

## 대학생 대상 소프트웨어역량 진단 도구 타당화 연구\*

## Validating a Diagnostic Tool for Assessing Software Competency in College Students

이윤선<sup>†</sup> · 김은수<sup>††</sup> · 이현숙<sup>††</sup> · 이세희<sup>†††</sup>Yoonsun Lee<sup>†</sup> · Eunsoo Kim<sup>††</sup> · Hyunsook Lee<sup>††</sup> · Sehee Lee<sup>†††</sup>

## 요약

본 연구의 목적은 고등교육에 확산되고 있는 소프트웨어 교육의 효과성을 확인할 수 있는 소프트웨어 역량 진단도구를 개발하고 이를 타당화하는 것이다. 이를 위해 선행연구와 대학교육과정을 토대로 소프트웨어 역량의 기본개념과 구성요소를 도출하고 이를 기반으로 검사 문항을 개발하여 전문가 검토를 통해 내용타당도 검증 완료하였다. 문항분석과 구인타당도 분석결과, 24개의 전공역량 문항과 37개의 일반역량이 확정되었다. 구체적으로, 전공역량은 컴퓨팅 사고력과 프로그래밍 능력, 데이터 분석역량, 창의적 문제해결 역량, 소프트웨어 융합역량이, 일반역량은 자기관리역량, 갈등해결역량, 협업역량, 현장적응역량, 상호문화이해역량, 소프트웨어 공학윤리역량으로 확인되었다. 개발된 소프트웨어 역량은 5점 리커트척도가 적절한 것으로 나타났고 피험자 능력과 문항난이도의 분포도 양호한 것으로 나타났다. 마지막으로 본 진단도구는 전공역량과 일반역량 모두 .95로 높은 신뢰도를 보였다. 향후 본 검사는 대학생의 소프트웨어 역량 수준을 확인하고 관련 교과목의 효과성을 파악하는 데 활용할 수 있으며 그 결과를 토대로 소프트웨어 교육 질 관리에 유용한 정보를 제공할 수 있을 것으로 기대한다.

**주제어:** 소프트웨어 교육, 소프트웨어 역량, 소프트웨어 교육 질 관리

## ABSTRACT

This study aims to develop and validate a software competency diagnostic tool that can assess the effectiveness of software education in higher education. First, based on prior research and university curricula, the fundamental concepts and components of software competency were derived and pilot items were developed. The items were validated for content validity through expert review. Through item analysis and construct validity verification, 24 items for major related competencies and 37 items for general competencies were finalized. Specifically, software major related competencies include computing mindset and programming skills, data analysis competency, creative problem-solving skills, and software integration competency. General competencies consist of self-management skills, conflict resolution skills, collaboration skills, field adaptation skills, intercultural understanding skills, and software engineering ethics competency. In addition, a 5-point Likert scale appeared appropriate for this tool, and the distribution of subjects' abilities and item difficulty was satisfactory. Lastly, the reliability of this diagnostic tool was high, with a Cronbach's alpha of .95 for both major and general competencies. In the future, this assessment can be utilized to gauge the software competency levels of university students and assess the effectiveness of related courses. The results are expected to provide valuable information for quality control in software education.

**Keywords:** Software Education, Software Competency, Quality Control in Software Education

<sup>†</sup>중신회원: 서울여자대학교 교육심리학과 교수

<sup>††</sup>정회원: 서울여자대학교 교육심리학과 박사수료

<sup>†††</sup>정회원: 서울여자교육대학교 교육심리학과 석사(교신저자)

논문투고: 2023년 12월 20일, 심사완료: 2024년 02월 23일, 게재확정: 2024년 02월 28일

\* 본 논문은 2022학년도 서울여자대학교 연구년수혜 및 2024 교내연구비 지원을 받았음

## 1. 서론

4차 산업혁명 시대가 도래하면서 고등교육도 다양한 전공 분야에 걸쳐 시대가 요구하는 핵심역량을 강화하는 방향으로 진화하고 있다. 특히, 4차 산업혁명을 경계가 없어지고 융합되는 기술 혁명으로 정의하며[1], “연결과 융합”이라는 키워드가 부각되면서 빅데이터, 인공지능, 사물인터넷, 가상현실과 같은 핵심 기술들이 각 분야에 새로운 가능성을 제시하고 있다. 이에 따라, 공과대학에서 시대 변화를 주도하는 교육과정 개발 및 교육 질 관리는 이전보다 더 중요한 과제로 부상하였다.

공학 교육의 질 관리는 공학 전문가의 역량을 향상시키기 위한 교육의 효과성을 검증하고, 교육자원 및 과정을 체계적으로 분석하며, 분석결과를 교육내용에 환류하는 시스템 구축을 목표로 한다. 공학 교육의 질 관리에서 중요한 요소 중 하나는 사회 변화와 함께 개편되고 있는 교육과정을 체계적으로 분석하는 일이다. 예를 들면, 1990년대 컴퓨터 활용을 강조한 컴퓨터 교양교육은 현재 소프트웨어 영역까지 교육의 범위를 확대하고 있다[2, 3]. 대학뿐만 아니라 초·중·고 교육에서도 소프트웨어 교육은 강조되어 최근 교육부가 주최하고 대학이 초·중·고 학생들을 대상으로 제공한 디지털 새(New)싹(Software · AI Camp) 사업은 소프트웨어 교육의 중요성을 확인할 수 있는 사례이다.

이러한 환경적 요구와 변화를 반영하여 본 연구에서는 소프트웨어 교육의 질 관리 시스템을 구축하기 위한 첫 단계로 소프트웨어 역량의 명확한 개념 정의와 그 역량을 측정하는 소프트웨어 역량 진단 도구를 개발하였다. 이를 위해 국내외 소프트웨어 역량에 대한 연구와 2015년부터 국내대학에서 시작된 소프트웨어 중심대학의 교육과정을 참고하여 소프트웨어 역량의 정의와 하위 구성요소를 선정하였다. 또한 도출된 정의와 구성요소를 측정할 수 있는 예비문항을 개발하고 신뢰도와 타당도를 검증하여 대학생들의 소프트웨어 역량 수준을 확인할 수 있는 도구를 표준화하였다. 개발된 도구는 소프트웨어 교육의 효과성을 검증하여 대학 소프트웨어 교육방향 및 인공지능교육 설계에 필요한 기초자료를 제공하는 데 활용될 것이다.

## 2. 관련연구

### 2.1. 소프트웨어 역량의 개념 및 구성요소

소프트웨어의 사전적 정의는 컴퓨터 프로그램 및 그와 관련된 문서들을 통틀어 이르는 것으로, 소프트웨어는 컴퓨터를 관리하는 시스템 프로그램과 문제 해결에 이용되는 다양한 형태의 응용 프로그램으로 분류된다[4]. 소프트웨어 역량은 유능한 소프트웨어 전문가가 되기 위해 필요한 역량으로 현장에서 일상적으로 쓰이는 용어이나 학계에서는 그 정의와 하위 구인이 다양하게 제시되고 있다. 본 연구에서는 국내외 연구결과에서 보고된 소프트웨어 역량 관련 하위 구인 및 소프트웨어 엔지니어의 필수 역량으로 대학생에게 필요한 소프트웨어 역량의 정의와 구성요소를 도출하였다.

Turley와 Bieman[5]은 20명의 소프트웨어 엔지니어를 인터뷰하여 38개의 역량을 찾고 이를 크게 네 가지로 분류하였다. 네 가지로 분류된 소프트웨어 역량은 과제수행역량(Task accomplishment), 개인역량(Personal attributes), 상황별역량(Situational skills competencies), 대인관계역량(Interpersonal skills competencies)이다.

소프트웨어 엔지니어를 위한 역량 체계를 기술적(Technical), 사회적(Social), 개인적(Personal)으로 구분한 Rivera-Ibarra 외[6]에서도 기술적 영역에서는 분석 및 학습능력에 대한 요인, 사회적 영역에서는 대인관계, 팀워크와 협력, 갈등처리와 해결에 대한 요인, 개인적 영역에서는 작업 환경에서의 능력, 개인의 성장, 권리와 범위로 설명하면서 대인관계역량과 개인역량을 강조하였다.

실무역량(Professional Practice)과 관련하여 ACM(Association for Computing Machinery)과 IEEE[7]에서는 소프트웨어공학 표준 커리큘럼 지침을 제시하였다. 이 지침에서는 소프트웨어역량을 소프트웨어 엔지니어가 전문가 수준에서 책임감 있고 윤리적으로 수행하기 위해 반드시 가져야 하는 지식, 기술, 태도로 정의하였다. 구체적으로, 하위 역량으로 집단 역학 및 심리, 의사소통 기술, 전문적 역량을 제시하고 있으며, 집단 역학 및 심리 역량에서는 개인, 팀, 집단에서의 인지능력, 불확실성과 모호성에 대한 처리, 다문화 환경에서의 대처능력, 개인 역량과 관련된 의사소통 기술에서는 팀과 집단에서의 소통능력, 발표기술에 대한 요인을 구성 요인으로 보았다. 전문적 역량에서는 윤리강령과 전문직업인으로서의 행동적 측면에서의 역량을 제시하였다.

IEEE[8]에서는 소프트웨어 집약적 시스템 개발 및 수정에 참여하는 소프트웨어 엔지니어를 위한 역량으

로 SWECOM(Software Engineering Competency Model)을 제시하고 있는데, 이 모델에서는 인지 기술, 행동 특성과 기술, 기술력, 필수 지식, 관련 분야로 엔지니어가 지녀야 할 역량을 구분하고 있다. 또한, 행동 특성 및 기술과 관련된 역량에서 인증, 종교, 성적 지향 관련 문화적 감수성과 팀과 집단에서의 의사소통 기술과 더불어 리더십 기술을 주요 구성 요인으로 선정하여 실무 역량에서의 개인 역량의 특성을 세부적으로 논의하였다.

