



## **A Method of Test Quality Management for Medical Software Quality Digitizing**

**A-Young Park<sup>1</sup>, Ki-Won Song<sup>2</sup>**

<sup>1</sup>*Department of Medical Engineering, Konyang University*

<sup>2</sup>*Department of Medical Information Technology Engineering, Konyang University*

---

### **A B S T R A C T**

This paper presents the model of realizing requirement to ensure the QCD(Quality Cost Delivery) of medical software during the development cycle. In many cases, customer needs are likely to change in the process of carrying out projects. When medical software is released to customers without the needs to be reflected, many more defects can be found when the software is actually used in the medical field. The defects are fatal in the medical domain. To solve these problems, GQM(Goal Question Metric) methodology is used. With this methodology, goals are set, questionnaires are devised, and review of domain expert is sought. In addition, requirements are extracted and defined. And then, test cases are finally completed based on the requirements. The test cases made through these processes are quantitatively prioritized by using the AHP(Analytic Hierarchy Process) methodology. Once the priority is determined, the test case is configured in order to determine how the test case has been carried out in accordance with the number of test cases. It is possible to digitize the quality of software before it is delivered to the customers to see whether enough tests are officially performed for the project. Using this TQM, it is expected that the level of software quality is determined while the project is being carried out, instead of evaluating the software at the end of the project.

© 2017 KKITS All rights reserved

---

**KEYWORDS :** TQM[Test quality management], AHP, GQM, Medical software, Quality assurance, Software process, QCD

---

**ARTICLE INFO:** Received 9 November 2016, Revised 24 January 2017, Accepted 10 February 2017.

---

---

\*Corresponding author is with the Department of Medical Engineering, Konyang University, 158 Gwanjeodong-ro

Seo-gu Daejeon, 35365, KOREA.

*E-mail address:* [kiwonsong@konyang.ac.kr](mailto:kiwonsong@konyang.ac.kr)

### 1. 서론

오늘날, 소프트웨어 시장의 성장세와 함께 소프트웨어의 규모가 커지고 다양화되어 가고 있다. 그에 따라 소프트웨어의 품질의 고급화가 요구되고 있으며, 품질이 소프트웨어 상품의 완성도를 위한 중요한 제약 조건으로 급부상 하고 있다. 특히 의료 및 전장과 같이 생명과 연관되어 있는 분야의 경우 결함 하나가 굉장히 치명적이다[1]. 따라서 안전성 및 신뢰성 보증을 위해 품질 활동이 반드시 필요하다[2]. 품질 활동을 시행하면 수시 및 정기적으로 점검을 수행해야 하며 기업의 크기와 역량에 따라 추구하는 제품에 대해 정확히 이해하고 시행해야 한다. 또한 품질 활동은 <그림 1>과 같이 개발 초기단계(분석, 설계)부터 시작되어야 한다.

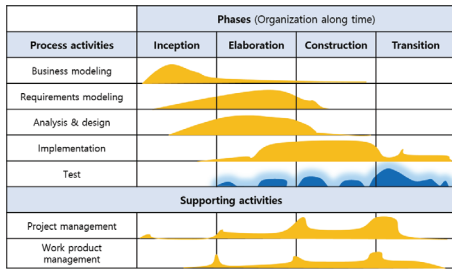


그림 1. 개발 프로세스  
Figure. 1. Development Process

<그림 2>와 같이 진행할 경우 프로세스 과정에서 테스트케이스의 추적이 불가능하고 요구사항에 대한 수작업 관리로 인해 결과에 대한 이력을 확보하기 어렵다. 결국 결함이 발견되어도 납기일을 맞추기 위해 그대로 상품을 내보낸다. 상품 출시 후 고객의 불만사항이 접수되기 시작하고, 회사는 유지보수에 많은 시간과 노력을 투입시켜 자원의 낭비로 이어지게 된다.

품질관리 부재로 인한 사례로서, 2014년 9월 신생아 인큐베이터 40% 노후 장비 사례를 들 수 있

다. 전국의 인큐베이터 3069대 중 제조시기를 확인할 수 없는 인큐베이터가 21.6%(664대)에 이르며 그 중에서도 10년 이상 사용된 오래된 장비가 40.5%에 달한 것으로 조사됐다(전자신문 2014. 09. 21).[3] ‘의료장비현황 신고대상 및 식별부호화에 관한 기준’ 고시에 포함된 의료장비 192종 중 품질관리를 하는 의료장비는 전산화단층촬영장치(CT), 자기공명영상촬영장치(MRI), 유방촬영용장치 3종뿐이다[4].

