

논문 2011-1-5

차세대 소프트웨어 품질 표준 SQaRE의 이해

김유경*, 도경구*

Understanding SQaRE - The Next Generation of the International Standard on Software Product Quality

Yukyong Kim*, Kyung-Goo Doh**

요 약

본 논문에서는 현재의 국제품질 표준인 ISO/IEC 9126과 14598 시리즈를 소개하고, 품질 모델들이 직면한 문제점들을 살펴본다. 차세대 표준 모델로 정의된 SQaRE의 구조를 설명하고, 국내 소프트웨어 품질 표준화 현황 및 문제점에 대해 고찰해본다.

Abstract

In this paper, overview of current ISO/IEC 9126 and 14598 series are introduced and some problems in those series for the improvement are identified. Then the concept and architecture of SQaRE which is the next generation of the standard on software product quality is discussed. In addition, we consider the present state and the confronted problem of standard on software product quality in Korea.

한글키워드 : 소프트웨어 품질, ISO/IEC 9126, SQaRE

1. 서 론

소프트웨어 품질 모델은 소프트웨어 품질에 영향을 주는 애플리케이션의 속성을 평가하기 위한 프레임워크로 사용되고 있다. 소프트웨어는 다양한 영역에서 폭 넓게 사용되고 있으며, 소프

트웨어의 정확하고 올바른 운영이 비즈니스를 성공으로 이끄는 관건이 되기도 한다. 따라서 고품질의 소프트웨어 제품을 선택하고 개발하는 것은 가장 기본적이면서 중요한 일이다.

소프트웨어 제품 품질의 정확한 명세(specification)와 평가는 적절한 품질을 보장하기 위해 필요한 핵심 요소이다. 소프트웨어 제품의 사용 목적을 충분히 설명할 수 있는 적합한 품질 속성을 정의하고, 이를 기반으로 널리 채택되어 사용되는 검증된 메트릭을 이용하여 모든 소프트웨어 제품 품질 속성을 명세(specify)하고 평가한

* 한양대학교 ERICA 캠퍼스 컴퓨터공학과
(email: {yukyong,doh}@hanyang.ac.kr)

접수일자: 2011.3.15 수정완료: 2011.5.11

※ 본 연구는 교육과학기술부/한국연구재단 우수연구센터 육성사업의 지원으로 수행되었음 (과제번호 2011-0000974)

다. 소프트웨어 제품 품질 보증을 제품이 개발되는 프로세스에 대한 인증 및 최종 제품의 품질 평가를 통해 이루어진다.

소프트웨어 최종 제품의 품질을 평가하기 위해서는 제품을 잘 나타낼 수 있는 품질 속성들이 필요하다. 품질 속성과 속성들 사이의 관계를 정의해 놓은 것을 품질 모델(quality model)이라고 하며, 품질 요구사항을 명세하고 품질을 평가하기 위한 기초를 제공한다.

1991년 이래로 ISO/IEC JTC1/SC7/WG6은 소프트웨어 제품 품질에 대한 두 가지 국제 표준인 ISO/IEC 9126과 14598 시리즈를 개발했다 [1-6]. 2001년 개선된 ISO/IEC 9126 시리즈는 소프트웨어 제품의 여섯 가지 품질 속성을 정의하고, 소프트웨어 제품 품질에 대한 3가지 관점으로 사용상의 품질(quality in use), 외부 품질(external quality), 그리고 내부 품질(internal quality)을 소개하고 있다 [7-10]. ISO/IEC 9126은 광범위하게 참조되고 사용되어 성공적인 표준으로 자리 잡았다. 그러나 WG6는 새로운 아키텍처의 등장과 같은 IT 환경 변화에 따른 포괄적인 가이드(umbrella guide) 마련에 대한 문제에 직면하여, 개선에 대한 필요성을 인식하게 되었다. 이에 따라 2005년 9126과 14598 시리즈를 통합하여 ISO/IEC 25000 형식으로 이루어진 차세대 품질 모델 SQuaRE(Systems and Software Quality Requirements and Evaluation)가 정의되기 시작했다. 2005년 ISO/IEC 25000이 처음 소개되고, 2011년 9126-1을 대신하는 ISO/IEC 25010이 제출되었다 [11].

