Table 2. Functional suitability metric

Metric Name	Measurement Contents	Measure Function
Functional Coverage	Percentage of the specified functions implemented	$X = 1 - \frac{A}{B}$ A: Number of missing features B: Specified number of functions
Functional Correctness	The percentage of features that provide accurate results	$X = 1 - \frac{A}{B}$ A: Number of features to be considered B: Number of tasks that would be good to have a demo function
Functional appropriateness of using objective	Ratios that provide adequate results	$X = 1 - \frac{A}{B}$ A: Number of functions missing from a specific purpose B: Number of functions for a specific purpose

Table 2는 소프트웨어 품질 평가에 있어서 기능적합성(Functional Suitability)을 평가하기 위한 평가 메트릭이다. 8개 품질 특성은 Table 2의 기능적합성과 같이 품질 평가를 위한 메트릭으로 구성되어져 있다.

3. 신뢰성 고장률 측정

ISO/IEC 25023의 신뢰성의 성숙성(Maturity) 측정은 결함수정(Fault Correction), 평균 고장 사이 시간(Mean Time Between Failure: MTBF), 고장률(Failure Rate), 테스트 커버리지(Test Coverage)를 측정하도록 메트릭이 구성되어져 있다. 본 연구에서 측정하려고 하는 고장률에 대한 메트릭 측정은 Table 3과 같다.

Table 3. Failure rate Metrics

Metric Name	Measurement Contents	Measure Function
Failure rate	Average number of failures over a set period of time	$X = \frac{A}{B}$ A: Number of failures detected during observation time B: observation time

소프트웨어 신뢰성 성장 모델을 적용하여 고장률을 측정하기 위한 비동질적 포아송 분포의 가정은 다음과 같다[5]. 비동질적 포아송분포(Non Homogeneous Poisson Distribution : NHPP)에 적용해서 모델을 적용하기 위해서는 첫째, t 시간까지 발견된 고장의 누적 수는 포아송분포를 따르야 한다는 것이다. 둘째, 모든 발생하는 고장은 독립이며, 고장이 발견될 가능성은 확률적을 같다는 것이다. 셋째, 모든 발견된 고장은 즉시 제거 되어져야 하며 새로운 고장을 만들어 내지 않는다는 것이다. 세 가지 가정하에 비동질적 포아송분포에 대한 소프트웨어 신뢰성 성장 모델의 대표적 모델을 살펴보면 다음과 같다.

첫째, Goel-Okumoto(GO) 모델에 대해서 평균값 함수는 식 (1)과 같다[6,7].

$$m(t) = a[1 - \exp(-bt)], \quad a > 0, \quad b > 0 \quad (1)$$

여기에서, a는 소프트웨어 시험 과정에서 발견 되어질 수 있는 전체 결함 수, b는 결함 수에 대한 실패 발생 비율이다. 이때 고장률 함수는 식 (2)와 같다.

$$\lambda(t) = \frac{dm(t)}{dt} = abe^{-bt} \quad (2)$$

또한, T시간 테스트가 진행되어지고 남아 있는 결함

수에 대한 것은 식 (3)과 같다.

$$\begin{aligned} E(T) &= m(\infty) - m(t) & (3) \\ &= a - a(1 - e^{-bt}) \\ &= ae^{-bt} \end{aligned}$$

둘째, S-shaped NHPP 모델의 평균값 함수는 식 (4)와 같다.[8]

$$m(t) = a[1 - (1 + bt)\exp(-bt)], \quad b > 0 \quad (4)$$

위의 평균값 함수에 대해서 고장률을 구하면 식 (5)와 같다.

$$\lambda(t) = \frac{dm(t)}{dt} = ab^2te^{-bt} \quad (5)$$

평균값 함수에서 남아 있는 고장 수에 대한 결과는 식 (6)과 같다.

$$\begin{aligned} E(T) &= m(\infty) - m(t) & (6) \\ &= a - a(1 - (1 + bt)e^{-bt}) \\ &= a(1 + bt)e^{-bt} \end{aligned}$$

셋째, Jelinski-Moranda(JM) 모델의 함수를 이용하여 소프트웨어 신뢰성 검정을 위한 JM 모델의 밀도함수는 식 (7)과 같다[9].

$$\begin{aligned} f(t_i) &= (N - i + 1) \phi \exp(-(N - i + 1)\phi t_i), & (7) \\ & i = 1, 2, \dots, N \end{aligned}$$

여기에서, N은 테스트 시작 전 총 고장 수, ϕ는 각 고장에 의한 고장강도이다. 모델에서 모수 N, ϕ는 최대우도추정법에 의해서 추정되어져야 한다[10,11].

식 (7)을 이용해서 고장률 함수를 구하여 보면 식 (8)과 같다.

$$\lambda(i) = \frac{df(t)}{dt} = (N_0 - i + 1)\phi \quad (8)$$

이때 신뢰도를 측정하기 위한 신뢰도 함수는 식 (9)와 같다.

$$R(t) = \exp(-(N_0 - i + 1)\phi t) \quad (9)$$

위의 결과를 이용하여 소프트웨어 시험 중 발견된 고장을 중심으로 소프트웨어 신뢰성 성장 모형을 중심으로 고장률을 측정된 결과와 ISO/IEC 25023을 중심으로 하여 제시된 측정항목을 비교하여 고장률을 측정된 결과는 Table 4와 같다.