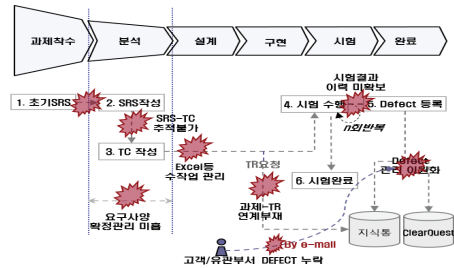


그림 2. 품질 추적 프로세스  
Figure. 2. Quality Trace Process

이런 점을 해결하기 위해 본 연구는 소프트웨어의 품질 표준인 ISO 9126 요소에 따라 ISO 12207에서 제시하는 프로세스를 반영한다. 품질은 소프트웨어 생명주기와 함께 변화한다. 품질 관정도 또한 여러 관점에서 반영된다. 프로세스 중심의 품질관리는 데이터 품질관리 프로세스를 유지 및 개선하는 방법이다. 이 방법에서는 데이터 오류뿐만 아니라 프로세스 개선을 통해 데이터 오류의 근본적인 원인을 제거하여 동일한 데이터 오류가 재발되는 것을 방지하게 된다.

프로세스 단계별로 품질 보증 활동에 대한 정의 및 항목, 범위 등을 정의하고 각 회사에 맞추어 적용할 수 있는 모델을 제안한다. 반드시 테스트해야 할 요구사항을 추출하고, 확정된 요구사항을 바탕으로 critical, major, minor로 수준을 나누어 테스트 계획을 자동 생성한다. 개발 프로세스를 진행하는

동안 발생하는 신규 및 수정 요구사항을 반영할 수 있고 구현과 함께 테스트가 진행된다. 불필요한 비용이나 스케줄 지연 없이 정량적 테스트가 가능해지며 요구사항을 문서화할 수 있고 테스트 커버리지 향상을 통해 추적성을 확립한다.

본 연구에서는 GQM과 AHP를 적용하여 요구사항이 항목별로 검증되었는지 정합률을 확인한다. 또한 구글의 어플리케이션 테스트 과정과 비교하여 TQM의 사용성에 대한 결과를 다룬다. 향후 품질 보증 활동에 대해 수치화된 결과를 기술한다.

## 2. 관련 연구

의료 소프트웨어의 품질 측정은 개발 전 단계에 걸쳐 진행된다. 의료 소프트웨어는 생명과 연관되어 있으므로 대상자의 안전성이 요구된다. 또한 의료기기 목적에 맞는 기능 및 사용 환경의 테스트 뿐만 아니라 국제표준 규격에 의거한 계획적이고 체계적인 품질 활동이 진행되어야 한다.

2절에서는 TQM 모델을 구성하는 방법론을 중심으로 품질활동을 진행하기 위한 테스트 계획이 만들어지는 과정을 소개하고자 한다.

### 2.1 품질 표준 ISO/IEC 9126의 특징

고객이 초기에 제시한 요구사항이 실제로 제품이 인도되었을 때 달라진다. 그러므로 여러 관점에서 품질을 평가해야 한다. 이를 위해 정의된 품질 모델인 ISO/IEC 9126 품질모형을 사용한다. 표준은 표현모형과 측정모형 간의 구조적 일관성이 우수해야 한다. 특히 서비스 품질에 대한 개념을 도입하여 개발단계, 시험 및 인수단계, 운용단계 전반에 걸쳐 적용이 가능한 품질모형을 구축함으로써 평가체계의 완전성을 확보하였다는 측면에서 우수하다고 할 수 있다[5].

ISO/IEC 9126 표현모형의 내·외부 품질 표현체계는 소프트웨어 품질을 6개의 품질특성으로 구성된다. 이를 다시 부특성들로 세분화한 품질 모형을 명시한다. 소프트웨어 품질 특성은 <그림 3>과 같이 여섯 가지 특성(기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성)으로 구분되며, 이러한 특성들은 다시 부특성들로 세분된다[6].

본 연구에서는 ISO9126의 특성을 반영하여 요구사항과 테스트케이스를 작성한다. 만족도에 따라 테스트 커버리지 측정 및 수치화가 가능하도록 한다.