위에서 제시된 사례는 엔지니어에게 요구되는 역량을 제시한 것으로 고등교육 교육과정에서도 해당 역량에 대한 관심과 지원방안을 고민할 필요가 있다. 대학생 대상 소프트웨어 교육과 관련된 국내 연구를 살펴보면, 주이화와 주익현[9]은 대학생 대상 코딩교육에 대한 인식 유형을 제시하고 유형별 특성을 분석하였다. 연구 결과에서 인식 유형별로 코딩교육에 대한 중요성과 확대여부에는 차이를 보이고 있었으나 복잡한 프로그래밍 언어에 초점을 맞추는 코딩교육보다 코딩식 사고방식에 근거한 문제해결력과 업무자동화를 위한 수업의 필요성을 강조하였다. 대학 소프트웨어 기초 교육 효과성 모형을 개발한 홍성연 외[10]의 연구에는 컴퓨팅 사고력, 소프트웨어 문해력, 소프트웨어 인식 및 태도변화라는 세 영역을 제시하여 주요 활동을 제시하고 있다. 각 영역에는 소프트웨어 전문성을 강조하는 알고리즘, 패턴인식, 정보활용 및 미래 기술 이해, 정보 보안 및 윤리 등과 개인적 사회적 차원에서 소프트웨어 교육의 필요성 인식, 흥미, 효능감 등을 강조한 소프트웨어에 대한 인식 및 태도를 포함하고 있다. 이러한 결과는 소프트웨어 교육에서 전공 관련 능력과 함께 정의적 특성인 태도 관련 역량도 강조되고 있음을 확인할 수 있다. 박금주[11]의 연구에서도 소프트웨어 교육에서 창의적 사고력을 증진하는 프로그램에 실습이 병행되었을 때 교육의 효과성이 나타나고 있음을 알 수 있다. 구체적으로 디자인 싱킹과 소그룹 활동기반 소프트웨어 교육이 문제해결력, 의사소통, 협업능력을 향상할 수 있다고 보고하고 있다.

대학생을 대상으로 한 소프트웨어 교육 및 역량에 대한 연구와 함께 초·중·고 학생들을 대상으로도 2015 개정 교육과정 이후 소프트웨어 역량의 하위 구인인 컴퓨팅 사고력 관련 소프트웨어 교육프로그램 효과성에 관한 연구와 소프트웨어 개발과 관련된 코딩기반 메이킹 역량에 초점을 맞춘 연구들이 보고되었다. 이재호와 장준형[12]의 연구에서는 메이커들이

활동 과정에서 발휘해야 할 역량인 소프트웨어 코딩기반 메이킹 역량을 제시하고 있는데, 문제 발견과 이해, 자료수집과 분석에 필요한 역량을 분석역량 (Analytical Competency)으로, 문제해결을 위해 다양한 방법을 찾고 해결책을 제안하고 평가하여 개선 사항을 표현하는 역량을 설계역량(Design Competency)으로 구분하였다. 또한, 해결책 구현과 구현된 소프트웨어 결과물을 평가하고 실행하기 위한 역량을 구현역량(Implementation Competency)으로 설명하였다. 강성원[13]은 소프트웨어 개발 역량 증진을 위한 중등교육 방향에 대한 연구에서 소프트 콘텐츠 역량과 소프트웨어 개발 역량을 모두 일컫는 개념으로 소프트웨어 역량을 정의하였고, 소프트 콘텐츠를 만드는 능력인 소프트 콘텐츠 역량과 소프트 콘텐츠를 소프트웨어에 담는 능력인 소프트웨어 개발 역량을 소프트웨어 역량의 구인으로 설명하였다. 고등학생의 소프트웨어 역량 증진 행동을 탐구한 임경희와 김미량[14]의 연구에서는 컴퓨터 사고력과 협력적 문제해결력을 기반으로 소프트웨어 문제를 이해하며 코딩까지 완성하는 종합적인 문제해결 능력이라고 소프트웨어 역량을 정의하면서 하위 구인으로 컴퓨팅 사고력, 협력적 문제해결력을 제시하였다.

이와 같이 엔지니어에게 요구되는 역량을 정리한 국외사례와 중등교육과 고등교육에서 보고된 소프트웨어 관련 국내 연구는 소프트웨어 교육의 특성, 소프트웨어 역량의 개념, 하위 구인 등을 학자마다 다양하게 정의되어 분류되고 있다. 이러한 맥락을 종합해 보면 소프트웨어 역량의 다양한 개념을 통합적으로 해석하여 정의하고 하위 구인의 체계적 분류를 통해 표준화된 소프트웨어 역량 진단 도구의 개발과 활용이 필요하다는 것을 알 수 있다. 대학생을 대상으로 한 소프트웨어 역량 평가와 진단을 위한 검사 도구는 현재 개발되어 있지 않으므로 본 연구에서는 국내외 선행연구를 분석하여 대학생에게 요구되는 소프트웨어 역량의 개념과 구성요소를 정리하고, 이를 기반으로 소프트웨어 역량 진단도구를 개발하고자 한다. 국내외 연구자들이 제시한 소프트웨어 역량의 개념과 하위요소는 Table 1과 같다.

**Table 1.** Summary of Previous Research for Software Competency

Researchers	Competencies
Turley & Bieman(1995)	Task accomplishment • Leverages / reuses / code

Researchers	Competencies
	<ul style="list-style-type: none"> <li>• Uses methodical problem solving</li> <li>• Mastery of skills and techniques</li> <li>• Writes / automates tests with code</li> <li>• Prior experience</li> <li>• Obtains necessary training / learning</li> <li>• Uses of new methods or tools</li> <li>• Schedules and estimates well</li> <li>• Use of prototypes to assess design</li> <li>• Possesses unique domain knowledge</li> <li>• Uses structured techniques for communications</li> </ul>
	<p>Personal attributes</p> <ul style="list-style-type: none"> <li>• Driven by desire to contribute</li> <li>• Pride in quality and productivity</li> <li>• Sense of fun</li> <li>• Lack of ego</li> <li>• Perseverance</li> <li>• Desire to improve things</li> <li>• Proactive / initiator / driver</li> <li>• Maintains “big picture” view</li> <li>• Desire to do / bias for action</li> <li>• Thoroughness – methodical, organized, cautious</li> <li>• Driven by sense of mission</li> <li>• Exhibits and articulates strong convictions</li> <li>• Mixes personal and work goals</li> <li>• Proactive skills competencies</li> </ul>
	<p>Situational skills competencies</p> <ul style="list-style-type: none"> <li>• Concerns for reliability and quality</li> <li>• Focus on user or customer needs</li> <li>• Thinking – strong analytic skills</li> <li>• Emphasizes elegant and simple solutions</li> <li>• Innovation</li> <li>• Attention to detail</li> <li>• Design style</li> <li>• Responds to schedule pressure by sacrificing parts of design process</li> </ul>
	<p>Interpersonal skills competencies</p> <ul style="list-style-type: none"> <li>• Seeks help</li> <li>• Team oriented</li> <li>• Help others</li> <li>• Willingness to confront others</li> </ul>
Rivera-Ibarra et al.(2010)	<p>Technical</p> <ul style="list-style-type: none"> <li>• Technical knowledge</li> <li>• Use of technology</li> </ul>
	<p>Social</p> <ul style="list-style-type: none"> <li>• Interpersonal relations</li> <li>• Cooperation and work in a team</li> <li>• Handling and solving conflicts</li> </ul>
	<p>Personal</p> <ul style="list-style-type: none"> <li>• Development in the job environment</li> <li>• Personal Development</li> <li>• Rights and limits</li> </ul>
ACM & IEEE(2015)	<p>Group dynamics and psychology</p> <ul style="list-style-type: none"> <li>• Dynamics of working in teams and groups</li> <li>• Individual cognition (e.g., limits)</li> <li>• Cognitive problem complexity</li> <li>• Interacting with stakeholders</li> <li>• Dealing with uncertainty and ambiguity</li> <li>• Dealing with multicultural environments</li> </ul> <p>Communication skills</p> <ul style="list-style-type: none"> <li>• Reading, understanding, and summarizing reading</li> </ul>

Researchers	Competencies
	<ul style="list-style-type: none"> <li>• Writing</li> <li>• Team and group communication</li> </ul> <p>Professionalism</p> <ul style="list-style-type: none"> <li>• Accreditation, certification, and licensing</li> <li>• Codes of ethics and professional conduct</li> <li>• Social, legal, historical, and professional issues and concerns</li> <li>• The nature and role of professional societies</li> <li>• The nature and role of software engineering standards</li> <li>• The economic impact of software</li> <li>• Employment contracts</li> </ul>
IEEE(2014)	<p>Cognitive skills</p> <ul style="list-style-type: none"> <li>• Reasoning</li> <li>• Analytical skills</li> <li>• Problem-solving</li> <li>• Innovation</li> </ul> <p>Behavioral attributes and skills</p> <ul style="list-style-type: none"> <li>• Aptitude</li> <li>• Initiative</li> <li>• enthusiasm</li> <li>• Work ethic</li> <li>• Willingness</li> <li>• Trustworthiness</li> <li>• Cultural sensitivity</li> <li>• Communication skills</li> <li>• Team participation skills</li> <li>• Technical leadership skills</li> </ul> <p>Technical skills</p> <p>Requisite knowledge</p> <p>Related disciplines</p>
Lee & Jang(2017)	<p>Analytical Competency</p> <ul style="list-style-type: none"> <li>• Problem Identification</li> <li>• Problem Understanding</li> <li>• Data Collection</li> <li>• Data Analysis</li> </ul> <p>Design Competency</p> <ul style="list-style-type: none"> <li>• Solution Proposal</li> <li>• Solution Evaluation</li> <li>• Solution Improvement</li> <li>• Solution Expression</li> </ul> <p>Implementation Competency</p> <ul style="list-style-type: none"> <li>• Tool Selection Ability, Resource Utilization Ability, Resource Management Ability</li> <li>• Software Coding</li> <li>• Software Outcome Evaluation</li> <li>• Software Coding Execution</li> </ul>
Kang(2017)	<p>Software content competency</p> <p>Software Development competency</p>
Lim & Kim(2020)	<p>Computational Thinking</p> <ul style="list-style-type: none"> <li>• Utilizing basic concepts and principles of computer science and computing systems</li> <li>• Understanding problems in real life and various academic fields</li> <li>• Creatively implementing and applying solutions</li> </ul> <p>Collaborative Problem-Solving</p> <ul style="list-style-type: none"> <li>• Sharing diverse knowledge and learning communities in a networked computing environment</li> <li>• Efficient communication and collaboration for creative problem-solving</li> </ul>