Table 4는 6일간 소프트웨어 시험을 하면서 발견된 고장 시간에 대한 자료이다. 이 자료는 소프트웨어 시험 기관의 테스터에게 시험 중 고장이 발견되었을 때 고장이 발견된 시간을 기록하도록 의뢰하여 받은 자료이다.

Table 4. Failure time(minute)

day	failure time
1	53 48 55 50 52 52 60 50
2	30 33 30 35 33 30 30 35 35 31 45 54
3	20 25 23 21 25 27 28 27 29 25 28 20 20 27 22 27 28 29 25
4	40 35 30 48 44 45 44 40 41 40 48
5	106 89 101 86 98
6	137 245

Table 4의 자료는 6일간의 소프트웨어 시험 평가 결과를 이용해서 측정된 결과, 첫째 날 고장이 8개, 둘째 날 고장이 12개, 셋째 날 고장이 19개, 넷째 날 고장이 11개, 다섯째 날 고장이 5개 여섯째 날 고장이 2개 발견되어서 처음에는 고장이 점점 증가하다가 최종적으로 감소하는 형태의 속성을 가지고 있는 소프트웨어 테스트 오류 시간 데이터이다. NHPP모델의 GO 모델을 이용하여 고장률을 측정하여 보면 고장률은 0.028 이었고 S-shaped NHPP 모델에 적용하면 고장률은 0.021 이었다. 또한 JM 모델을 이용하여 고장률을 측정하여 보면 0.033으로 측정되었다. 현재 소프트웨어 품질 측정을 하기 위해서 활용되어지고 있는 ISO/IEC25023의 고장률 측정 항목을 이용해서 단위를 분으로 하여 고장률을 측정하면 고장률은 0.019로 측정되었다. 고장률이 다소 차이는 있으나 데이터의 특성을 반영하면 S-shaped NHPP 모델이 가장 고장률을 잘 예측한다.

다음은 소프트웨어를 평가해서 시험한 결과를 위의 Table 4와 같이 분 단위로 고장 발생 시간을 측정하여 6일간 소프트웨어 시험을 하면서 고장 발생 시에 고장발생 시간을 측정된 자료를 수집한 결과는 Table 5와 같다[12].

Table 5. Failure time(minute)

day	failure time
1	17 15 18 15 15 26 24 27 26 25 23 30 28 29 32 27 33 38 28
2	21 24 28 34 39 32 37 38 32 33 34 35 35 30
3	38 42 34 36 33 39 38 40 37 49 44 46
4	57 56 55 45 60 53 57 55 40
5	75 83 78 84 82 105
6	124 125 130

Table 5를 보면, 첫째 날 고장 19개, 둘째 날 고장 14개, 셋째 날 고장 12개, 넷째 날 고장 9개, 다섯째 날 고장 6개, 여섯째 날 고장 3개가 발견된 경우로 이 경우는 계속적으로 고장의 수가 감소하는 형태의 속성을 가지고 있는 소프트웨어 고장 수에 대한 시험 자료이다. 이 경우 GO 모델을 이용하여 고장률을 측정하여 보면 0.032로 측정되었으며, S-shaped NHPP 모델을 이용하여 고장률을 측정하여 보면 0.028로 측정되었으며, JM모델을 이용하여 고장률을 측정하여 보면 0.023으로 측정되었다. 국제표준에서 제시하고 있는 고장률을 메트릭을 이용하여 측정하면 0.0219로 나타났다 [13]. 이자료는 계속적으로 고장이 줄어들고 있어 모델의 모형으로 비교하여 보면 JM 모델이 가장 정확하다는 것을 확인할 수 있다.

첫 번째 주어진 자료의 경우는 일반적으로 소프트웨어 신뢰성 데이터의 속성을 만족하는 경우로 초기에 소프트웨어 고장 수가 증가하다가 감소하는 형태를 만족하여 S-shaped NHPP 모델이 가장 정확한 고장률을 측정함을 알 수 있다. 두 번째 고장에 대한 자료는 계속적으로 고장 수가 감소하는 형태의 속성을 지니고 있어 JM모델의 고장률이 가장 정확함을 알 수 있다[14]. 이러한 고장률 측정의 결과는 현재 하나의 메트릭으로 데이터의 특성을 고려하지 않은 채 신뢰성을 측정하고 있으나 데이터의 특성을 고려하여 신뢰성을 측정할 수 있도록 제안되어야 한다. 본 연구 결과는 ISO/IEC 25023 신뢰성 측정을 데이터 특성을 고려하여 측정할 수 있도록 제시된 것이다[15].