본 논문은 현재 9126과 14598 시리즈의 구성과 당면한 문제점들을 살펴보고, 차세대 표준 모델인 SQuaRE의 개념 및 구조를 소개한다. 또한 SQuaRE에서 새롭게 정의된 품질 요구사항(Quality Requirements) 부분을 자세히 설명하고, 국내 소프트웨어 품질 평가 표준화 현황 및 문제

점을 고찰해 보고자 한다.

본 논문의 나머지는 다음과 같이 구성된다. 2장에서는 소프트웨어 품질 모델에 대한 개요와 ISO/IEC 9126과 14598 시리즈의 현황을 기술한다. 3장에서 SQuaRE 개념과 구조를 소개하고, 소프트웨어 품질 요구사항에 대해 살펴본다. 4장에서는 소프트웨어 제품 품질 평가에 관한 국내 표준화 현황에 대해 정의하고, 5장에서 결론으로 마무리한다.

II. 소프트웨어 품질 모델

2장에서는 가장 널리 알려진 McCall의 소프트웨어 품질 모델을 소개하고, ISO/IEC 9126과 14598 품질 표준 모델을 기술한다.

1. McCall의 소프트웨어 품질 모델

McCall이 정의한 소프트웨어 품질 모델은 운영상 특징을 표현하는 제품 운영(product operation) 측면, 제품의 변화를 수용하는 능력을 나타내는 제품 개정(product revision) 측면과 새로운 환경에 대한 적응력을 나타내는 제품 전환(product transitions) 측면으로 나누어 11 가지 품질 요소(quality factor)를 정의하고 있다. 각 품질 요소는 세부적인 품질 기준 항목과 그에 따른 매트릭으로 구성되어 있다 [12].

McCall의 품질 모델은 외부 품질 요소와 제품 품질 기준 사이의 관계를 평가하기 위한 것으로, 시스템 개발자와 시스템 개발 프로세스를 겨냥해서 개발되었다. 사용자의 관점과 개발자의 우선 순위 모두를 반영할 수 있도록 소프트웨어 품질 요소의 개수에 초점을 맞추으로써 사용자와 개발자 사이의 격차를 해소하기 위한 시도를 했다. 그러나 품질 요소와 매핑되는 기준 항목 간에 매우 많은 중복이 이루어지고 있다.

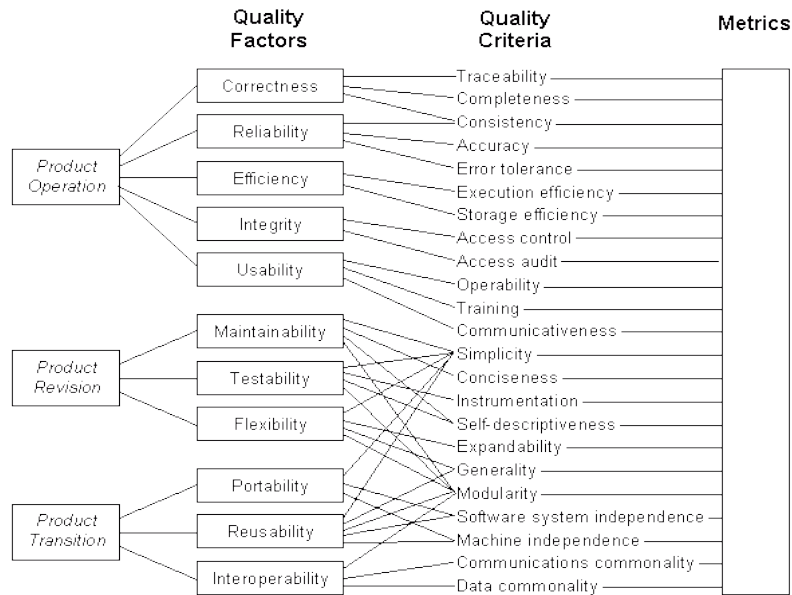


그림 1. McCall의 품질 모델
Figure 1. McCall's Quality Model

McCall의 모델 이외에 Boehm [13], Dromey [14]의 품질 모델이 있으며, 이들은 주로 개발이 완료된 소프트웨어 애플리케이션을 위한 품질 프레임워크를 제공하고 있다. Boehm의 모델은 프로그램의 효율성과 관련해서 소프트웨어 제품을 평가하고자 했으며, 사용자의 요구(needs)을 고려하고 있다. 그 외에 휴렛팩커드에서 개발한 FURPS 모델 [15]은 소프트웨어 품질을 기능(functional) 요구사항과 비기능(non-functional) 요구사항으로 나누어 정의하였다.