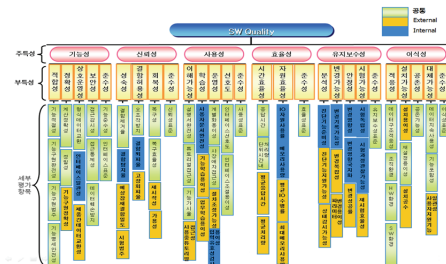


그림 3. ISO/IEC 9126 품질 특성 및 부특성  
Figure. 3. ISO 9126 quality characteristics & sub-characteristics

#### 2.1.1 기능성(Functionality)

기능성은 소프트웨어가 특정 조건에서 사용될 때, 명시된 요구와 내재된 요구를 만족하는 기능을 제공하는 소프트웨어 제품의 능력이다. 이 특성은 요구를 충족하기 위해서 소프트웨어가 무엇을 하는가에 대하여 관심을 둔다. 부특성으로는 적합성, 정확성, 상호운영성, 보안성, 준수성이 해당된다.

#### 2.1.2 신뢰성(Reliability)

신뢰성은 명세된 조건에서 사용될 때, 성능 수준을 유지할 수 있는 소프트웨어 제품의 능력이다. 신뢰성의 한계는 요구사항, 설계 및 현상의 결합에

기인한다. 부특성으로는 성숙성, 결합허용성, 복구성, 준수성이 해당된다.

### 2.1.3 사용성(Usability)

사용성은 소프트웨어가 명시된 조건에서 사용될 경우, 사용자에게 의해 이해되고, 학습되고, 사용되고 선호될 수 있는 소프트웨어 제품의 능력이다. 사용자에는 소프트웨어 사용에 영향을 받거나 의존하는 운영자, 최종 사용자, 그리고 간접 사용자 등이 포함된다. 부특성으로는 이해도, 학습성, 운영성, 선호도, 준수성이 해당된다.

### 2.1.4 효율성(Efficiency)

효율성은 명시된 조건에서 사용되는 자원의 양에 따라 요구된 성능을 제공하는 소프트웨어 제품의 능력이다. 자원은 다른 소프트웨어 제품, 하드웨어 장비, 재료 등을 포함한다. 부특성으로는 시간 반응성, 자원 활용성, 준수성이 해당된다.

### 2.1.5 유지보수성(Maintainability)

유지보수성은 소프트웨어 제품이 변경될 수 있는 능력이다. 변경에는 환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정과 개선 혹은 개작 등이 포함된다. 부특성으로는 분석성, 변경성, 안정성, 시험성, 준수성이 해당된다.

### 2.1.6 이식성(Portability)

이식성은 소프트웨어가 특정 조건에서 사용될 때, 특정 환경에서 다른 환경으로 전이될 수 있는 소프트웨어 제품의 능력이다. 여기서 환경은 조직, 하드웨어 혹은 소프트웨어 환경을 말한다. 부특성

으로는 적응성, 설치성, 공존성, 대체성, 준수성이 해당된다[7].

## 2.2 GQM 방법론

목표중심 GQM 방법론은 개념적인 단계로서 조직의 목표를 정의하고 목표를 달성하기 위해 질문(Question)을 하고 질문을 정량적으로 측정할 수 있는 지표(Metric)을 만드는 것이다.

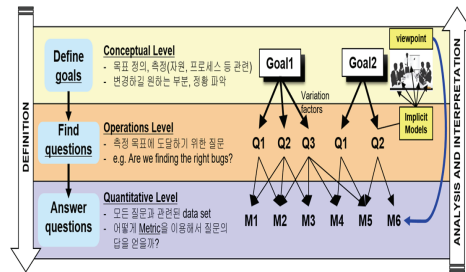


그림 4. GQM 방법론 계산 흐름  
Figure. 4. GQM Method Calculation Process

GQM 방법론의 계산은 <그림 4>와 같은 순서로 진행된다. 개념적인 단계로서 목표(Goal)를 정의하고 목표를 달성하기 위한 질문(Question)을 한다.

일반적으로 목표기반 프로세스는 <그림 5>와 같이 비전과 목표를 설정하고 대외효과(External effectiveness)와 대내효율(Internal efficiency)을 구분하는 영역들로 구성된다[8][9].



그림 5. Lynch와 Cross의 수행 피라미드  
Figure. 5. Lynch and Cross Performance Pyramid

Quality와 Delivery는 조직의 외부 효과성을 측정하는 목표이고 Cycle time과 Waste는 내부 효율성을 측정하는 목표이다[10][11]. 이러한 각각의 영역에서 다시 프로젝트의 목표를 잡고 GQM을 통하여 좀 더 프로세스를 개선할 수 있는 전략을 세우고 측정하는 방법이다[12].

본 논문에서는 GQM을 활용하여 테스트가 필요한 품질 속성을 추출하고 이를 테스트 계획에 반영할 수 있도록 한다.