## 2.2 대학의 소프트웨어 교육

현재 대학생의 소프트웨어 역량을 측정할 수 있는 진단도구는 개발되어 있지 않은 상황이다. 유수진 외 [15]가 컴퓨팅 사고력, 문제해결력, 인공지능 역량 등 구체적인 역량명을 가지고 소프트웨어 교육의 효과성을 검증하는 경우는 있으나, 소프트웨어 역량을 종합적이고 융합적인 특성으로 접근하여 도구를 개발한 경우는 없다. 따라서 소프트웨어 역량 진단 도구 개발에 앞서 국내 대학에서 현재 진행되고 있는 소프트웨어 교육 현황을 살펴보는 것은 중요하다. 이원주[16]는 해외대학, 국내 소프트웨어 중심대학, 거점 국립대학의 소프트웨어 교육과정을 비교 분석하여 효과적인 소프트웨어 교육을 제안하는 연구를 수행하였다. 연구결과를 요약하면, 대학의 소프트웨어 교육은 직무 분석을 통해 산업체 현장 적응력을 높일 수 있는 교과목개발, 4차 산업혁명의 핵심기술 및 다양한 분야와의 융합, 소프트웨어 융합 교과목에서 프로젝트 구현을 통한 프로그래밍 언어교육, 자기 주도적 학습이 가능한 프로젝트 기반 교과목 강화, 인턴십과 산학연계 프로그램 강화가 필요하다고 제안하였다. 장수미[17]는 비전공생을 위한 AI-소프트웨어 교육의 필요성을 강조하였고, 교양 교육에서 AI 및 소프트웨어 교육의 체계를 확립하고 교육의 효과성을 검증할 것을 제안하였다. 소프트웨어 중심대학 지원 사업에 선정된 대학을 대상으로 소프트웨어 기초교육 내용을 분석한 장은실과 김재현[18]은 비전공자를 대상으로 많은 대학이 컴퓨팅 사고력 중심의 문제해결력 향상과 융합 능력 향상을 목적으로 소프트웨어 교육을 진행하고 있다고 보고하였다. 교수법으로는 프로젝트 중심이나 문제중심 학습을 통해 의사소통 능력과 협업능력을 함께 강조하고 있는 것으로 나타났다. 결국, 대학생에게 요구되는 소프트웨어 역량은 교양 및 전공 교과목 통해 얻어질 수 있는 결과물로 전공 지식 능력과 함께 인턴십, 산학협력, 프로젝트 수행 등에 필요한 태도 기반 일반역량을 함께 함양해야 한다는 것을 간접적으로 확인할 수 있다.

## 3. 연구절차

본 연구는 대학생의 소프트웨어 역량을 측정할 수 있는 진단도구를 개발하고 타당화하는 것으로 연구절차는 Figure 1과 같다.

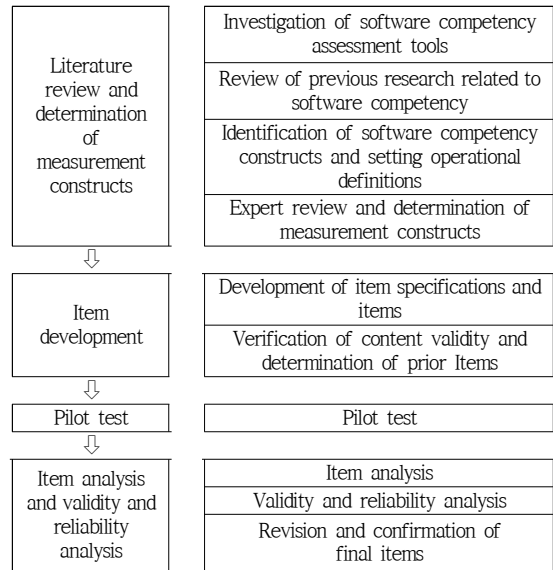


Figure 1. Study Procedure

### 3.1 문헌연구와 측정구인 및 예비문항 개발

소프트웨어 역량에 대한 개념 및 측정구인을 확인하기 위하여 관련 선행연구와 기존에 개발된 대학생 핵심역량 관련 검사들을 검토·분석하였다. 선행연구 분석을 토대로 본 연구에서는 소프트웨어역량을 ‘유능한 소프트웨어 전문가가 되기 위하여 대학생들이 갖춰야 할 지식, 태도, 기술’로 정의하였다. 또한, 소프트웨어 역량은 소프트웨어 교육과 훈련을 통해 얻어지는 지식 및 기술 차원의 전공역량과 업무 상황에서 전반적으로 요구되는 일반역량으로 구분되는 것을 확인하였다. 전공역량은 소프트웨어 전문 분야에서 특수하게 요구되는 지식 관련 역량이며, 일반역량은 전문성을 발휘하기 위한 태도와 관련된 역량이다. 추가적으로, 전공역량의 하위 구인을 확인하기 위해 국내 소프트웨어 관련 학과의 교육목표 및 커리큘럼을 참고하였고 일반역량은 공학 계열 대학생의 핵심역량 연구를 검토하여 하위구인을 설정하였다.

문헌조사를 통해 도출된 구인과 조작적 정의의 타당성을 확보하기 위해, 현재 대학에서 소프트웨어 교육을 담당하고 있는 8인의 소프트웨어 전문가들을 대상으로 서면 자문을 진행하였다. 참여한 전문가들은 소프트웨어 교육경력이 최소 10명 이상이거나 교육경력이 낮은 경우에는 10년 이상의 산업체 경력을 보유한 전문가로 선정되었다. 또한 이들은 소프트웨어 중심대학 사업단에서 다양한 소프트웨어 교육 및 AI교

육을 진행한 경험이 있다. 전문가 집단에게 배부된 자문서류는 전공역량과 일반역량으로 구분된 소프트웨어 역량 개념 모형에 대한 의견과 각 측정구인의 중요성을 5점 리커트 척도로 묻는 내용으로 구성하였다. 수집된 검토의견을 바탕으로 측정구인 및 조작적 정의를 수정 보완하였다.

전문가 검토 이후, 일차적으로 전공역량의 구인은 수학·통계학에 대한 이해, 프로그래밍 능력, 데이터 처리 및 분석 능력, 컴퓨팅 사고력, 창의적 문제해결 능력, 소프트웨어 응용 분야에 대한 지식과 기술로 구분하여 26개의 예비문항을 개발하였다. 일반역량 구인은 자기관리역량, 의사소통역량, 갈등해결역량, 초협업역량, 현장 적용역량, 상호문화이해역량, 소프트웨어 공학 윤리역량으로 정의하고 예비문항 42문항을 개발하였다. 문항은 측정구인의 조작적 정의를 바탕으로 구인별로 3~7개를 개발하였으며 응답양식은 5점 리커트 척도로 구성하였다.

### 3.2 내용타당도 검증 및 예비문항 결정

개발된 예비문항의 타당도를 검증하기 위해 내용타당도 검증을 총 2회 실시하였다. 1차 내용타당도는 3인의 소프트웨어 전문가로부터 문항의 타당도를 5점 리커트 척도로 진행하였다. 2차 내용타당도는 소프트웨어 전문가 2인으로부터 1차 내용타당도 결과반영에 따른 확인절차와 더불어 각 영역에 포함된 문항에 대한 용어 적절성 및 타당성을 검증받았다. 더불어, 검사 대상자인 대학생 3인을 대상으로 안면타당도 검증을 실시하였고, 모든 문항의 이해도가 높은 것으로 나타나 문항수정은 진행하지 않았다. 내용타당도와 안면타당도 검증 결과를 바탕으로, 각 측정구인의 문항을 수정·보완하여 예비검사에 사용할 총 68개의 문항을 확정하였다.

### 3.3 예비조사 실시

예비조사는 개발된 검사의 예비문항이 구인을 제대로 측정하고 있는지 확인하기 위해 2022년 서울 소재 A대학 공과대학에 소속된 학생 294명을 대상으로 실시하였다. 설문에 참여한 학생들의 전공영역은 컴퓨터공학, 데이터과학, 소프트웨어융합, 디지털 미디어, 정보보호 등을 포함한다. 참여 학생의 학년분포는 1학년이 66명, 2학년이 74명, 3학년이 92명, 4학년 이상

이 62명으로 나타났다.

### 3.4 예비조사 결과분석 및 최종문항 확정

수집된 자료에서 문항을 성실하게 답변하지 않은 설문지를 제외하고 최종 288명이 분석에 포함되었다. 먼저, 예비문항의 타당도와 신뢰도를 확인하고자 구인타당도 검증, 신뢰도 검증, 문항분석을 실시하였다. 구인타당도 검증을 위해서는 검사의 구조를 탐색하는 탐색적 요인분석(Exploratory Factor Analysis; EFA)을 실시하였다. 탐색적 요인분석은 요인의 모형을 가정하기 전 전체 자료를 구성하는 요인의 수를 파악하고자 실시하는 방법으로, 요인의 수를 정리, 요약하여 최소화하는 과정을 통해 전체 검사의 구성요인의 특성을 탐색한다. 본 연구에서는 요인 간 관계성이 없음을 가정하는 직교회전(varimax)을 통해 행렬을 회전하여 요인을 추출하였다. 또한, 탐색적 요인분석의 적합성을 확인하기 위해 KMO (Kaiser-Meyer-Olkin) 척도와 Bartlett의 구형성(sphericity) 검정 결과를 확인하였다. 이후, 스크리 도표(scree plot)와 고유값(eigen value)을 확인하여 검사의 요인구성을 파악하였다. 탐색적 요인분석 결과를 바탕으로 요인의 수를 재구성하고 문항내용을 수정하였다.

다음으로 각 문항의 분석을 위해서 고전검사이론(Classical Test Theory; CTT)과 문항반응이론(Item Response Theory; IRT)에 근거한 Rasch 모형을 적용하여 문항분석을 실시하였다. 문항의 난이도는 고전검사이론과 Rasch 모형을 적용하여 산출된 값을 확인하였고, 문항의 타당성 확보를 위해서는 각 문항에 대한 학생들의 반응이 모형으로 예측되는 반응과 얼마나 일치하는지를 나타내는 문항적합도를 확인하였다. 적합도 통계치(fit statistics)인 내적 적합도(Infit MNSQ) 지수가 1.5 이상인 문항을 부적합 문항으로 판단하였다. 또한, 각 문항이 학생의 능력 수준별로 잘 구분하고 있는가를 확인하기 위해 문항변별도 지표인 점이연상관(Point-biserial correlation) 지수를 확인하였다. 점이연상관의 값이 .30 보다 낮은 경우, 해당 문항은 변별도가 떨어지는 문항으로 판단하였다.

