

# 소프트웨어 품질 인증을 위한 참조 모델

김우식\*<sup>0</sup> 오재원\* 윤경환\* 이종원\* 우치수\* 장우현\*\* 이숙희\*\*\*

\* 서울대학교 컴퓨터공학부

{uskim, jwoh, life, ljw, wuchisu}@sefab.snu.ac.kr

\*\* 한국전자통신연구원

whjang@etri.re.kr

\*\*\* 서경대학교 전산정보관리학과

oleesh@bukak.skuniv.ac.kr

## A Reference Model for Software Quality Certification

Woo-Sik Kim\*<sup>0</sup> Jaewon Oh\* Kyung-hwan Yoon\* Chongwon Lee\* Chisu Wu\* Woo-Hyun Jang\*\* Shookhee Lee\*\*\*

School of Computer Science and Engineering

Seoul National University

\*\* ETRI

\*\*\* Department of Computer Information Management

Seokyeong University

### 요 약

최근 소프트웨어 산업의 동향을 살펴보면 소프트웨어의 생산성 뿐만 아니라 소프트웨어의 품질에 대해서 개발자와 사용자가 그 중요성을 깊이 인식하고 있음을 알 수 있다. 이에 따라 소프트웨어 품질 인증 서비스에 대한 요구가 증가하고 있으며 품질 인증을 위한 인증 기관들이 점차 설립되고 있다. 하지만 인증 기관들이 현재 수행하고 있는 인증 방법과 체계를 일반 소프트웨어 품질 인증에 그대로 적용하는 것은 인증 대상에 제약이 가졌으며 일반 소프트웨어 품질 인증에 적합하지 않은 인증 프로세스로 인해 여러 가지 문제점들을 야기시킨다. 따라서 일반 소프트웨어 제품에 대한 인증을 적절하게 수행하기 위해서는 새로운 인증 체계가 필요하다. 본 연구는 이러한 시각에서 새로운 인증 체계의 정립을 위한 인증 모델을 제시하였다.

### 1. 서 론

현재 소프트웨어 품질 인증은 의료용 소프트웨어와 같이 높은 안정성과 가용성을 요구하는 분야에서 제한적으로 수행되고 있다.

하지만 최근의 소프트웨어의 산업 동향을 살펴보면 이러한 제한된 영역의 소프트웨어 뿐만 아니라 일반 소프트웨어의 품질에 대해서도 개발자와 소비자들의 인식이 높아졌음을 알 수 있다. 즉 소프트웨어 생산자들은 개발 활동의 생산성은 물론 품질의 중요성을 깊이 생각하게 되었고 소비자들은 또한 소프트웨어 품질에 대한 요구수준이 높아지고 있다. 그리고 이를 뒷받침 할 소프트웨어 품질 인증의 방법에 대하여 활발한 연구가 진행되고 있으며 품질 인증을 위한 인증 기관들도 점차 생겨나고 있다.

소프트웨어 품질 인증 서비스를 할 경우 개발자 측면에서는 개발 소프트웨어의 이미지 제고와 신뢰성 확보를 통해 제품의 판매 촉진을 유발할 수 있다. 또한 제품 개발 시 제품이 만족해야 하는 요구사항을 파악하여 개발 위험을 감소시킬 수 있고 기술 개발 방향을 사전에 파악할 수 있다. 소프트웨어 사용자는 진본 인증 기관에 의한 상세한 품질 인증 결과를 통해 신뢰성 있는 제품을 선택할 수 있다[1].

현재 가장 많이 쓰이는 품질 인증 방법은 개발 프로세스를 인증하는 방법이다. 하지만 개발 프로세스를 인증하는 방법은 일반 소비자가 접하는 대다수의 소프트웨어는 프로세스가 정립되지 않은 소규모 기업에서 생산하기 때문에 적합하지 않은 방법이다[2]. 따라서 소프트웨어 인증은 소프트웨어 완제품 자체를 중심으로 이루어지는 것이 바람직하다.

이러한 관점에서 본 연구는 일반 소프트웨어에 대한 신뢰성 있는 품질 인증 수행을 위해 소프트웨어 도메인(Software Domain), 품질 요소(Quality Characteristic), 소프트웨어 사용환경(Software Environment), 사용자(End User)의 네 가지 요소를 고려한 체계화된 소프트웨어 품질 인증 모델을 제시한다.