### 2.3 AHP 방법론

분석 계층 작업 AHP 방법론은 우선순위를 측정하기 위한 이론이다. 이론을 수행하기 위해서는 문제를 정의하고 목표를 설정한다. 이후 한 가지 문제에 대하여 여러 가지 요소와 대안을 마련한다. 의사결정 요소들 간의 관계를 분석하여 작성된 계층 구조를 바탕으로 비교 행렬을 작성한다. 행렬을 통해 상대적 중요도를 판단하여 종합 중요도를 결정한다. 각각의 요소를 1:1비교를 하여 접근한다. 이 비교값을 집합하여 기하평균을 적용함으로써 점수화를 통한 신뢰도 높은 객관적인 결정이 가능하다[13].

본 연구에서는 GQM 방법론을 통해 추출된 요구사항의 상대적 중요도를 측정하기 위해 AHP 방법론을 사용한다. 효과적으로 요구사항의 우선순위를 의사결정하며 테스트 계획에 반영한다. AHP 방법론 적용과정을 단계별로 분류하여 내용을 살펴보면 다음과 같다.

#### 2.3.1 계층 구조 형성

의사 결정의 최종 목표와 문제를 정의하고 더 세밀한 요소와 대안 등 구성요소들 사이의 관계를 표현하기 위해 계층구조가 이용된다. 최상위 레벨

은 하나의 원소로 구성되며 전반적인 목적을 나타낸다. 하위 계층들은 각기 복수의 요소들로 구성되며 바로 상위 계층의 요소의 세부 요소이다. 하위 계층으로 갈수록 더 세밀한 요소들이 나타난다. 마지막으로 최하위 계층에는 대안들이 나타난다.

#### 2.3.2 계층별 쌍대비교

계층별 쌍대비교를 통해 계층별로 쌍대 비교 행렬을 구한다. 계층 내에 있는 요소들의 중요도에 대한 의사 결정자의 선호도를 평가하는 단계이다.

#### 2.3.3 상대적 중요도 계산

이 단계에서는 계층별 쌍대비교를 통해 얻어진 쌍대비교 행렬을 이용하여 각 행렬별로 요소들의 상대적 중요도를 구한다. 중요도를 계산하는 방법은 여러 가지가 있으며 대표적으로 고유 벡터 방법, 산술평균법 등이 있다.

#### 2.3.4 일관성 검증

계산 결과에 대한 전체적인 일관성을 검토한다. 이 결과 값이 0.1 이하일 경우 실제로 신뢰성이 높다고 판단한다. 결과 값이 0.1 이상일 경우 비교판단의 일관성이 결여되어 있는지 혹은 계층 구조 설정에 문제가 있었는지 검토한다.

#### 2.3.5 복합우선순위 설정

대안들에 대한 전체적인 우선순위를 설정하는 단계이다. 대안들 각각의 복합우선순위는 우선순위 벡터의 곱의 합으로 나타낸다.

### 3. TQM 개요

#### 3.1 개요

본 논문에서는 품질 보증 활동을 수행하기 위해 필요한 TQM 모델에 대한 기본적인 개념과 구조를 제시한다. 또한 각 프로세스에 해당하는 활동에 대해 정의한다. 마지막으로 이 모델의 특징에 대하여 언급한다.

#### 3.2 TQM 모델

테스트케이스는 본질적으로 테스트할 데이터의 집합을 알아내는 것이다[14]. 이에 TQM은 테스트케이스를 구성하고 수행 여부를 판단하여 수치화할 수 있도록 한다. 또한 요구사항 기반의 테스트 계획을 자동으로 생성한다. 품질 보증 활동에 대한 정의 및 항목, 범위 등을 정의한 후 이를 반영하여 회사에 최적화된 테스트 계획을 생성한다.

거에 완료된 과제 정보(과제명, 과제코드, 요구사항 등)와 현재 과제와의 연계를 통해 재사용, 수정될 요구사항이나 테스트케이스를 파악한다.

② 품질 활동의 개선을 위해서는 요구사항 및 테스트케이스에 대한 관리가 요구된다. TQM은 필요에 따라 MS project와 같은 EPM(Enterprise Project Manager) 프로그램을 활용하여 일정과 함께 관리할 수도 있다. 과제정보를 연계할 수 있게 되는 것이다.

③ 분석(Analysis) 단계에서는 EPM 프로그램에서 일정, 계획 등을 확인하고 TQM에서는 요구사항 데이터베이스와 테스트케이스 데이터베이스를 활용한다. 요구사항과 테스트케이스에 대한 재사용, 수정 또는 신규 내역에 대하여 파악한다. SRS와 테스트케이스 정보를 연계하고 표준에 맞추어 SRS(Software Requirement Specification) 문서를 작성한다. SRS가 확정되면 master plan이 자동 생성된다.