마지막으로 검사결과의 안정성 및 일관성을 나타내는 신뢰도를 검증하기 위해 문항내적일관성 계수인 Cronbach's  $\alpha$ 를 산출하였다. 검사의 신뢰도, 타당도 검증 및 문항분석을 위한 과정은 SPSS 21.0과 Winsteps를 사용하여 실시하였다.

예비조사를 분석한 결과, 요인분석과 문항분석을

통해 전공역량 2문항, 일반문항 5문항이 제외되어 최종 61문항이 소프트웨어 역량 진단도구에 선정되었다. 최종문항에 포함된 측정구인 및 구인별 문항 수는 Table 2와 같으며, 응답양식은 5점 리커트척도로 구성하였다. 구체적으로, 전공역량에는 복잡한 문제를 추상화 및 절차화하여 컴퓨터로 자동화하여 해결하는 역량을 강조한 컴퓨터 사고력과 프로그래밍 능력, 데이터 수집, 처리 및 분석에 필요한 지식과 기술을 의미하는 데이터 분석역량, 소프트웨어 영역에서 해결해야 할 문제를 발견하고 자원을 수집하여 해결할 수 있는 창의적 문제해결역량, 알고리즘적 사고를 기반으로 데이터의 구조를 통합하고 활용할 수 있는 소프트웨어 융합역량이 포함되었다. 일반역량에서는 주로적으로 학습하고 자신의 자원을 관리하는 자기관리역량, 직무수행시 발생하는 갈등을 인지하고 능동적으로 조정할 수 있는 갈등해결역량, 가상공간과 현실공간이 연결된 초연결 사회에서 집단지성을 발휘하여 문제를 해결할 수 있는 초협업역량, 현장 실무 경험을 통해 직무를 효과적으로 수행하는 현장작용역량, 배경이 다른 사람들과 상호작용하여 소프트웨어를 개발할 수 있는 상호문화이해역량, 공공의 이익과 안전을 추구하여 윤리적 감수성을 가지고 판단하여 실천하는 소프트웨어 공학윤리역량이 포함되었다.

**Table 2.** Operational Definitions of Software Competencies Constructs

Category	Constructs	Operational definitions
Major-related competencies	Computing mindset and programming skills	The ability to abstract and proceduralize large and complex problems into algorithms and automate them with computers for resolution, and the capability to utilize programming languages to perform this process
	Data analysis competency	The possession of knowledge and skills necessary for data collection, processing, and analysis to extract insights or knowledge from data, and the capacity to apply these abilities
	Creative problem-solving skills	The competency to identify problems that need to be solved in the software domain, gather necessary resources, seek new executable solutions, derive optimal strategies for resolution, and implement them

General competencies	Software integration competency	The capacity to integrate and utilize the structure of data based on algorithmic thinking
	Self-management skills	The capacity to set and execute plans for achieving personal goals as a software major, while proactively learning and managing one's own resources (time, social resources)
	Conflict resolution skills	The ability to recognize and resolve conflicts with others that arise during software-related tasks, by analyzing the causes and actively and effectively mediating the conflict
	Collaboration skills	The capability to create new value and solve problems in a hyper-connected society, where virtual and real spaces are linked, by sharing information and exhibiting collective intelligence with global talents from various backgrounds, aiming for a common purpose
	Field adaptation skills	The competence to adapt effectively to the actual industry through field practical experience, based on the self-efficacy of one's identity as a software major and adaptation to the field
	Intercultural understanding skills	The ability to develop software effectively through open-minded interactions with people from diverse backgrounds, respecting cultural diversity
	Software engineering ethics competency	The capacity to autonomously make ethical decisions and act upon them in the development and utilization of software, with a focus on public interest and safety, and possessing professional expertise as a software major, while being sensitive to the ethical issues that may arise

## 4. 연구결과

### 4.1 문항분석

#### 4.1.1 전공역량 문항분석결과

전공역량 문항의 타당성을 검증하기 위해 고전검사 이론과 문항반응이론에 근거한 Rasch 모형을 적용하

여 문항난이도와 문항적합도 분석(item fit analysis)을 실시하였으며, 전공역량 예비문항에 대한 분석결과는 Table 3과 같다. 분석결과, 부적합 문항으로 판단할 수 있는 내적합도(Infit) 값이 1.5 이상인 문항은 나타나지 않았고 검토가 필요한 문항으로 판단할 수 있는 점이면 측정상관계수(point-measure correlation; 변별도)가 0.3 이하인 문항도 나타나지 않아 모든 문항을 유지하였다.

**Table 3.** Item Analysis Result of Major-related Competencies

Items	CTT item difficulty	Rasch Model		
		Item difficulty	Item Fit Statistic	Item discrimination
Item 1	3.07	-0.14	1.20	0.55
Item 2	2.47	0.95	1.31	0.52
Item 3	3.15	-0.30	0.94	0.65
Item 4	3.74	1.19	1.06	0.61
Item 5	2.90	0.01	0.87	0.69
Item 6	2.32	1.19	0.82	0.71
Item 7	3.96	1.74	0.93	0.64
Item 8	2.54	0.49	0.98	0.67
Item 9	2.52	0.76	0.88	0.69
Item 10	3.31	-0.41	1.06	0.60
Item 11	2.61	0.55	1.10	0.63
Item 12	2.69	0.65	0.74	0.74
Item 13	2.79	0.43	0.72	0.75
Item 14	3.39	-0.66	0.92	0.66
Item 15	3.19	-0.28	0.83	0.70
Item 16	2.79	0.26	0.71	0.76
Item 17	3.22	-0.35	0.82	0.71
Item 18	2.83	0.40	0.83	0.70
Item 19	2.71	0.47	1.04	0.63
Item 20	3.29	-0.55	1.41	0.50
Item 21	3.20	-0.37	0.91	0.67
Item 22	3.38	-0.73	1.18	0.55
Item 23	3.11	-0.25	1.13	0.59
Item 24	2.63	0.77	1.24	0.55
Item 25	3.23	-0.33	1.18	0.59
Item 26	2.76	0.39	1.15	0.60

#### 4.1.2 일반역량 문항분석결과

일반역량 문항도 동일한 분석을 통해 문항의 타당성을 검증하였다. Table 4에 제시된 분석결과, 문항 6(나의 감정이나 기분이 내가 하는 일에 영향을 주지 않는다)과 문항 32(나는 전공 관련 프로젝트나 인턴십 등을 통해 얻은 실무경험을 가지고 있다)의 내적합도(Infit)의 값이 각각 1.73과 1.76으로 부적합 문항으로 판단할 수 있는 1.5이상의 값을 나타냈다. 이에 따라, 문항 6과 문항 32를 삭제하였다. 반면, 점이면 측정상관계수(point-measure correlation;

변별도)가 0.3이하인 문항은 나타나지 않았다.

**Table 4.** Item Analysis Result of General Competencies

Items	CTT item difficulty	Rasch Model		
		Item difficulty	Item Fit Statistic	Item discrimination
Item 1	3.08	1.00	1.16	0.59
Item 2	3.19	0.94	1.05	0.61
Item 3	3.52	0.46	0.98	0.61
Item 4	3.92	-0.02	1.17	0.48
Item 5	3.44	0.62	1.12	0.57
Item 6	3.05	1.11	1.73	0.37
Item 7	3.37	0.76	1.2	0.54
Item 8	4.34	-0.98	0.88	0.54
Item 9	4.16	-0.49	0.9	0.57
Item 10	3.80	-0.11	0.87	0.64
Item 11	3.71	0.19	0.92	0.61
Item 12	3.63	0.27	0.94	0.61
Item 13	3.98	-0.47	0.93	0.58
Item 14	4.16	-0.37	0.81	0.62
Item 15	4.00	-0.01	0.92	0.58
Item 16	4.23	-0.38	0.80	0.6
Item 17	4.17	-0.31	0.83	0.6
Item 18	4.02	0.24	1.00	0.53
Item 19	4.34	-0.99	1.13	0.37
Item 20	3.95	-0.47	0.94	0.59
Item 21	3.66	-0.16	0.78	0.7
Item 22	3.78	-0.24	0.81	0.66
Item 23	3.59	0.20	0.95	0.62
Item 24	3.82	-0.27	0.80	0.66
Item 25	4.27	-0.67	0.90	0.55
Item 26	3.95	-0.27	1.07	0.52
Item 27	3.78	-0.25	0.91	0.62
Item 28	3.24	0.68	1.17	0.57
Item 29	3.47	0.26	0.96	0.62
Item 30	3.85	0.00	1.07	0.53
Item 31	2.90	1.40	1.07	0.61
Item 32	2.72	1.67	1.76	0.42
Item 33	4.28	-0.68	0.90	0.54
Item 34	3.88	0.19	0.86	0.64
Item 35	4.15	-0.75	0.89	0.58
Item 36	4.14	-0.67	1.05	0.51
Item 37	4.21	-0.49	0.97	0.54
Item 38	4.00	-0.35	1.04	0.54
Item 39	4.38	-0.96	1.10	0.45
Item 40	3.93	0.05	0.87	0.61
Item 41	3.86	-0.35	0.89	0.61
Item 42	4.18	0.66	0.89	0.57

#### 4.2 구인타당도 검증

문항 분석을 통해 문항의 적절성과 타당도 여부를 확인하여 부적절한 문항을 삭제한 후, 구인타당도를 검증하기 위해 1차 탐색적 요인분석을 실행하였다.



구인타당도 검증을 위한 탐색적 요인분석의 적합성을 나타내는 KMO 측도는 전공영역 .937, 일반영역 .936으로 나타났고, Bartlett의 구형성 검정 결과도 유의확률이 .05 미만으로 요인분석 모형이 적합한 것으로 판단되었다. 요인분석 결과 전공역량은 4개 요인, 일반역량은 6개 요인으로 구성되었고, 누적분산이 전공영역 61.67%, 일반영역 65.07%로 나타나 구성된 요인의 설명력이 높은 것으로 확인되었다. 요인분석 결과와 초기에 계획된 검사구조 및 문항분석 결과를 비교하여 구인과 문항 내용분석을 실시하였고, 문항의 요인 부하량(factor loading)이 0.3 이하인 경우 삭제하여 전공역량 요인 4개, 일반역량 요인 6개를 최종 결정하였다.