2. ISO/IEC 9126과 14598 품질 모델

ISO/IEC 9126과 14598 시리즈는 범용 소프트웨어의 품질을 정의한 대표적인 표준 모델이다. 소프트웨어 품질특성에 관한 표준은 1991년에 제정된 후 1994년부터 품질특성과 내/외부 메트릭을 조정하고 품질 측정 절차를 별도의 ISO/IEC 14598 표준으로 분리하는 등의 개정작업이 진행되어 현재의 모습을 갖추게 되었다. 그림 2는 현재의 9126과 14598 시리즈의 구조를 정리한 것이다.

ISO/IEC 9126(Information Technology-Software Quality Characteristics and Metrics)은 품질 특성 및 메트릭을 정의하고 있는 표준으로, 소프트웨어 품질을 측정·평가하기 위해 소프트웨어의 품질요소와 특성을 정의하고 개발공정에서 품질을 객관적으로 정량화하는데 요구되며, 일반적으로 이러한 품질 모델은 계층구조로 세분화되어 표현된다. 최상위 계층은 사용자 관점에

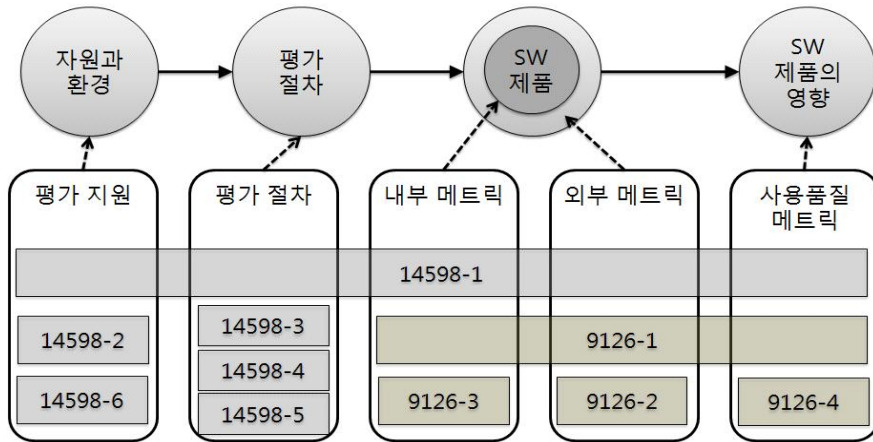


그림 2. ISO/IEC 9126과 14598의 현재 구조 [16]

Figure 2. Current architecture of ISO/IEC 9126 and 14598 Series

서 소프트웨어의 품질 목표를 정의하고, 제2계층은 품질 목표를 달성할 수 있는 광범위한 품질특성(characteristics)을, 제3계층은 상위 특성을 구성하는 구체적인 품질부특성(sub-characteristics)을 갖게 된다. 그리고 최하위 계

층에는 소프트웨어 특성을 측정하기 위한 메트릭(metric)이나 품질인자가 위치하게 된다. 표 1은 9126의 품질모델에서 정의한 품질특성과 각 품질특성에 대응하는 부특성들을 보여준다. 부특성들에는 메트릭들이 대응하고 있으며 외부메트릭