④ 설계(Design) 단계에서는 신규 및 수정된 요구사항들이 반영된 STC(Software Test Case)와 UTC(Unit Test Case)를 작성한다.

⑤ 구현(Implementation) 단계에서는 단위시험을 반복해서 수행한다. 이 때 개발팀이 결함을 등록하면 등록된 결함을 처리한다. 이후 UTC가 100% 만족하면 단위시험을 완료하고 SQE(Software Quality Engineer) 담당자에게 TR(Test Request)을 요청한다.

⑥ 시험(Test) 단계에서는 시스템시험을 반복해서 수행한다. 이 때 SQE가 결함을 등록하면 등록된 결함을 처리한다. 이후 STC가 100% 만족하면 시스템테스트를 완료하고 성능시험을 수행한다. 테스트케이스의 수행 개수에 따라 어느 정도 품질 활동이 진행되었는지 테스트 coverage를 측정할 수 있다. 해당 내용은 TQM 데이터베이스에 저장되며 언제든지 재사용 가능하다.

이 모델의 특징은 다음과 같다. :

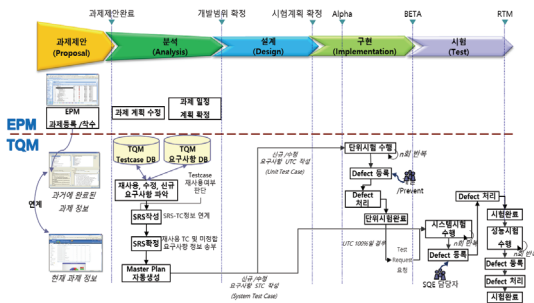


그림 6. TQMM 모델  
Figure. 6. TQMM Model

TQM 모델은 <그림 6>과 같다.

① 품질 활동의 개선을 이루기 위해 핵심 프로세스는 TQM 중심으로 연속적으로 수행되어야 한다. STP(System Test Plan)에는 과제명과 일정이 포함되며 테스트케이스는 포함되지 않는다. 또한 과

첫째로, ISO12207과 ISO9126을 호환한다는 점이다. 기존의 품질 활동들은 일관되게 정의되어 있지 않다. 품질 관리를 위해 필요한 조직이나 자원 또는 인력의 교육 수준을 고려하기 위해 TQM 모델은 해당 표준에서 제시된 프로세스 분류 방식과 산출물을 활용한다. ISO12207을 통해 SQE의 할 일을 정의하고 ISO9126의 특성을 반영하여 테스트 커버리지 측정이 가능하도록 한다.

둘째로, 결함 처리에 대한 문제점을 보완하였다. 이전 테스트 수행 방식에서는 프로세스의 말단인 테스트 단계에서만 한꺼번에 진행하는 경향을 보였다. 따라서 결함이 발견되어도 납기일을 맞추기 위해 그대로 출시할 수밖에 없었다. TQM에서는 단위테스트 및 시스템테스트를 반복하여 진행한다. 지속적으로 요구사항을 업데이트하고 발견되는 결함을 등록 및 처리하여 충분한 테스트를 수행한다.

셋째로, 재사용이 가능하도록 하였다. 프로젝트를 시작하면 요구사항을 새로 구성해야하는 불편함이 있었다. 방대한 양의 테스트케이스를 작성해야했다. 이에 TQM은 요구사항이 변하지 않으면 테스트케이스도 재사용이 가능하다. 재사용, 신규, 수정해야할 고객의 요구사항을 분리시키고 재사용 및 수정된 요구사항의 경우 기존의 것을 테스트케이스와 매핑하여 어느 시기에 어떠한 테스트를 해야 할지 테스트케이스와 테스트 계획을 자동 생성할 수 있다. 테스트 계획에서 제시된 각각의 테스트에 대한 수행 여부에 따라 품질 활동에 대한 수치화도 가능하다.

### 3.3. TQM과 구글의 테스트기법 비교

구글에서는 어플리케이션을 등록하기 전에 테스트를 진행하도록 되어 있다. 알파/베타 테스트를 제공하며 테스트는 30분 정도의 짧은 시간 내에 수행된다. 그러나 구글에서 제공하는 테스트 과정에

서는 완벽한 기능 테스트가 불가능하며 단순히 어플리케이션이 정상 동작하는지만 확인한다.

앞서 제시한 TQM과 구글 어플리케이션 테스트의 사용성을 종합적으로 평가하기 위해 AHP 기법을 활용한다.