전공영역은 개발단계에서 명명한 데이터 처리 및 분석 능력이 데이터 수집과 분석, DB 구현 능력의 포괄적인 개념인 데이터 관리 능력을 포함하여 ‘데이터 분석역량’으로 수정되었고, 요인부하량과 문항 내용을 고려하여 문항이 해당 요인으로 이동되었다. 프로그래밍 능력, 컴퓨팅 사고력이 통합되어 ‘컴퓨팅 사고력과 프로그래밍 능력’으로 구인명이 수정되었고, 프로그래밍을 위한 기초역량인 수학 및 통계학의 문항들이 이 구인으로 흡수되었다. 창의적 문제해결력은 문항내용 분석을 통해 지식 검색능력을 문제해결력의 기초 단계로 판단하여 해당 구인의 조작적 정의를 수정하였다. 소프트웨어 응용 분야 지식과 기술은 알고리즘적 사고를 통해 해결방안을 도출하고, 데이터의 구조를 통합하고 활용할 수 있는 역량으로 판단하여 조작적 정의 수정 후 ‘소프트웨어 융합역량’으로 구인명을 수정하였다. 구인명과 문항위치를 이동한 후 최종 문항을 정리하여 2차 탐색적 요인분석을 실행하였다. 최종문항과 부하량은 Table 5와 같다.

**Table 5.** Factor Loadings of Major-related Competencies Items

	format suitable for computer-based solutions.	
	I can automate efficient solutions to various problems using a computer.	.526
	I can apply the problem-solving process using a computer to similar problems.	.551
Data analysis competency	I can collect data through programming, such as big data collection and web crawling.	.795
	I can analyze data using various data analysis techniques to derive insights tailored to specific objectives.	.755
	I can analyze and logically organize data.	.520
	I can implement databases necessary for various data management purposes.	.556
	I can abstract the key elements of various collected data.	.523
	I know the knowledge underlying big data analysis.	.717
Creative problem-solving skills	I have the knowledge retrieval skills (e.g. Googling) needed to develop software.	.629
	I can think of ways to improve software already developed in the field related to my major.	.508
	I can combine various methods that can be solved when solving complex problems.	.730
	I tend to stop when I find problems during software development and look for new alternatives.	.777
	During team projects, my team members tend to ask me for advice on problem solving.	.568
Software integration competency	I am aware of the security threats that exist at each layer of the computer network.	.544
	I know about computer algorithms.	.688
	I understand many ways to process the basic structure of data (arrangements, records, stacks, queues, lists, graphs, trees, etc.).	.592
	I can derive solutions through algorithmic thinking.	.646
	I understand the basic concept of artificial intelligence.	.480

Factors	Items	Loadings
Computing mindset and programming skills	I know the fundamental mathematics required in my field of study.	.696
	I possess the mathematical reasoning and applied skills necessary for learning in my field of study.	.652
	I can use more than two programming languages.	.692
	My proficiency in programming languages is intermediate or above.	.735
	I possess coding skills that meet the requirements of industry standards.	.488
	I can structure complex problems into a	.505

일반영역도 구인타당도의 결과에 따라 개발단계의 검사 구인과 문항의 조정이 있었다. 먼저, ‘나는 온라인 플랫폼을 활용하여 소통하는 과정에서 나의 사고와 감정을 자연스럽게 표현하는 편이다.’ 라는 문항은 자기관리 역량으로 개발되었으나, 문항 내용분석 결과 해당 요인에 적합하지 않은 것으로 판단되어 최종 삭제하는 것으로 결정하였다. 갈등해결 역량에서는 갈등해결에 필요한 역량인 의사소통 역량 문항 2개( ‘나는 팀 프로젝트를 수행할 때 일방적인 내 의견 주장이 아닌 상대방의 의견에도 귀 기울인다.’

‘나는 상대방 의견을 경청함으로써 내 의견의 옳고, 그름을 객관적으로 판단한다.’)를 해당 요인으로 이동하였고, 초협업 역량에서 의사소통 역량 3개(‘나는 다양한 직업군의 정보 또는 지식을 활용하여 팀 프로젝트를 수행한다.’, ‘나는 온라인 플랫폼에 제공된 정보를 취합하여 대중이 원하는 것을 이해할 수 있다.’, ‘나는 소프트웨어를 개발하기 위해 대중의 요구를 파악할 수 있다.’)를 요인부하량과 문항 내용분석 결과 온라인 상에서 필요한 협업 능력으로 판단하여 초협업 역량으로 이동하였다. 마지막으로 ‘나는 공동의 목적을 달성하기 위해 전체 팀원의 공헌이 중요하다고 생각한다.’와 ‘나는 다양한 사람들이 협업할 수 있도록 공동체가 정한 협업 규칙을 잘 따른다.’는 가상 공간이 아닌 현실 공간에서의 일반적인 협업의 의미로 판단하여 최종 삭제하기로 결정하였다. 일반역량의 최종문항과 부하량은 Table 6과 같다.

결과적으로 전공역량은 2개 문항 삭제 후 최종 4개 요인, 24개 문항, 일반역량은 5개 문항 삭제 후 6개 요인, 최종 37개 문항이 소프트웨어역량 진단 도구의 최종 문항으로 결정되었다.

**Table 6.** Factor Loadings of General Competencies Items

Factors	Items	Loadings
Self-management skills	I have a specific plan for my career.	.804
	I am implementing the plan I have established to achieve the goals I have set.	.849
	I seek and learn the knowledge and skills necessary for my studies on my own.	.695
	I efficiently complete assigned tasks and roles within the designated time frame.	.357
Conflict resolution skills	I have people around me whom I can ask for help when I face difficulties.	.650
	When working on team projects, I listen to others' opinions rather than just asserting my own.	.669
	I listen to others' opinions to objectively assess my views.	.626
	When conflicts arise in relationships with others, I tend to recognize them first.	.636
	When a conflict arises, I look at the difference between the other person and my position.	.744
	I consider the cause of the conflict from many perspectives.	.597
	I tend to try to find a solution when when conflicts occur.	.536
	If I find a way to resolve a conflict, I tend to implement it.	.438
When conflicts arise, I seek cooperation from others to find a solution.	.496	
Collaboration skills	I actively utilize digital communication tools necessary to share information with various talented people.	.522

	In a hyper-connected society, I can combine various types of information.	.678
	I can reorganize collected information for my intended purposes.	.723
	I enjoy the repetitive process of combining, verifying, and learning from diverse information sources.	.591
	I fully utilize my capabilities to achieve the goals of the community.	.516
	I can share the content I developed or collected information with others for the good of the public.	.304
	I carry out team projects using information or knowledge from various occupational groups.	.565
	I can understand what the public wants by gathering the information provided on the online platform.	.762
	I can gather information from online platforms to understand public needs for software development.	.709
Field adaptation skills	I consider how what I am currently studying can be applied in the field.	.597
	I actively participate in seminars, exhibitions, fairs, and academic activities related to my major.	.391
	I tend to demonstrate my competence well in practice-related courses.	.462
	I believe the knowledge in my major will be useful in the related industry.	.377
Intercultural understanding skills	I possess the abilities and skills needed for actual job performance in the industry related to my major.	.367
	I understand that people with the same cultural background can have different beliefs and values.	.595
	I know how to get along well with people who have different beliefs and values from mine.	.755
	I respect and get along well with people from different cultures and values.	.738
Software engineering ethics competency	I don't feel awkward or uncomfortable when I encounter people of different cultures and races in school or neighborhoods.	.472
	I would like to participate if I have the opportunity to work with people from diverse cultural and ethnic backgrounds.	.377
	As a software major, I feel a responsibility to pursue the public interest and safety.	.770
	I think software professionals should have both expertise and ethics.	.742
	I am aware of the ethical issues that can arise in software development and utilization.	.635
	I can autonomously judge ethical issues that may arise during software development and utilization.	.571
	I follow the learning ethics or research ethics required of students.	.593

### 4.3 문항반응 척도의 적절성 확인

소프트웨어 역량의 전공역량과 일반역량 측정을 위해 사용한 리커트 5점 척도가 적절한지 확인하기 위하여 문항 범주 특성 곡선을 살펴보았다. Figure 2에 제시한 것과 같이 전공역량 최종문항의 문항 범주 특성 곡선은 전반적으로 범주 1에서 5까지 모든 범주의 곡선이 뚜렷한 기능을 하고 있는 것으로 나타나 5점 척도가 적절한 것으로 나타났다. Figure 3에 나타난 일반역량 문항의 문항 범주 특성 곡선은 4번 문항 범주 2와 32번 문항에서의 범주 5가 뚜렷한 기능을 나타내고 있지 않는 부분이 발견되었으나, 대부분의 문항에서 범주 1에서 5까지 모든 범주의 곡선이 뚜렷한 기능을 하고 있는 것으로 나타나 5점 척도가 적절하다고 판단할 수 있다.