본 논문의 구성은 다음과 같다. 제2절에서는 현재 수행되고 있는

소프트웨어 품질 인증에 대한 일반적인 상황과 문제점에 대해서 기술하고, 제3절에서는 소프트웨어 품질 인증을 위한 인증 모델을 제시한다. 제4절에서는 제안된 인증 모델의 실제 적용 사례를 보이고 마지막 제5절에서는 결론과 함께 향후 연구과제를 기술한다

### 2. 소프트웨어 품질 인증 방법과 문제점

일반적으로 소프트웨어 인증을 수행하는 방법에는 세가지 방법이 존재한다. 개발 프로세스를 인증하는 방법과 개발 인력을 인증하는 방법, 소프트웨어 완제품을 인증하는 방법이다[3].

현재 일반적으로 쓰이고 있는 인증 방법으로는 SEI-CMM과 ISO 9001과 같이 개발 프로세스를 인증하는 방법과 의료용 소프트웨어나 항공 관제 시스템 등 높은 안정성을 요구하는 분야에서 사용되는 RTCA DO179-B, SAE ARP 4754, UK Defence Standard 055과 같이 개발 프로세스 인증과 소프트웨어 완제품 인증을 병합한 방법이 있다[4][5].

이 두 가지 방법은 일반 소프트웨어 품질 인증 수행에 그대로 적용되기에는 몇 가지 문제점을 가지고 있다.

우선 개발 프로세스를 인증하는 방법은 개발 프로세스를 갖추고 있는 규모 있는 기업에 적합하다. 하지만 일반 사용자가 접하는 대다수의 소프트웨어는 프로세스를 갖추고 있지 않은 소규모 기업들이 개발한 것이 대부분이므로[2] 현재 생산되는 모든 소프트웨어에 대해서 품질 인증을 수행하기에는 개발 프로세스 인증 방법은 적합하지 않다. 따라서 일반 소프트웨어 제품에 대한 품질 인증을 수행하기 위해서는 제품 중심의 인증 방식이 필요하다.

또한 일반 소프트웨어 제품을 인증하기 위해서 높은 안정성을 요구하는 소프트웨어에 쓰이고 있는 인증 방법을 그대로 일반화하여 적용할 경우 다음과 같은 문제점이 발생할 수 있다.

1) 복잡한 인증 절차 - 높은 안정성을 요구하는 소프트웨어에 대한 인증 방법은 일반 소프트웨어 제품용 인증하기에는 과도한 인증 사항과 인증 절차를 요구한다.

2) 일반화의 어려움 - 높은 안정성을 요구하는 소프트웨어에 대한 인증은 해당 분야에 특화된 인증 과정과 방법을 가지므로 일반 소프트웨어의 특징을 반영할 수 있을 정도로 일반화되기 어렵다.

3) 정보전달의 어려움 - 높은 안정성을 요구하는 소프트웨어에 대한 인증을 요청하는 사람이나 단체는 그 수가 많지 않으며 인증 기관과 밀접한 관계를 가지므로 인증 정보에 대한 정보 교환이 쉽다. 반면에 일반 소프트웨어 인증의 경우 인증을 접하는 소비자가 불특정 다수이며 인증 기관과 밀접한 관계를 가지지 않는다. 이런 이유로 인증에 대한 정보물 유통이 인증 사용자에게 전달하기 어렵다.

이와 같은 이유로 일반 소프트웨어 제품에 대해 품질 인증이 적절히 수행되기 위해서는 다양한 영역의 소프트웨어에 적용이 가능한 제품 중심의 인증 체계와 방법이 필요하다. 본 논문은 새로운 인증 체계를 정립하기 위한 기반 연구로서 인증 요구사항을 기술하고 이를 반영한 인증 모델을 제시하고자 한다.

인증 모델이란 인증 기관이 인증 방법, 인증 대상, 인증 범위 등을 결정하기 위해서 참조하는 기준을 말한다. 인증 기관은 인증 서비스 전에 인증 모델을 정립하여 이를 기반으로 인증과 관련된 전반적인 요소를 결정하게 되고 만들어진 인증 모델은 인증 프로세스 전체에서 인증기관에 의해 참조된다.