#### 3.3.1 계층 구조 형성

최상위 레벨은 품질 보증 활동이며 전반적인 목적을 나타낸다. 각각 선정한 요인과 대안을 <그림 7>처럼 계층화된 구조도로 표현한다[15].

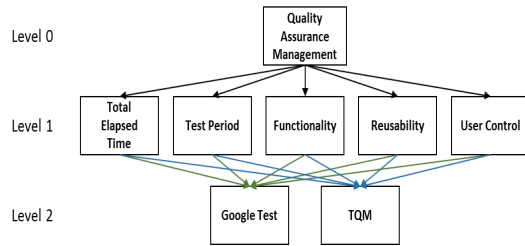


그림 7. AHP 기법을 위한 계층 구조화  
Figure. 7. Decomposition and Model Structuring for AHP method

<그림 7>에서처럼 품질 활동을 목표로 2가지 대안인 구글 어플리케이션 테스트, TQM을 5가지 요인(Total Elapsed time, Test Period, Test Range, Reusability, User Control)과 함께 계층구조화시켰다.

각 요인들은 테스트에 소요되는 시간(Total Elapsed time)과 프로세스에 대입해 보았을 때 테스트가 시작되는 시점(Test Period), 테스트 범위(Test Range), 재사용성(Reuseability) 그리고 사용자가 중간에 결함을 등록하거나 산출물을 관리할 수 있는지에 대한 파악과 사용자가 프로그램을 제어할 수 있는 정도(User Control)를 비교할 수 있도록 구성한다. 요인 및 대안은 <표 1>에서 확인할 수 있다.

표 1. 요인 및 대안  
Table 1. Factor and Alternative

|                     |                         |
|---------------------|-------------------------|
| 요인<br>(Factor)      | Total Elapsed time (TE) |
|                     | Test Period (TP)        |
|                     | Test Range (TR)         |
|                     | Reusability (R)         |
|                     | User control (UC)       |
| 대안<br>(Alternative) | Google test (GT)<br>TQM |

### 3.3.2 계층별 쌍대 비교

특정 계층 내에 있는 요소들의 중요도에 대한 의사 결정자의 선호도를 평가하기 위해 계층별로 쌍대 비교 행렬을 구한다. 요인별로 1:1 비교하여 우선순위를 정하고 그 결과를 바탕으로 중요도를 계산한다. 해당 중요도는 <그림 8>과 같은 설문지를 활용하여 계산한다.

- 1) 과제명 :
- 2) 과제 코드 :
- 3) 과제 수행 기관 :

| Goal  | Question  | Metrics             | Score  |
|---|---|---------------------|--------|
| Quality<br>(고객에게 전달되는 제품의 품질을 고객이 인식할 정도로 높여야 함)        | Defect Density<br>(프로젝트 안에서 도입이 얼마나 밀집하고 얼마나 재발되는가) | total 요구사항 수        |        |
|   |   | Total Testcase 수    |        |
|   |   | 변경된 요구사항 수          |        |
|   |   | 반경된 Testcase 수      |        |
|   |   | 재용 질량도              |        |
|   |   | 시간당 결함 행렬 수 (상위 기준) |        |
|   |   | 기술문서 결함도            |        |
|   |   | 기술문서 결함 조치율         |        |
|   |   | 코드 결함도              |        |
|   |   | 코드 결함 조치율           |        |
| Impact of Requirement<br>(고객이 요구 변경이 프로젝트에 얼마나 영향을 주는가) |   | 요구기능 변경율            |        |
|   |   | 생산성                 |        |
|   |   | 코드당 문서화 규모          |        |
|   |   | 원당 문서화 규모           |        |
| Waste<br>(프로젝트 진행에서 가용자원의 낭비를 줄여야 함)                    | Productivity<br>(현재 프로젝트의 생산성이 얼마나 되는가)             | 코드 보충 재사용율          |        |
|   |   | Testcase 재사용율       |        |
|   | Reuse<br>(일마나 재사용 되는가)                              | 요구사항 재사용율           |        |
|   |   | 재작업 시간 (내내나 보는가)    | 재작업 시간 |
|   | Size estimation<br>(Size의 리소스를 얼마나 보는가)             | Size 추적 역량          |        |

그림 8. GQM을 기반으로 한 정량적 설문지  
Figure. 8. Quantitative Questionnaire based GQM

전문가들의 의견을 반영한 설문지의 결과를 바탕으로 각각의 요인별로 쌍대 비교를 수행하면 <표 2>와 같다. (1:같음, 5:중요, 9:매우중요)