표 1. ISO/IEC 9126의 소프트웨어 품질 특성

Table 1. Software quality characteristics of ISO/IEC 9126

| 품질 특성 | 개념 | 품질부특성 |
|-------------------------|---|----------------------------|
| 기능성 (Functionality) | 소프트웨어가 특정조건에서 사용될 때, 명시된 요구와 내재된 요구를 만족하는 기능을 제공하는 능력 | 적합성, 정확성, 상호 운용성, 보안성, 준수성 |
| 신뢰성 (Reliability) | 명시된 조건에서 사용될 때, 일정한 성능 수준을 유지할 수 있는 능력 | 성숙성, 결함허용성, 회복성, 준수성 |
| 사용성 (Usability) | 명시된 조건에서 사용 될 때, 사용자에게 의해 이해되고 학습되고 사용되고 선호될 수 있는 능력 | 이해성, 학습성, 운용성, 준수성, 친밀성 |
| 효율성 (Efficiency) | 명시된 조건에서 사용되는 자원의 양과 요구된 성능을 제공하는 능력 | 시간반응성, 자원효율성, 준수성 |
| 이식성 (Portability) | 한 환경에서 다른 환경으로 전환될 수 있는 능력 | 분석성, 변경성, 안정성, 시험성, 준수성 |
| 유지보수성 (Maintainability) | 소프트웨어 제품이 변경(수정/개선/개작)되는 능력 | 적응성, 설치성, 공존성, 대체성, 준수성 |

(ISO/IEC 9126-2)과 내부메트릭(ISO/IEC 9126-3), 그리고 사용품질 메트릭(ISO/IEC 9126-4)으로 구성되어 있다.

ISO/IEC 14598(Information Technology-Software Product Evaluation)은 소프트웨어 품질평가 프로세스에 관한 국제표준으로서 소프트웨어 제품의 품질을 측정하거나 평가하는데 필요한 방법과 절차를 그림 3과 같이 여섯 부분으로 나누어서 정의하고 있다.

평가 명세서 작성, 평가계획 수립, 평가수행 및 결과도출 등의 단계를 제시하고, 평가 명세서를 작성할 때는 ISO/IEC 9126에 따른 내·외부 메트릭을 활용하도록 하고 있다. 메트릭을 적용하여 평가를 수행하는 과정에서 객관성 및 공정성을 확보하기 위하여 각 메트릭 적용절차 및 기준 등을 명시한 평가모듈 라이브러리를 이 표준에서 제공하고 있기도 하다.

III. SQuaRE 개요

1. ISO/IEC 9126과 14598의 문제점

1) 새로운 아키텍처 및 포괄적인 가이드에 대한 필요성

ISO/IEC 9126과 14598은 서로 매우 밀접하게 연관되어 있으나 이들이 의미 없는 일련번호들을 기억하는 것이 매우 어렵다. 이런 사실들은 표준이 대중적으로 널리 채택되어 이용되는데 장애요소로 작용하게 된다. 따라서 소프트웨어 시스템에서 좋은 아키텍처가 중요하듯이, 일련의 표준안에 대해서도 구조에 대한 개선이 필요하다.

또한, 소프트웨어 품질은 품질 모델에 기반을 두고 품질요구사항에 대비되는 메트릭으로 평가해야 하므로, 소프트웨어 제품 품질 표준(ISO/IEC 9126)과 이를 평가하는 표준(ISO/IEC 14598)이 분리되어서는 포괄적인 품질 가이드의 역할을 제공하기에 부족하다.

2) 메트릭에 대해 제기된 문제들

내부/외부 메트릭과 같은 서로 다른 메트릭을 조합하여 공통적으로 사용되는 메트릭이 존재하



그림 3. ISO/IEC 14598의 구성

Figure 3. Structure of ISO/IEC 14598

며, 제품 품질을 평가하기 위해 필요한 몇몇 메트릭은 컴파일 시간과 같이 소프트웨어 제품의 속성이 아닌 것을 측정하기 위해 사용되고 있다. 사이클로메틱 복잡도(cyclomatic complexity), 모듈의 응집도(cohesion)와 팬인(fan-in)과 같은 메트릭에 대해서는 단지 제품 품질과 관련 있다고 알려져 있을 뿐, 품질특성과 직접적으로 상호 관련성이 있음을 검증하지 못한 경우도 있다. 또한, 메트릭에 대한 중요한 정보가 표준 시리즈의 여러 부분에 흩어져 있어서, 깊이 있는 이해를 위해 필요한 서로 연관된 정보들을 찾고 이해하는데 어려움을 겪게 된다.