4. 비교분석

본 연구에서는 소프트웨어의 신뢰성 검증에서 고장률을 측정하는 방법에 대한 연구를 진행했다. 현재 국제표준 ISO/IEC 25023에서는 소프트웨어 시험을 위

해서 제공된 소프트웨어 시험시간에서 고장 수의 비율로 단순히 고장률을 측정하고 있으나 앞의 두 가지 고장데이터에 대한 예제만 살펴보더라도 고장의 속성에 따라서 다소 고장률이 다르게 측정됨을 알 수 있다. 고장률에 대한 정확한 측정은 고장데이터의 속성을 파악해서 정확한 소프트웨어 신뢰성 성장 모델을 적용해서 고장률을 측정하는 것이 바람직하다.

소프트웨어의 품질 특성 중에서 신뢰성에 대한 측정은 상당히 중요하다고 할 수 있으며, 특히 고장률 측정에 대한 것은 소프트웨어 품질 향상을 위해서 상당히 중요하다고 할 수 있다. 정확한 신뢰성의 고장률 평가를 위해서는 고장 데이터에 대한 속성 분석과 고장 데이터에 따른 소프트웨어 신뢰도 함수를 적용하는 것이 신뢰도의 고장률 측정에 바람직하다고 볼 수 있다. 앞으로 좀 더 많은 소프트웨어 고장데이터를 이용해서 고장데이터에 대한 속성별 고장률 측정을 위한 소프트웨어 신뢰도 함수를 연구할 것이다.

REFERENCES

- [1] ISO/IEC 9126-2. (2003). *Software Engineering - Product Quality -Part 2: External metrics*.
- [2] ISO/IEC 25010. (2011). *System and software engineering-System and software Quality Requirements and Evaluation(SQuaRE) -System and software quality model*.
- [3] ISO/IEC 25023. (2015). *System and software engineering-System and software Quality Requirements and Evaluation(SQuaRE) - Measurement of system and software product quality*.
- [4] ISO/IEC 25000. (2005). *System and software engineering: System and software Quality Requirements and Evaluation(SQuaRE) -Guide to SQuaRE*.
- [5] H. J. Jung & G. H. Han. (2019). The Software Reliability Growth Model base on Software Error Data. *Journal of the Korea Convergence Society*, 10(3), 59-66.
DOI : 10.15207/JKCS.2019.10.3.059
- [6] H. J. Jung & G. H. Han (2020). Text analysis of software test report. *Journal of the Korea Convergence Society*, 11(3), 59-66.
DOI : 10.15207/JKCS.2020.11.11.025
- [7] A. L. Goel & K. Okumoto. (1979), Time dependent error-detection rate model for software reliavbility and other performance measures. *IEEE Trans. Reliability*, R-28, 206-211.
- [8] H. J. Jung. (2014). The Effect Analysis of Software, *The Journal of Digital Police & Management*, 12(1), 271-377.
DOI : 10.14400/JDPM.2014.12.1.371
- [9] H. J. Jung. (2003). *Performance Evaluation of Software Reliability Growth Model Using Plot of Fault Data*. Korea Information Processing Society.
- [10] H. J. Jung. (2018). Reliability measurement applied to software quality assessment metrics. *The Journal of Multimedia*, 21(1), 373-376.
- [11] H. J. Jung. (2019). The software quality measurement based on software reliability model. *Journal of the Korea Convergence Society*, 10(4), 45-50.
DOI : 10.15207/JKCS.2019.10.4.045
- [12] S. W. Kang & H. S. Yang. (2013). Quality Evaluation of Criterion Construction for Open Source Software. *The Journal of Digital Police & Management*, 11(2), 323-330.
DOI : 10.14400/JDPM.2013.11.2.323
- [13] S. Y. Kim, Y. T. Kim & S. J. Lee. (2015). Influence Comparison of Customer Satisfaction using Quantile Regression Model. *The Journal of Digital Police & Management*, 13(6), 125-132.
DOI : 10.14400/JDC.2015.13.6.125
- [14] W. I. Keon. (2010). *Software Testing for Developer*, STA.
- [15] W. I. Keon & G. S. Lee. (2010). 'Software Testing', Aicon.

정혜정(Hye-Jung Jung)

[정회원]



- 1988년: 경북대학교 통계학과 조기 졸업(이학사)
- 1991년: 경북대학교 대학원 통계학과 졸업(이학석사)
- 2004년 : 경북대학교 대학원 통계학과 졸업(이학박사)
- 2008년 7월 ~ 2009년 6월: UNLV 교환교수
- 1995년 ~ 현재 : 팽택대학교 데이터정보학과 교수
- 2001년 ~ 현재 : 금융결제원자문위원, 팽택시 노사협의 회자문위원, ISO/IEC JTC1/ SC7위원, ISO/IEC SC 34 위원, 소프트 품질인증위원, 국가기술표준원 정보기술위원, 멀티미디어학회이사, 경기정보산업협회 편집위원장, 한국융합학회 이사, 한국디지털정책학회 이사 등
- 관심분야 : 소프트웨어 품질, 소프트웨어 신뢰성평가, 빅 데이터분석, 소프트웨어 국제 표준화 분야 등
- E-mail : jhjung@ptu.ac.kr