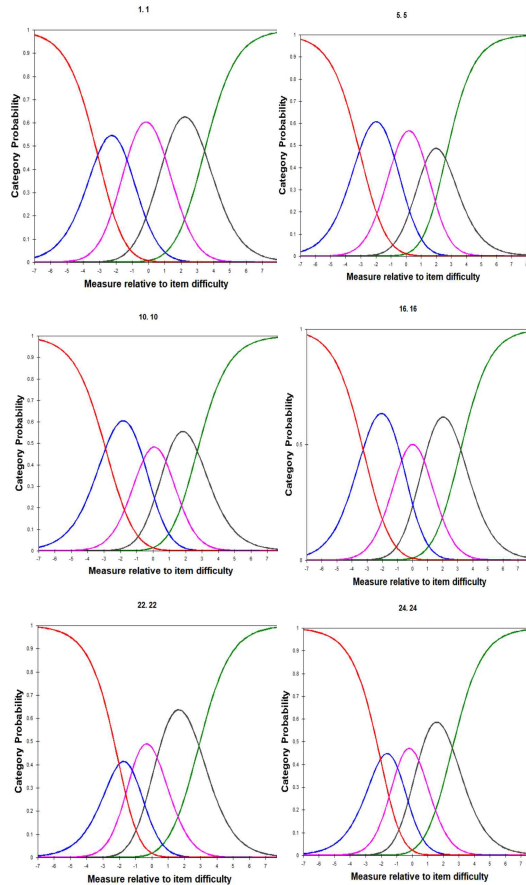


Figure 2. Item Characteristic Curves of Major-related Competencies

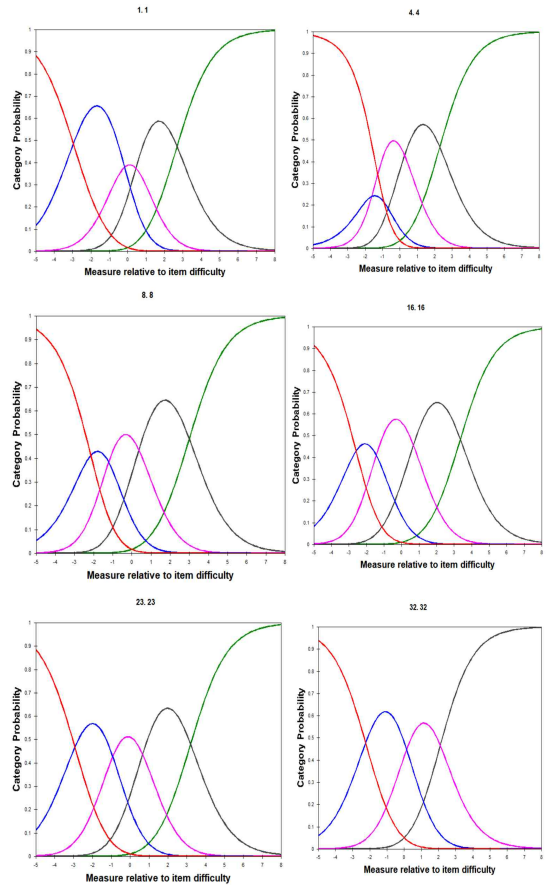


Figure 3. Item Characteristic Curves of General Competencies

### 4.4 피험자와 문항측정치 분포

학생들의 전공역량 수준과 문항 난이도의 분포를 동시에 비교하기 위해 Rasch 측정치에 의한 피험자-문항 지도(person-item map)를 확인하였다.

Figure 4의 좌측은 피험자 분포를, 우측은 문항 분포를 나타내며, 위쪽에 위치할수록 피험자의 전공역량 수준이 높고 문항 난이도가 높음(학생들의 응답값이 상대적으로 낮은 경우)을 의미한다. 피험자와 문항의 분포가 중간 지점에서 고르게 분포되어 있어 개발된 전공역량 문항이 피험자 능력에 적절한 난이도를 가지고 있는 것으로 나타났다. Figure 5에 제시한 일반역량 문항을 대상으로 한 피험자-문항지도는 피험자 능력추정치 분포가 문항 난이도 분포에 비해 상대적으로 높은 지점에 위치하고 있음을 알 수 있다. 이는 피험자의 능력수준이 문항의 난이도 분포보다 높

은 것을 의미한다. 이러한 결과는 대부분의 심리검사 문항에서 나타나는 패턴으로 본 검사의 목적이 상위 집단의 학생들을 변별하기 위한 목적이 아니라 소프트웨어 분야에서 대학생들이 요구되는 역량을 함양하고 있는지를 파악하는 것이므로 피험자 능력과 문항 난이도 분포는 검사활용에 직접적인 영향을 미치지 않을 것이다.

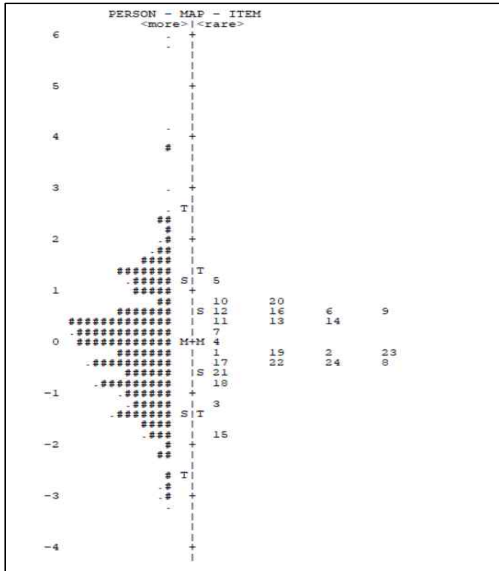


Figure 4. Person-item Map of Major-related Competencies

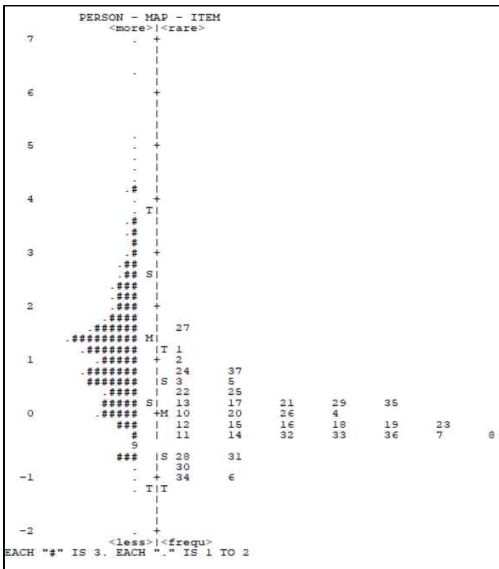


Figure 5. Person-item Map of General Competencies

#### 4.5 신뢰도 검증

소프트웨어 역량 진단도구의 신뢰도를 확인하기 위하여 내적일관성 계수 Cronbach's  $\alpha$  값을 산출하였다. 총 24개 문항으로 구성된 전공역량 진단도구의 신뢰도는 .95, 총 37개의 문항으로 구성된 일반 역량 진단도구의 신뢰도는 .95로 높은 신뢰도를 보였다. 요인별 신뢰도는 .79에서 .89 사이로 모두 양호한 수준이었다. 이를 통해 개발된 문항이 일관적으로 소프트웨어역량 구인을 측정하고 있음을 알 수 있다. 각 요인에 따른 신뢰도는 Table 7과 같다.

Table 7. Reliability Analysis

Major-related competencies	# of items	$\alpha$	General competencies	# of items	$\alpha$
Computing mindset and programming skills	8	.89	Self-management skills	5	.83
Data analysis competency	6	.87	Conflict resolution skills	8	.89
Creative problem-solving skills	5	.81	Collaboration skills	9	.89
Software integration competency	5	.80	Filed adaptation skills	5	.79
			Intercultural understanding skills	5	.84
			Software engineering ethics competency	5	.85
Total	24	.95	Total	37	.95

#### 5. 결론

본 연구의 목적은 대학생 대상 소프트웨어 역량 수준을 진단할 수 있는 측정도구를 개발하고 타당화하는 것이다. 개발된 소프트웨어역량 진단도구는 소프트웨어 전공 관련 지식 및 기술을 측정할 수 있는 전공역량과 전공역량을 발휘하기 위해 필요한 태도를 측정하는 일반역량으로 구성되어 있다. 기존 선행연구에서 소프트웨어 역량을 기술적, 사회적, 개인적 역량으로 구분하거나 소프트웨어 역량 내 기술적인면과 협업 의사소통과 같은 일반역량을 함께 제시한 것과는 달리, 본 연구에서는 소프트웨어 역량이 요구하는 기술적인 요소와 다양한 분야와의 협업에 필요한 일

반역량을 같은 비중으로 강조하기 위하여 소프트웨어 역량을 전공역량과 일반역량으로 분류하였다.

본 연구는 예비문항들의 타당도와 신뢰도를 확인하기 위해 구인타당도 검증, 신뢰도 검증, 문항분석을 실시하였으며 이를 통해 최종문항을 결정하였다. 전공역량과 일반역량 영역 모두 Cronbach  $\alpha$  값이 .95로 높은 신뢰도를 보였다. 또한 Rasch 모형을 적용하여 문항들의 적합도를 확인하고 문항 적합도가 기준치를 넘은 문항은 삭제하였다. 검사의 구인을 확인하는 구인타당도 검증에서도 문항이 계획된 구인에 포함되지 않은 경우, 내용분석을 통해 삭제 또는 더 적합한 구인으로 이동하였다. 삭제된 문항을 제외하고 최종문항을 대상으로 문항응답척도의 적절성과 피험자 능력치와 문항 난이도의 분포를 확인하였고, 응답척도와 분포는 모두 적절한 것으로 나타났다. 따라서 본 연구에서 개발한 소프트웨어역량 진단도구는 학생들의 역량 수준을 확인하고 교과과목별 교과과정과 프로그램의 효과성을 검증하고 목표달성 여부를 확인하기 위해 사용할 수 있는 신뢰롭고 타당한 검사도구라고 볼 수 있다.

이렇게 개발된 진단도구를 향후 어떻게 사용할 것인가는 중요한 문제이다. 현재까지 소프트웨어 역량을 자기 보고식으로 진단할 수 있는 도구는 보고된 바 없다. TOPCIT(Test Of Practical Competency in IT)와 같은 소프트웨어 전문가 업무를 수행하는 데 요구되는 지식 기술 태도를 평가하는 시험은 있지만, 자기 보고식으로 소프트웨어 역량을 진단하는 검사는 처음 개발되었다. 따라서 개발된 소프트웨어 역량 진단도구를 효과적으로 활용하는 방안을 논의하는 것은 필요하다.

먼저, 진단도구를 활용하여 학생들의 강점과 약점을 파악하고 이를 기반으로 학생 성장 계획을 수립할 수 있다. 소프트웨어 역량이 높은 부분은 유지하고, 부족한 부분은 교과목 또는 비교과 프로그램을 통해 보완할 수 있을 것이다. 특히, 전공생과 비전공생의 강점이 다를 수 있으므로 역량수준을 기반으로 맞춤형 지원이 가능할 것이다. 박금주와 최영준[19]의 연구는 비전공생을 대상으로 한 소프트웨어 교육은 기술을 습득하는 것보다 창의력과 문제해결력을 향상하는데 초점을 두어야 한다고 강조하였다. 또한, 각 전공과 연계한 융합교육을 통해 소프트웨어 교육이 진행되어야 함을 밝히고 있다. 본 연구에서 개발한 도구의 하위요소인 창의적 문제해결력과 소프트웨어 융합 능력은 비전공생을 이해하는 데 도움을 줄 수 있는 구

인이 될 것이다.