3) 품질 요구사항 표준의 부족

품질 모델과 메트릭은 제품 평가뿐만 아니라 품질 요구사항 명세화(specification)를 포함한 다른 다양한 목적으로도 유용하게 사용될 수 있다. 특히 품질 요구사항이 명확하게 명시되었을 때에만 품질 평가를 수행할 수 있으며, 의미 있는 평가가 이루어질 수 있다. 그러나 품질 요구사항 표준이 이들 시리즈의 일부가 아니라 독립적인 표준으로 제공된다면, 사용자들에게 또 다른 혼란을 야기할 수밖에 없다. 이에 국제표준위원회 워킹그룹에서는 품질 요구사항을 표준 시리즈에 추가하여 제안하기로 결정했다.

2. SQuaRE 개요

앞에서 언급한 문제들에 직면하여, 기존의 소프트웨어 제품 평가 표준들에 대한 개정의 필요성이 대두되고 표준 개발의 체계를 확보하기 위한 목적으로 프로젝트가 탄생되었고, ISO/IEC 25000이란 이름으로 SQuaRE(Software Quality Requirement and Evaluation) 프레임워크가 개발되어 표준화되고 있다. ISO/IEC 9126과 14598과

같이 그동안 개별적으로 개발된 소프트웨어 제품 품질 표준이 SQuaRE라는 하나의 구조(Framework)를 이루게 된다. 그림 4와 같이 SQuaRE는 “4+1”의 구조를 갖는다.



그림 4. SQuaRE 표준 구조

Figure 4. Architecture of the SQuaRE

- 품질 모델(Quality Model Division): 제품 품질 평가의 일반모델을 제시하는 부문이며 2501n의 번호를 부여
- 품질 메트릭(Quality Metrics Division): 품질 측정 메트릭에 관한 부문이며 2502n의 번호를 부여
- 품질 요구사항(Quality Requirement Division): 품질 요구사항에 관한 부문이며 2503n의 번호를 부여
- 품질 평가(Quality Evaluation Division): 품질 평가 절차에 관한 부문이며 2504n의 번호를 부여
- 품질 관리(Quality Management Division): SQuaRE 표준의 가이드라인과 품질평가의 관리에 관한 표준을 제시하고 있으며 2500n으로 표준번호를 부여

SQuaRE는 기존의 ISO/IEC 9126과 14598 표준을 합치는 과정에서 문서의 일관성을 유지하기 위해 약간 또는 상당부분 문서자체를 변경하거나 기술적으로 상당부분 개정하였으며, 다음과 같은 부분은 새로 추가되었다.

- 개요 및 가이드(General Overview and Guide to the SQuaRE): SQuaRE의 전체적인 개요와 구조(architecture) 모델에 대해 정의
- 메트릭 참조모델과 가이드(Metric Reference Model and Guide): 메트릭 디비전의 포괄적인 문서로 품질 메트릭에 대한 일반적인 정보를 제공하는 메트릭 참조모델을 정의
- 기본 메트릭(Base Metrics): 소프트웨어 개발 생명주기에서 공통적으로 사용하도록 권고되는 기본 메트릭을 정의하고 설명
- 품질 요구사항(Quality Requirement): 소프트웨어 제품 품질이 품질 요구사항으로 명시되도록 소프트웨어 품질 요구사항과 이의 추적, 타당성 및 관련성을 정의

그리고 “외부 메트릭”과 “사용 품질 메트릭(Quality in Use Metrics)”은 기술적으로 상당 부분 개정되었고, 나머지 부분은 표준 문서의 일관성 유지를 위해 편집한 것이다.

3. 소프트웨어 품질 요구사항

요구사항 명세는 기능적 요구사항, 비기능적인 소프트웨어 제품 품질에 관한 국제표준화 요구사항 및 품질 요구사항을 포함한다. 품질 요구사항은 기능성(functionality) 요구사항, 신뢰성 요구사항, 사용성 요구사항 등을 포함한다. 이러한 요구사항은 소프트웨어 제품의 사용자, 판매자, 구매자 등의 스테이크홀더(stakeholders)의 요구를