**3. 인증 모델**

일반적으로 소프트웨어 품질 인증이 갖추어야 할 요구사항은 아래와 같다.

1) 비교 가능성: 인증을 받지 않은 소프트웨어와 인증을 받은 소프트웨어가 비교 가능해야 한다. 아울러 인증 받은 소프트웨어 사이의 비교도 가능해야 된다.

2) 정보력: 인증은 소프트웨어를 사용하는 최종 소비자의 선택과 판단을 돕기 위한 것이므로 인증을 통해서 최종 소비자는 명확한 정보를 얻을 수 있어야 한다.

3) 신뢰성: 인증은 소프트웨어가 일정 수준 이상의 품질을 획득하고 있다는 것을 인증하는 것이므로 인증 받은 소프트웨어는 품질에 대해서 소비자에게 확실한 신뢰를 줄 수 있어야 한다.

인증 기관이 위의 세가지 요구사항을 만족시키며 인증을 하기 위해서는 그 전에 명확한 인증 모델을 세울 필요가 있다.

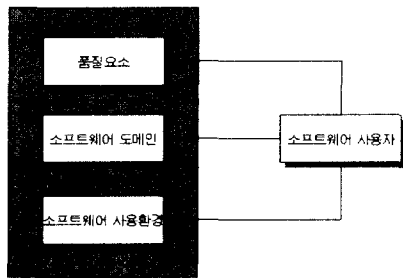


그림 1 소프트웨어 품질 인증의 네 가지 요소

앞 절에서 언급한 문제점과 소프트웨어 인증의 일반적인 요구사항을 고려하여 본 논문에서 제시하는 소프트웨어 인증 모델은 그림 1과 같으며 각 구성 요소들에 대한 설명은 다음에 나와 있다.

**3.1 소프트웨어 도메인(Software Domain)**

소프트웨어 도메인이란 인증 대상이 되는 소프트웨어가 어떠한 종류의 소프트웨어인가를 의미하는 것이다.

소프트웨어의 품질 측정 과정은 해당 소프트웨어의 종류에 따라서 달라지게 된다. 예를 들면 워드프로세서와 웹 브라우저의 성능(Performance)은 같은 품질 기준으로 평가 될 수 없다. 따라서 정확한 인증을 위해서는 소프트웨어를 분류할 필요가 있으며 이렇게 분류된 소프트웨어군마다 적절한 품질 측정 방법을 마련할 필요가 있다.

또한 소프트웨어 분류표는 소프트웨어 사용자에게 비교가 가능한 소프트웨어를 묶어 주어 인증을 받은 소프트웨어는 어떤 소프트웨어와 비교가능한지를 제시해주므로 소프트웨어 사용자의 판단을 도울 수 있다.

따라서 인증 기관은 같은 그룹으로 분류된 소프트웨어들을 동일한 품질 요소를 기준으로 하여 비교가 가능하도록 분류체계를 세워야 한다.

**3.2 품질 요소(Quality Characteristic)**

소프트웨어의 품질은 단일하게 나타나지 않고 기능성(Functionality), 효율성(Efficiency), 편의성(Usability) 등 여러 가지 하부 요소의 결합으로 나타내게 되는데 소프트웨어 품질을 이루는 이 하부 구성 요소를 품질 요소라고 한다[6][7].



그림 2 ISO 9126 품질 모델

그림 2는 ISO 9126 품질 모델에 따른 여섯 가지 품질 요소를 제시하고 있다[6]. 소프트웨어의 품질을 어떤 시각으로 보는가에 따라 여러 가지 품질 모델이 있을 수 있으며 품질 모델이 달라짐에 따라 품질 요소도 달라진다.

소프트웨어 인증 작업은 기본적으로 그림 2와 같은 품질 모델을 기반으로 하여 품질 요소들을 측정함으로써 이루어 진다. 하지만 소프트웨어의 품질 인증 작업은 품질 요소들 전체를 측정하고 종합함으로써 이루어 지는 것은 적절하지 못하다. 품질 인증은 품질 요소 단위로 수행하는 것이 바람직하다.