둘째, 수집된 결과를 기반으로 대학 내 소프트웨어 교육 및 훈련 프로그램을 개선할 수 있다. 어떤 부분에서 학생들이 높은 역량을 보이고 어떤 부분에서 부족한지를 파악하여 교육 커리큘럼을 수정하거나 보완함으로써 효율적인 교육을 제공할 수 있을 것이다.

셋째, 소프트웨어 역량 진단도구 결과를 토대로 취업 및 진로 탐색을 진행할 수 있다. 구체적으로 소프트웨어 역량 검사결과는 학생들이 어떤 직무에 적합한지를 파악하는데 유용한 정보를 제공할 것이다. 이를 기반으로 직무별 요구되는 역량과 맵핑하여 학생의 강점을 살릴 수 있는 진로상담을 진행할 수 있을 것이다.

넷째, 소프트웨어 역량 조사 결과는 학생 수준별 지도를 가능하게 할 것이다. 김경미와 김현숙[20]은 학습자 맞춤형 소프트웨어 교육 방안연구에서 프로그래밍 수업은 학습자의 진행 단계를 반영한 교육과정 수립이 중요하다고 설명하였다. 또한, 학습자의 성향이나 학습태도를 고려하여 맞춤형 교육과정이 필요함을 강조하였다. 소프트웨어 역량의 기술적인 수준은 학생별로 차이가 있을 수밖에 없다. 소프트웨어 역량이 높은 학생들은 수준에 맞는 프로젝트 수행 기회를 제공할 수 있고 외부에서 제공되는 소프트웨어 인증 프로그램이나 대회 등에 참여할 수 있도록 유도할 수 있다. 이를 통해 학생들은 자신의 역량을 실전에서 펼칠 수 있는 기회를 얻을 수 있다. 반면, 역량수준이 낮은 학생들은 해당 콘텐츠에 대한 두려움을 극복할 수 있게 교육을 지원해야 하고 기본이론 이해 및 실습을 통해 기초를 다질 수 있는 기회를 제공해야 한다.

마지막으로, 소프트웨어 역량 진단 도구를 활용하여 학생들의 변화추이 및 교육의 성과를 분석할 수 있다. 시간이 흐름에 따라 역량 지수의 추이를 정기적으로 분석하여 교과 및 비교과 프로그램의 효과를 모니터링하고 교육과정 개선하는 데 반영할 수 있다. 결론적으로, 소프트웨어 역량 진단도구 결과를 종합적으로 활용하여 학생들의 성장 방향을 지원하고, 교육과정을 향상시켜 산업계에 요구하는 우수한 소프트웨어 엔지니어를 양성할 수 있는 기회를 제공할 수 있을 것으로 기대한다.

본 연구에서 개발한 소프트웨어 역량 진단 도구의 기대효과에도 불구하고 개발된 진단도구의 활용을 향상시키기 위한 추후 노력도 필요하다. 본 연구의 도구 개발에는 제한된 인원이 참여했기 때문에 보다 다양한 피험자를 대상으로 도구를 표준화하여 일반화 가

능성을 높이는 것이 필요하다. 또한, 기준점 설정 방법을 통해 등급화 또는 지표화를 진행하여 결과를 쉽게 해석할 수 있도록 지원한다면 본 도구의 활용도를 향상시킬 수 있을 것이다.

## 참고문헌

- [1] Hyundai Research Institute. (2016). *The Emergence of the Fourth Industrial Revolution and its Implications: Reflections on Creating a New Economic System 16-32*(v705). Seoul. Hyundai Research Institute.
- [2] Seo, J. Y. (2016). Status and Strategies of Basic Software Education for Non-Majors: A Case Study of Ajou University. *The Korean Association of General Education*, 123-130.
- [3] Kim, W. S. (2019). A Study on the Students' Perceptions Trend for Software Essentials Subject in University. *Korean Journal of General Education*, 13(4), 161-180.
- [4] Software, Naver Dictionary, <https://dict.naver.com/>
- [5] Turley, R. T., & Bieman, J. M. (1995). Competencies of exceptional and nonexceptional software engineers. *Journal of Systems and Software*, 2(1), 19-38. DOI : 10.1016/0164-1212(94)00078-2
- [6] Rivera-Ibarra, J. G., Rodríguez-Jacobo, J., Fernández-Zepeda, J. A., & Serrano-Vargas, M. A. (2010, March). Competency framework for software engineers. *Proceeding of 2010 23rd IEEE Conference on Software Engineering Education and Training, Pittsburgh, Pennsylvania, USA(pp. 33-40)*. DOI : 10.1109/CSEET.2010.21.
- [7] Association for Computing Machinery, & IEEE Computer Society (2015). *Software Engineering 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering: A Volume of the Computing Curricula Series*. Association for Computing Machinery, & IEEE Computer Society.
- [8] IEEE Computer Society. (2014). *Software Engineering Competency Model*. Washington, DC. IEEE Computer Society.
- [9] Joo, E. W., & Joo, I. H. (2023). Analyzing the Types of Subjective Perceptions of Coding Education among College Students. *Journal of KSSSS*, 6(2), 27-48. DOI: 10.18346/KSSSS.64.2
- [10] Hong, S. H., Seo, J., Goo, E., Shin, S. Oh, H., & Lee, T. (2019). Exploratory Study on the Model of the Software Educational Effectiveness for Non-major Undergraduate Students. *Journal of the Korean Association of Information Education*, 23(5), 427-440. DOI: /10.14352/jkaie.2019.23.5.427
- [11] Park, G. (2019). Network Analysis for the Evaluation of Software Education. *Global Creative Leader: Education & Learning*, 9(4), 67-88. DOI : 10.34226/gcl.2019.9.4.67
- [12] Lee, J. H., & Jang, J. H. (2017). Study of Software Coding-based Making Competencies. *The Korean Society for Creative Information Culture*, 3(2), 81-92. DOI : 10.32823/jcic.3.2.201710.81
- [13] Kang, S. W. (2017). Fostering Software Development Competencies in Secondary Education: Directions and Strategies. *Communications of the Korean Institute of Information Scientists and Engineers*, 35(4), 34-43.
- [14] Lim, K. H., & Kim, M. R. (2021). High School Students' Software Capability Enhancement Behavior Exploration: As an Individualized Instructional Strategy For Elective Curriculum 'Programming'. *The Korean Association of Computer Education*, 24(1), 17-28. DOI : 10.32431/kace.2021.24.1.003
- [15] Yoo, S., Baek, J., & Jang, Y. (2022). Analysis of the Relationship between AI competency and Computational Thinking of AI Liberal Arts Class Students. *Journal of Korean Association of Computer Education*, 25(5), 15-26. DOI : 10.32431/kace.2022.25.5.002
- [16] Lee, W. J. (2020). A Study on the Improvement Scheme of University's Software Education, *Journal of The Korea Society of Computer and Information*, 23(3), 243-250. DOI :10.9708/jksci.2020.25.03.243
- [17] Jang, S. M. (2021). AI · Software Education for University Liberal Education. *The Journal of Education and Culture*, 3(1), 33-43.
- [18] Jang, E. & Kim, J. (2019). Contents Analysis of Basic Software Education of Non-majors Students for Problem Solving Ability Improvement-Focus on SW oriented University in Korea- *Journal of internet Computing and Services*, 20(4), 81-90. DOI : 10.7472/jksii.2019.20.4.81
- [19] Park, G. J. & Choi, Y. J. (2018). Exploratory study on the direction of software education for the non-major undergraduate students. *Journal of Education & Culture*, 24(4), 273-292. DOI : 10.24159/JOEC.2018.24.4.273

이 윤 선



1993 서울여자대학교  
교육심리학과(학사)  
1995 서울여자대학교  
교육심리학과(석사)  
2004 University of Washington(박사)

2008 ~ 현재 서울여자대학교 교육심리학과 교수  
관심분야: AI기반 데이터분석, 차별문화기능, 검사타당도,  
교육측정 및 평가  
E-Mail: ylee@swu.ac.kr

김 은 수



2000 서울여자대학교 기독교학과(학사)  
2003 서울여자대학교  
가족상담학과(석사)  
2021 서울여자대학교  
교육심리학과(박사수료)

2010 ~ 현재 서울여자대학교 교양대학 바롬인성교육부 강사  
관심분야: 검사개발 및 타당화, 교육성과 측정 및 평가, 빅데이터,  
인성교육  
E-Mail: eskim411@swu.ac.kr

이 현 숙



2005 서울여자대학교 경영학과(학사)  
2018 서울여자대학교  
인성교육학과(석사)  
2023 서울여자대학교  
교육심리학과(박사수료)

2023년 ~ 현재 서울여자대학교 데이터혁신팀 전임연구원  
관심분야: 검사개발 및 표준화, 교육성과 측정 및 평가, 중단연구,  
머신러닝, XAI  
E-Mail: hyunsooklee@swu.ac.kr

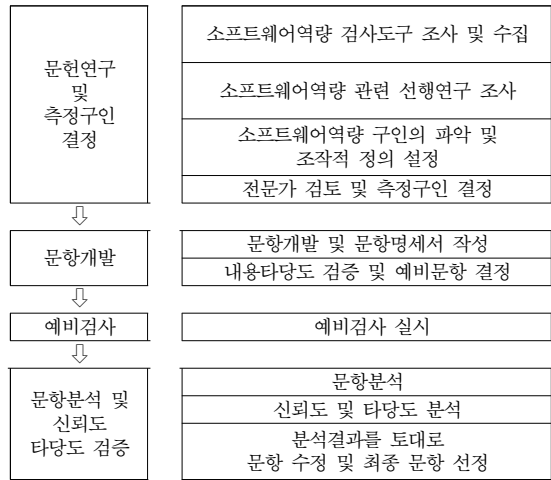
이 세 희



2021 서울여자대학교  
교육심리학과(학사)  
2024 서울여자대학교  
교육심리학과(석사)