반영한 것이다. 품질의 종류, 관련자의 종류, 개발 생명주기 등 여러 가지 측면에서 품질 요구사항을 정의할 수 있으며, 이를 좀 더 형식적이고 구체적으로 나타내면 측정 가능한 요구사항(measurable requirement)이 된다. 측정 가능한 요구사항은 소프트웨어 내부적인 관점(internal view), 외부적인 관점 (external view), 그리고 시스템 관점(system view)으로 나누어 볼 수 있으며, 각각의 관점들은 개발된 소프트웨어 제품, 개발된 정보시스템, 현실화된 비즈니스 시스템(realized business system)을 평가하기 위한 기초로 활용된다. 즉, 각각의 관점에서의 요구사항에 비취 소프트웨어나 시스템이 얼마나 잘 구현되었는지를 확인하는 것이 소프트웨어 제품 품질 평가를 의미한다. 다음 그림 5는 이런 관계를 잘 나타내고 있다.

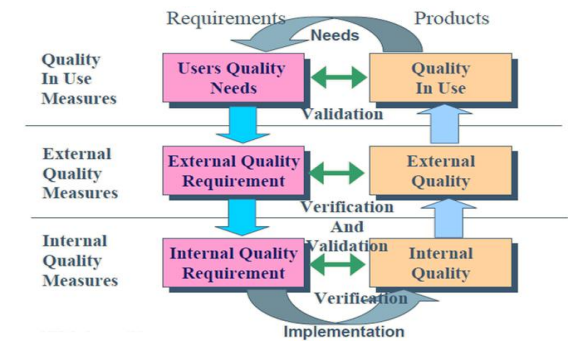


그림 5. 소프트웨어 내/외부 품질 요구사항과 평가
Figure 5. Quality requirements and Quality Measure

SQuaRE에서는 품질 요구사항이 명확하게 명시되었을 때에만 의미 있는 품질 평가가 이루어진다고 정의한다. 따라서 소프트웨어 제품 품질은 품질 요구사항으로 명시되어 정의되어야 한다. SQuaRE의 품질 요구사항 부분에 정의된 바

에 따르면, 요구사항은 획득(acquisition), 요구 분석, 개발, 사용, 평가, 지원, 유지보수, 품질 보증, 그리고 소프트웨어 감사(audit)와 관련된 서로 다른 관점에서 평가되도록 추적 가능하고, 검증되고, 관리되어야 한다. 이를 위해 SQuaRE에서는 요구사항 정의를 위한 모델과 메트릭을 사용하기 위한 가이드를 포함하고 있다.

IV. 소프트웨어 제품 품질 국내 표준화 현황

국내의 소프트웨어 품질 관리는 2000년 한국 정보통신기술협회(TTA)에서 소프트웨어 시험센터를 설치하고 소프트웨어 제품에 대한 시험평가를 수행하면서 본격적으로 소프트웨어 품질 개선을 위한 노력이 이어지고 있다. 초기 단계 소프트웨어 품질이란 인식 부족으로 인하여 소프트웨어 시험에 대한 관심이 적었으나, 10년이 지난 지금의 시점에서는 소프트웨어 개발 업체에서는 GS(Good Software) 인증을 획득하기 위한 노력을 기울이고 있다 [17]. KS 인증이 실제적인 제품 위주의 인증이라면, GS 인증은 소프트웨어 제품 자체를 대상으로 하는 유일한 국내 품질 평가 표준이라고 할 수 있다. GS 인증은 거의 모든 분야의 소프트웨어를 대상으로 TTA와 한국산업기술시험원(KTL)에서 국제표준 ISO/IEC 9126-2, 25051, 14598에 의거하여 한국형 평가모델을 개발하여 적용하고 있다.

그 외에 소프트웨어 품질보증 관련 표준으로 1994년 국립기술품질원(NITQ)에서 제정, 관리하는 소프트웨어 제품 품질 표준으로 KS X 2216 (소프트웨어 제품평가의 품질 특성 및 사용지침)이 있다. 이 표준은 소프트웨어의 품질을 기술함에 있어 중복을 최소화한 6가지 특성으로 규정하고 보다 상세한 부특성 및 메트릭, 측정방법 등

은 부속서로 처리하여 표준화 범위에서 제외하였다. 그러나 이 표준은 1991년 제정된 ISO/IEC 9126을 원안 그대로 채택하였으며, 6가지 품질특성도 ISO/IEC 9126에 정의된 바와 완전히 일치한다.