표 2. 각 요인별 쌍대비교표  
Table 2. Paired comparison table for each factor

|    | TE    | TP    | TR    | R     | UC    |
|----|-------|-------|-------|-------|-------|
| TE | 1     | 0.143 | 0.111 | 0.200 | 0.333 |
| TP | 7.000 | 1     | 0.333 | 3.000 | 5.000 |
| TR | 9.000 | 3.000 | 1     | 5.000 | 7.000 |
| R  | 5.000 | 0.333 | 0.200 | 1     | 3.000 |
| UC | 3.000 | 0.200 | 0.143 | 0.333 | 1     |

동일한 방식으로 하나의 요인에 따라 2가지의 대안을 1:1 로 쌍대 비교한다. 수행결과는 <표 3>과 같다. TE에서는 GT의 중요도가 더 높았지만 나머지 4가지 요인(TP, TR, R, UC)에서는 TQM이 더 높은 점수를 받았다.

표 3. 계층별 쌍대비교표  
Table 3. Paired comparison table by hierarchy

| TE  | GT    | TQM   |
|-----|-------|-------|
| GT  | 1     | 7.000 |
| TQM | 0.143 | 1     |
| TP  | GT    | TQM   |
| GT  | 1     | 0.250 |
| TQM | 4.000 | 1     |
| TR  | GT    | TQM   |
| GT  | 1     | 0.143 |
| TQM | 7.000 | 1     |
| R   | GT    | TQM   |
| GT  | 1     | 0.125 |
| TQM | 8.000 | 1     |
| UC  | GT    | TQM   |
| GT  | 1     | 0.125 |
| TQM | 8.000 | 1     |

### 3.3.3 상대적 중요도 계산

쌍대비교표를 이용하여 각 행렬 요소들의 상대

적 중요도 점수를 계산한다. 이때 모든 중요도의 합은 1이어야 한다. 각각의 요인들을 <표 2>의 각 요인별 쌍대비교표의 열의 합으로 나누어 계산한다. 이렇게 계산된 쌍대계산 결과는 <표 4>와 같다. 5가지 요인 중 TR이 높은 점수를 받았다.

표 4. 요인의 상대적 중요도  
Table 4. Relative importance of factors

|    | Importance |
|----|------------|
| TE | 0.035      |
| TP | 0.260      |
| TR | 0.503      |
| R  | 0.134      |
| UC | 0.068      |

동일한 방식으로 요인별 대안의 상대적 중요도를 계산한다. 계산 결과는 <표 5>와 같다.

표 5. 대안의 상대적 중요도  
Table 5. Relative importance of alternative

| TE  | Importance |
|-----|------------|
| GT  | 0.875      |
| TQM | 0.125      |

| TP  | Importance |
|-----|------------|
| GT  | 0.200      |
| TQM | 0.800      |

| TR  | Importance |
|-----|------------|
| GT  | 0.125      |
| TQM | 0.875      |

| R   | Importance |
|-----|------------|
| GT  | 0.111      |
| TQM | 0.889      |

| UC  | Importance |
|-----|------------|
| GT  | 0.111      |
| TQM | 0.889      |

### 3.3.4 일관성 검증

지금까지의 계산 결과에 대한 전체적인 일관성을 검토한다. 이때 일관성지수(CI)값을 이용한다. CI

는 Consistency Index의 약자로 비교수행자가 얼마만큼의 일관성을 가지고 결과를 적었는가를 보여주는 지표이다. 응답자가 논리적으로 모순을 유발하게 되면 CI값이 증가하게 된다. 일관성지수 계산식은 아래 식 1과 같다.  $\lambda_{max}$ 는 고유최대치이며 요인의 쌍대 비교값과 요인의 중요도의 두 행렬의 곱하고 각 행의 합을 계산한다. 행의 합은 상대적 중요도를 뜻하며 행벡터를 상대적 중요도 벡터로 나누면 최대 고유치가 계산된다. n은 요인의 개수이다[15].

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (1)$$

$\lambda_{max}$ 는 5.243이고 n은 5이므로 식 (1)로 계산된 일관성 지수의 값은 0.061이다. CI값이 0.1 이하일 경우 실제로 신뢰성이 높다고 판단한다.

### 3.3.5 복합우선순위 설정

대안들에 대한 전체적인 우선순위를 설정하는 단계이다. 우선 각 요인별 대안의 중요도와 일관성 지수의 총합을 보면 <표 6>과 같다.