2023년 ~ 현재 서울여자대학교 성과환류데이터센터 전임연구원  
관심분야: 교육 데이터마ining, 고등교육 성과관리, 심리측정,  
양적연구방법론  
E-Mail: seheelee@swu.ac.kr

부 록



[그림 1] 연구절차

<표 2> 측정구인의 조작적 정의

영역	측정구인	조작적 정의
전공역량	컴퓨팅 사고력과 프로그래밍 능력(8문항)	크고 복잡한 문제를 추상화 및 알고리즘으로 절차화하고 컴퓨터로 자동화하여 해결할 수 있는 역량과 이 과정을 수행하기 위한 프로그래밍 언어를 활용할 수 있는 역량
	데이터 분석역량(6문항)	데이터에서 인사이트 혹은 지식을 추출하기 위한 데이터 수집, 처리 및 분석에 필요한 지식과 기술을 갖추고 있으며 이를 활용할 수 있는 역량
	창의적 문제해결 역량(5문항)	소프트웨어 영역에서 자신이 해결해야 할 문제를 발견하고 필요한 자원을 수집하여 새롭게 실행 가능한 해결안을 모색하여 최적의 해결전략을 도출하고 실행할 수 있는 역량
	소프트웨어 융합역량(5문항)	알고리즘적 사고를 기반으로 데이터의 구조를 통합하고 활용할 수 있는 역량
일반역량	자기관리 역량(5문항)	소프트웨어 전공자로서의 개인적인 목표를 달성하기 위해 계획을 세우고 실행하면서 주도적으로 학습을 하고 자신의 자원(시간, 사회적 자원)을 스스로 관리하는 역량
	갈등해결 역량(8문항)	소프트웨어 관련 직무를 수행할 때 발생하는 타인과의 갈등 상황을 인지하고 그 원인을 분석하여 해결책을 모색한 후 갈등을 능동적이고 효과적으로 조정할 수 있는 역량
	초협업역량(9문항)	가상공간과 현실공간이 연결된 초연결 사회에서 공동의 목적을 달성하기 위해 다양한 배경의 전문지식을 가진 글로벌 인재들과 정보를 공유하고 집단지성을 발휘하여 문제해결과 더불어 새로운 가치를 만들어 내는 역량

현장적응 역량(5문항)	소프트웨어 전공자로서의 정체성과 현장적응에 대한 자기효능감을 바탕으로 현장 실무경험을 통해 실제 산업현장에서 잘 적응하고 주어진 직무를 효과적으로 수행할 수 있는 역량
상호문화 이해역량(5문항)	문화적 다양성을 존중하는 개방적인 자세를 가지고 서로 다른 배경의 사람들과 효과적으로 상호작용하여 소프트웨어를 개발할 수 있는 역량
소프트웨어 공학윤리 역량(5문항)	공공의 이익과 안전을 추구하고 소프트웨어 전공자로서의 전문성을 가지고, 소프트웨어 개발과 활용 단계에서 직면할 수 있는 윤리적 문제에 대해 윤리적 감수성을 바탕으로 자율적으로 판단하며 이를 실천할 수 있는 역량

〈표 8〉 소프트웨어역량 진단 도구 전공역량 최종문항

번호	역량	문항
1	컴퓨터사고력과 프로그래밍능력	나는 전공 분야에서 필수적으로 요구되는 기초 수학에 대해서 알고 있다.
2		나는 나의 전공 분야를 학습하는 데 필요한 수학적 사고방식과 응용력을 갖추고 있다.
3		나는 2개 이상의 프로그래밍 언어를 사용할 수 있다.
4		나의 프로그래밍 언어 활용 능력은 중급이상이다.
5		나는 산업체 현장 요구 수준에 부합하는 코딩 능력을 갖추고 있다.
6		나는 복잡한 문제를 컴퓨터로 해결할 수 있는 형태로 구조화할 수 있다.
7		나는 다양한 문제에 적합한 효율적인 해결방법을 컴퓨터로 자동화할 수 있다.
8		나는 컴퓨터를 이용한 문제 해결 과정을 유사한 다른 문제에 적용할 수 있다.
9	데이터분석역량	나는 프로그래밍을 통해 데이터를 수집할 수 있다(예: 빅데이터 수집, 웹 크롤링).
10		나는 데이터 분석 기법을 활용하여 목적에 맞게 데이터를 분석할 수 있다.
11		나는 데이터를 분석하고 논리적으로 조직할 수 있다.
12		나는 다양한 데이터 관리에 필요한 데이터베이스를 구현할 수 있다.
13		나는 수집한 여러 가지 데이터의 핵심 요소를 추상화할 수 있다.
14		나는 빅데이터 분석의 기초가 되는 지식을 알고 있다.
15	창의적문제해결역량	나는 소프트웨어 개발에 필요한 지식검색 능력(예: 구글링)을 갖추고 있다.
16		나는 전공 관련 분야에서 이미 개발된 소프트웨어를 개선할 수 있는 방법을 생각할 수 있다.
17		나는 복잡한 문제를 해결할 때 해결 가능한 다양한 방법들을 조합할 수 있다.
18		나는 소프트웨어 개발 과정에서 문제가 발견되면 멈추고 새로운 대안을 모색하는 편이다.
19		팀 프로젝트를 수행할 때 팀원들이 나에게 문제 해결과 관련된 조언을 구하는 편이다.

20	소프트웨어융합역량	나는 컴퓨터 네트워크의 각 계층에 존재하는 보안 위협에 대해 알고 있다.
21		나는 컴퓨터 알고리즘에 대해 알고 있다.
22		나는 데이터의 기본구조를 처리하는 여러 가지 방법들(배열, 레코드, 스택, 큐, 리스트, 그래프, 트리 등)을 이해하고 있다.
23		나는 알고리즘적 사고를 통하여 해결방법을 도출할 수 있다.
24		나는 인공지능의 기본개념을 이해하고 있다.

〈표 9〉 소프트웨어역량 진단 도구 일반역량 최종문항

번호	역량	문항	
1	자기관리역량	나는 희망진로에 대한 구체적인 계획이 있다.	
2		나는 내가 설정한 목표를 이루기 위해 수립한 계획을 실행하고 있다.	
3		나는 학업에 필요한 지식과 기술을 스스로 찾아서 학습한다.	
4		나는 정해진 시간 내에 주어진 과제나 역할을 잘 수행한다.	
5		나는 내가 어려움을 겪을 때 도움을 요청할 수 있는 주변 사람들이 있다.	
6	갈등해결역량	나는 팀 프로젝트를 수행할 때 일방적인 내 의견 주장이 아닌 상대방의 의견에도 귀 기울인다.	
7		나는 상대방 의견을 경청함으로써 내 의견의 옳고 그름을 객관적으로 판단한다.	
8		나는 타인과의 관계에서 갈등이 발생했을 때 이를 먼저 인지하는 편이다.	
9		나는 갈등이 발생하면 상대방과 내 입장의 차이점이 무엇인지 살펴본다.	
10		나는 갈등의 원인을 여러 관점에서 살펴본다.	
11		나는 갈등이 생기면 해결 방법을 찾으려고 노력하는 편이다.	
12		나는 갈등을 해결할 수 있는 방법을 찾으면 이를 실행하는 편이다.	
13		나는 갈등이 생기면 상대방의 협조를 구하면서 해결책을 찾으려고 노력한다.	
14		초협업역량	나는 다양한 인제들과의 정보 공유를 위해 필요한 디지털 소통 도구를 적극적으로 활용한다.
15			나는 초연결 사회에서 다양한 정보들을 조합할 수 있다.
16			나는 수집한 정보를 내가 원하는 목적에 맞게 재조합할 수 있다.
17			나는 다양한 배경의 정보를 조합하고, 검증하면서 배우는 반복적인 과정을 즐긴다.
18			나는 공동체의 목표를 달성하기 위해 나의 역량을 충분히 발휘한다.
19			나는 공공의 선을 위해 내가 개발한 콘텐츠나 수집한 정보를 타인과 공유할 수 있다.
20			나는 다양한 직업군의 정보 또는 지식을 활용하여 팀 프로젝트를 수행한다.
21			나는 온라인 플랫폼에 제공된 정보를 취합하여 대중이 원하는 것을 이해할 수 있다.
22		나는 소프트웨어를 개발하기 위해 대중의 요구를 파악할 수 있다.	



23	현장적용역량	나는 현재 내가 공부하고 있는 것이 현장에 어떻게 적용될 수 있는지를 생각해본다.
24		나는 전공과 관련한 교내의 세미나, 전시회, 박람회, 학술활동 등에 적극적으로 참여한다.
25		나는 실습관련 과목에서 나의 역량을 잘 발휘하는 편이다.
26		나는 내가 배우고 있는 전공지식이 관련 산업현장에서 유용하게 쓰일 것이라고 믿는다.
27		나는 산업현장에서 실제 직무수행 시 필요한 전공 관련 능력이나 기술을 갖추고 있다.
28	상호문화이해역량	나는 같은 문화적 배경을 가진 사람일지라도 서로 다른 신념이나 가치관을 가질 수 있음을 이해하고 있다.
29		나는 나와 다른 신념이나 가치관을 가진 사람들과 잘 지내는 방법을 알고 있다.
30		나는 나와 다른 문화나 가치관을 가진 사람들을 존중하며 잘 지내는 편이다.
31		나는 학교나 동네에서 다른 문화와 인종을 가진 사람들과 마주쳤을 때 어색하거나 불편한 감정이 없다.
32		나는 다양한 문화적, 인종적 배경을 가진 사람들과 함께 일할 기회가 있다면 참여하고 싶다.
33	소프트웨어공학윤리역량	나는 소프트웨어 전공자로서 공공의 이익과 안전을 추구해야 한다는 책임감을 느낀다.
34		나는 소프트웨어 전문가는 소프트웨어 분야의 전문성과 윤리식을 함께 가져야 한다고 생각한다.
35		나는 소프트웨어 개발과 활용 단계에서 발생할 수 있는 윤리적인 문제에 대해 알고 있다.
36		나는 소프트웨어 개발과 활용 단계에서 발생할 수 있는 윤리적인 문제에 대해 자율적으로 판단할 수 있다.
37		나는 학생에게 요구되는 학습윤리나 연구윤리를 따른다.