그렇지만 실제로 지금까지 진행된 많은 연구들은 하위 품질 요소들을 종합하여 단일한 값을 얻기 위해서 노력하였다[8][9]. 하지만 소프트웨어의 품질을 단일 결과로 표현하거나 소프트웨어 전체 품질에 대해서 품질 인증을 수행하는 것은 바람직하지 못하다. 그 이유는 다음과 같다.

- 1) 개별 품질 요소들을 정보 손실 없이 종합하는 것이 쉽지 않다.
- 2) 단일 인증 방식으로 소프트웨어의 품질 인증이 이루어진 경우 사용자는 인증 받은 소프트웨어의 다양한 하위 품질 요소에 대한 내용을 잘 알지 못하므로 품질 인증 내용을 잘 파악 할 수 없다.
- 3) 최종 사용자 마다 소프트웨어 품질에 대한 기대 수준이 다르다. 왜냐하면 사용자마다 소프트웨어의 품질 요소에 대한 우선 순위가 다르기 때문이다. 따라서 인증 기관에서 정한 일률적인 품질 요소 종합 기준을 모든 최종 소프트웨어 사용자에게 강요하는 것은 바람직하지 못하다.

따라서 편의성(Usability)에 대한 인증, 가용성(availability)에 대한 인증, 성능(Performance)에 대한 인증과 같이 소프트웨어 품질 인증은 최종 사용자가 이해 가능한 소프트웨어 개별 품질 요소들에 대해서 단위별 인증을 수행하여야 한다.

**3.3 소프트웨어의 사용환경(Software Environment)**

소프트웨어의 품질 인증은 인증 대상 소프트웨어의 실제 사용환경을 고려해야 한다. 여기서 환경이란 단순한 플랫폼을 지칭하지 않는다. 플랫폼 품질을 포함하여 소프트웨어를 사용하는 사용자 층, 사용자의 안정성 요구 정도 등 실제로 소프트웨어가 사용되는 영역의 모든 주변 요소를 포함하는 개념이다.

일반 소프트웨어에 내장되는 것을 가정하여 안정성에 대한 인증을 받은 컴포넌트가 의료 시스템이나 항공 시스템과 같이 높은 안정성을 요구하는 시스템의 컴포넌트로 쓰이는 경우는 이전에 받은 안정성 인증은 더 이상 유효하지 않다. 왜냐하면 그 사용환경에 따라 안정성에 대한 최종 사용자들의 요구 수준과 개념이 다르기 때문이다.

따라서 인증 기관은 소프트웨어가 어떠한 소프트웨어 환경을 가정하여 인증을 획득했는지 명확히 밝혀서 소비자가 인증의 유효 범위와 경계를 알 수 있도록 해야 한다.

**3.4 소프트웨어 사용자(End User)**

인증 모델의 마지막을 이루는 요소는 소프트웨어의 최종 사용자이다. 인증 모델 속에 최종 사용자가 포함된 것은 위에서 언급한 세 가지 요소와 인증 프로세스를 결합하는 과정에서 소프트웨어 최종 사용자가 중심이 되어야 함을 의미한다.

소프트웨어 인증 작업은 품질 관리를 위한 개발 과정 내의 작업이 아니고 소프트웨어 일반 사용자의 소프트웨어의 선택과 소프트웨어에 대한 평가를 돕기 위한 것이다. 따라서 모든 과정이 사용자 위주로 이루어져야 한다[4].

인증 기관은 인증 프로세스와 품질 측정 과정에서 최종 사용자의 의사와 실제 사용 프로파일을 최대한 반영 할 수 있도록 해야 하며, 소프트웨어 도메인과 품질 요소를 결정 할 때도 일반 소프트웨어 사용자가 이해 가능하도록 결정해야 한다. 또한 소프트웨어 사용환경을 결정하는 것도 실제 사용자의 환경을 반영하도록 해야 한다.

**4. 인증 모델의 적용 예**

인증 모델을 세우는 작업은 제안된 소프트웨어 도메인, 품질 요소, 소프트웨어 사용환경의 분류 기준을 정립하는 작업과 인증 프로세스에서 최종 사용자의 참여 방법을 정하는 일이다.