SQuaRE에 의해 2007년 발표된 ISO/IEC 25001을 채택한 KS X ISO/IEC 25001 표준이 2009년 고시되어 소프트웨어 제품 품질 요구 명세 및 소프트웨어 품질평가에 대한 계획수립과 관리를 담당하는 조직에서 해당 기술, 도구, 경험 및 관리기술 등을 준비하는데 필요한 요구사항과 권고사항을 제시하고 있다. 그러나 이 표준도 SQuaRE 표준을 그대로 채택한 것으로 표준을 제정한 후 국내의 실정에 맞춰 현지화하기위한 후속 연구와 노력이 미흡한 실정이다.

우리나라는 소프트웨어 품질평가에 대한 단일화되고 일관된 프레임워크가 없고, 국제 표준의 원안을 그대로 국내 표준으로 제정하고 있어서, 국내 소프트웨어 산업계에 대한 고려가 이루어지지 않고 있다. 또한 소프트웨어 품질 표준의 관리가 적절히 이루어지지 못하고 표준 기관간의 중복 발생에 대한 충분한 검토가 없고, 소프트웨어 분야별 품질 모델에 대한 표준화 작업이 시도되지 못하고 있다 [18]. 국내 소프트웨어 품질 평가의 문제점은 다음과 같다.

- 단일화된 소프트웨어 품질 모델 및 품질 평가 프레임워크 부재
- 국제 표준 원안을 그대로 채택하여 적용
- 소프트웨어 분야별 품질 모델 표준화에 대한 인식 부재
- 국내 소프트웨어 산업의 특성에 대한 고려 부재
- 품질 표준의 개정 등의 관리 부재
- 표준 사용자에게 대한 피드백이 전혀 없음

V. 결론

소프트웨어 품질 모델에 대한 표준화는 소프트웨어의 일반적 품질특성을 정의하는데 초점을 두고 있다. 소프트웨어의 다양한 분야에 대하여 품질 모델의 개발이 필요하지만, 아직까지 소프트웨어 분야별 품질 모델이 개발되지 못하고 있다. 소프트웨어 분야는 그동안 짧은 역사를 가지고 있는데 비하여 다양한 영역으로 발전되었고 지금도 새로운 영역이 지속적으로 생성되고 있어 각각의 영역에 대한 품질평가 모델을 개발하기가 매우 어려운 것이 사실이다.

현재까지 소프트웨어 품질에 대한 표준화 작업은 소프트웨어 제품평가 분야, 프로세스 평가 분야, 품질시스템 구축 분야에 대해서 진행되고 있으며, ISO/IEC가 이를 주도하고 있다. 소프트웨어 품질 평가 모델의 국제 표준화는 소프트웨어 제품의 일반적이고 추상적인 모델을 정립하는데 중점을 두고 있으며, 아직까지 산업계의 적용 및 피드백이 미흡한 실정이다. 국내의 소프트웨어 품질 관련 표준 역시 국제표준을 그대로 채택하여 아직까지 소프트웨어 제품의 품질을 평가할 때 제한적으로 적용할 수 있는 수준이다.

그러나 최근 TTA 등의 소프트웨어 제품 품질 인증 제도가 정착되어 산업계의 다양한 적용 경험을 가지고 있으며, 품질평가에 대한 노하우가 축적되고 있어 우리나라에서 독자적인 품질 및 평가 모델을 개발하면 산업계의 적용 과정을 거쳐 유용한 산업 표준으로 등장할 수 있는 가능성을 가지고 있다고 본다.

SQuARE 프로젝트에서 볼 수 있듯이, 현재 소프트웨어 품질에 대한 국제표준이 변화하고 있다. ISO/IEC 25000의 경우 소프트웨어 품질을 정확히 평가할 수 있는 평가 모델에 대한 요구와 함께 사용자의 요구사항을 분석하기 위한 연구도 병행해서 이루어지고 있다. 좋은 품질의 소프트

웨어는 결국 고객의 요구사항을 잘 반영한 제품이기 때문이다.