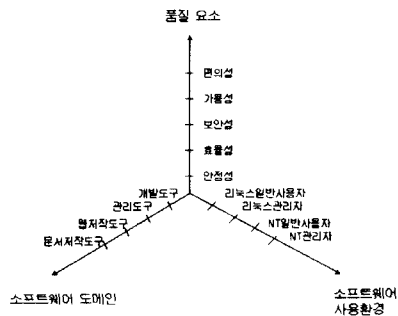


그림 3 인증 모델 적용 예

그림 3은 인증 모델 적용 예이다. 실제 인증 기관에서는 위 그래프 상의 점 단위로 인증 서비스를 수행할 것이다. 위 그래프에서 한 점이 어느 한 축으로 움직이는 경우는 새로운 인증 작업이 필요함을 의미한다. 예를 들면 소프트웨어가 [개발도구, 안정성, 리눅스 일반 사용자]로 표시된 점에서 인증을 받았다고 해도 [개발도구, 안정성, 리눅스 관리자] 관점에서는 새로운 인증이 필요하다. 앞에서 언급한 바와 같이 소프트웨어 사용환경이 변경된 경우에는 이전 인증이 더 이상 유효하지 않기 때문이다.

인증 모델을 정립하는 작업은 인증기관의 특성과 목표에 따라 달라질 수 있을 것이다. 예를 들면 안정성 위주로 인증을 수행하는 기관이라면 위 예에서는 안정성 품질 요소만 남겨 두고 다른 품질 요소는 없는 간단한 모델을 사용하거나, 품질 요소 축을 안정성 품질 요소를 구체적으로

세분화한 축으로 새로이 구성된 모델을 사용할 수 있을 것이다.

**5. 결론 및 향후 과제**

인증의 의미는 인증된 소프트웨어가 모든 품질 요소에 대해 기대하는 품질 수준을 만족하고 모든 소프트웨어 사용환경에서 사용이 가능하다는 것은 아니다. 즉 특정한 소프트웨어 환경에서 사용할 경우 해당 소프트웨어가 특정한 품질 요소에 대해 만족할 만한 수준을 보인다는 것을 의미한다. 그리고 소프트웨어군 별로 품질 요구사항 및 측정 방법이 달라질 수 있다. 따라서 소프트웨어를 분류하는 작업이 우선 이루어져야 하며 이러한 분류를 기반으로 하여 인증을 시행하여야 한다. 또한 정확한 인증을 위해서는 인증 모델과 인증에 관련된 모든 절차와 방법은 사용자 중심의 작업이 되어야 한다. 이를 위해서 본 논문에서는 위 같은 요구 사항들을 반영한 소프트웨어 인증 모델을 제시하였다.

향후 과제로는 소프트웨어와 사용환경에 대한 체계적인 분류 방법을 세우는 일이며 이를 바탕으로 궁극적으로는 인증 기관에서 일반적으로 사용 가능한 인증 모델 템플릿을 개발하는 일이다.

**참고 문헌**

- [1] Jeffrey M. Voas, "Certification: Reducing the hidden costs of poor quality," IEEE Software, Vol. 16, No. 4, 1999.
- [2] William T. Council, "Third-Party Testing and the Quality of Software Components," IEEE Software, Vol. 16, No. 4, 1999.
- [3] Jeffrey M. Voas, "The Software Quality Certification Triangle," Crosstalk, pp.12-14, Nov. 1998.
- [4] Jeffrey M. Voas, "Developing a Usage-Based Software Certification Process," IEEE Computer, Vol. 33, No. 8, 2000.
- [5] Patricia Rodriguez-Dapena, "Software Safety Certification: A Multidomain Problem," IEEE Software, Vol. 16, No. 4, 1999.
- [6] ISO/IEC, "FCD 9126-1.2 Information Technology - Software Product Quality - Part 1: Quality Model," ISO/IEC JTC1/SC7 N1949, 1998.
- [7] Roger S. Pressman, "Software Engineering: A Practitioner's Approach," 5th Ed., pp. 507-538, McGraw-Hill, 2001.
- [8] Gary Johnson and Xingbo Yu, "Objective Software Quality Assessment," Proceedings of the 1999 IEEE Nuclear Science Symposium Conference, Vol. 3, 1999.
- [9] Renu Kumar, Suresh Rai, and Jerry L. Trahan, "Neural - Network Technique for Software Quality Evaluation," Proceedings of annual Reliability and Maintainability Symposium, 1998.