현재 “표준의 변화기”에 국내 소프트웨어 품질 관련 기술을 국제적인 수준으로 끌어올리기 위해서는 국내의 소프트웨어 산업 환경과 기술수준을 고려한 한국형 소프트웨어 품질 모델을 개발하고 적용함으로써 경험과 기술을 축적하는 것이 필요하다. 궁극적으로 고객의 요구사항을 반영 할 수 있는 다양한 연구를 통해 국제 표준에서 제시하고 있는 평가항목을 중심으로 정확한 평가를 통한 좋은 품질의 소프트웨어를 개발하는 것이 우리의 당면 과제이다. 진화하고 있는 소프트웨어 품질평가 모델을 통한 제품의 완벽한 시험평가를 앞으로 다가올 첨단 산업 사회에서 중요한 역할을 할 것으로 기대된다.

참고 문헌

- [1] ISO/IEC 9126: Information technology-Software product evaluation-Quality characteristics and the guidelines for their use, 1991.
- [2] ISO/IEC 14598-1: Software Engineering - Product evaluation (Part 1: General), 1999.
- [3] ISO/IEC 14598-2: Software Engineering - Product evaluation (Part 2: Planning and management), 1999.
- [4] ISO/IEC 14598-3: Software Engineering - Product evaluation (Part 3: Process for developers), 1999.
- [5] ISO/IEC 14598-4: Software Engineering - Product evaluation (Part 4: Process for acquirers), 1999.
- [6] ISO/IEC 14598-5: Software Engineering - Product evaluation (Part 5: Process for evaluators), 1999.
- [7] ISO/IEC 9126-1: Software engineering - Product quality, 2001.
- [8] ISO/IEC 9126-2: Software engineering -

- Product quality, External Metrics, 2003.
- [9] ISO/IEC 9126-3: Software engineering - Product quality, Internal Metrics, 2003.
- [10] ISO/IEC 9126-4: Software engineering - Product quality, Quality in use metrics, 2004.
- [11] ISO/IEC 25010: Systems and software engineering ? SQuaRE, System and software quality models, 2011.
- [12] J. McCall, P. Richard, G. Walters, "Factors in Software Quality", NTIS AD-A049-014, 015, 055, November 1977.
- [13] B. Boehm, J. Brown, H. Lipow, G. MacLeod, M. Merrit, Characteristics of software quality, Elsevier North Holland, 1978.
- [14] R. G. Dromey, "A Model for Software Product Quality", IEEE Transactions on Software Engineering, vol. 21, no. 2, pp.146-162, 1995.
- [15] R. B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice-Hall, 1992.
- [16] D. Zubrow, "Software Quality Requirements and Evaluation, the ISO 25000 Series", PSM Technical Working Group, February 2004.
- [17] 정혜정, "소프트웨어 품질 측정 표준화 동향", TTA 저널 128호, 2010년 3월.
- [18] 오영배, "소프트웨어 제품 품질 평가", TTA 저널 105호, 2006년 6월.

저 자 소 개



김 유 경(金裕卿)

1991년 2월 숙명여자대학교 수학과 (학사)
 1994년 8월 숙명여자대학교 전산학과 (석사)
 2001년 8월 숙명여자대학교 컴퓨터과학과 (박사)
 2001년 9월-2005년 8월 숙명여자대학교 정보과학부 초빙교수
 2005년 9월-2006년 8월 미국 University of California Davis, Post-doc.
 2006년 9월-현재 한양대학교 컴퓨터공학과 연구교수

<주관심분야> SW품질평가, SOA, 웹서비스



도 경 구

1980년 한양대학교 산업공학과(학사)
 1987년 미국 아이오와주립대학교 전산학과 (석사)
 1992년 미국 캔사스주립대학교 전산학과(박사)
 1993.4~1995.8, 일본 Aizu 대학 교수
 2000.9~2001.12, 스마트카드연구소 대표이사
 2005.3~2006.2, 미국 캘리포니아대학 데이비스캠퍼스 방문교수
 1995.9~현재, 한양대학교 안산캠퍼스 컴퓨터공학과 교수

<주관심 분야> 프로그래밍언어, 프로그램 분석, SW 보안, 프로그래밍언어 의미표기법