표 6. 각 요인별 대안의 중요도  
Table 6. importance of each alternative for each factor

|           | TE    | TP    | TR    | R     | UC    |
|-----------|-------|-------|-------|-------|-------|
| GT        | 0.875 | 0.200 | 0.125 | 0.111 | 0.111 |
| TQM       | 0.125 | 0.800 | 0.875 | 0.889 | 0.889 |
| $\lambda$ | 2     | 2     | 2     | 2     | 2     |
| CI        | 0     | 0     | 0     | 0     | 0     |

## 4. 결 론

AHP 기법을 사용하여 얻은 결과를 보면 구글 테스트는 17% TQM은 83%로 TQM이 구글 테스트

보다 66% 더 사용성이 높게 나왔음을 <표 7>에서 확인할 수 있다.

표 7. 최종 우선순위  
Table 7. Final priority

|             | Importance |
|-------------|------------|
| Google Test | 0.167826   |
| TQM         | 0.832174   |

이는 요인들의 상대적 중요도에 각 요인별 대안의 중요도를 행렬의 곱으로 계산을 수행한 것이다.

의료 소프트웨어는 하드웨어와 결합하여 사람의 생명을 다루는 의료 분야에서 사용된다. 품질 보증 활동이 중요한 분야라고 할 수 있다. 따라서 타 소프트웨어보다 안전성을 보장할 수 있어야 한다.

기존의 연구보다 ISO 9126에 대한 호환성을 높이고 요구사항 및 테스트케이스의 재사용, 수정, 신규에 대한 파악 및 반영에 초점을 맞추어 진행하였다.

본 논문에서는 소프트웨어 품질 측정을 위한 방안으로 GQM과 AHP 방법론을 사용할 것을 제안하였고, 품질 표준과 평가 방안으로 제시했다. 이는 테스트를 위해 많은 시간과 비용을 지불하지 않고도 메트릭을 활용하여 테스트 계획을 얻을 수 있음을 의미한다. 또한 과거 과제의 요구사항과 테스트케이스의 재사용이 가능하고 테스트 커버리지 측정이 가능하도록 하여 품질 보증 활동을 수치화할 수 있다고 주장한다. 이를 통해 중소기업과 대기업, 각 기업에 수준에 맞게 테스트 계획 수립이 가능하다.

향후 연구방향으로는 GQM과 AHP방법론을 활용한 테스트 계획을 제공하기 위해 오픈 플랫폼을 구현할 것이다. 또한 구현된 플랫폼에서 요구사항을 문서화하고 중요도를 수치화하여 회사에 맞는 메트릭을 생성한다. 이후 개발 과정에서 사례연구를 통한 실질적인 신뢰도를 검증한다.

## References

- [1] F. E. Young, *Validation of medical software: present policy of the Food and Drug Administration*, Annals of Internal Medicine, Vol. 106, pp. 628-629, Apr. 1987.
- [2] J. R. Evans, and W. M. Lindsay, *The management and control of quality*, OH: South-Western, pp. 115-128, 2002.
- [3] Medical Quality Management, <http://www.vop.co.kr/A00000794825.html>, Sep. 2014.
- [4] J. Park, *A development of indicators to decide replacement priority for medical equipment in hospital*, The Korean Operations Research and Management Science Society, Vol. 37, No. 4, pp. 161-180, 2012.
- [5] K-W. Song, *Measurement and management of the level of quality control process in Soc(system on chip) embedded software development*, International Journal of Advanced Robotic Systems, Mar. 2012.
- [6] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, *Usability meanings and interpretations in ISO standards*, Software Quality Journal, Vol. 11, No. 4, pp. 325-338, 2003.
- [7] C. Lee, H-S. Yang, *Development of quality evaluation model for software based on SOA*, Journal of the Korea contents association, Vol. 8, No. 8, pp. 82-93, May 2008.
- [8] R. L. Lynch, and K. F. Cross, *Measure up!*, Blackwell, 1995.
- [9] T. Kasse, *Action focused assessment for software process improvement*, Artech House, 2002.
- [10] K-W. Song, *Opportunity tree framework*

design for quality and delivery of S/W product, International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer Berlin Heidelberg, pp. 1331-1337, 2005.

[11] F. Van Latum, and R. Van Soligen, *Adopting GQM-based measurement in an industrial environment*, IEEE software, 1998.

[12] B. Boehm, *Value-based Software engineering: Case study*, IEEE Computer, pp. 33-41, 2003.

[13] T. L. Saaty, *Decision making with the analytic hierarchy process*, pp. 3-4, 2008.

[14] P. C. Jorgensen, *Software testing: A craftsman's approach*, CRC press, pp. 4-6, 2016.

[15] S. C. Kim, and H-J. Eo, *Priority aggregation for AHP based on experts opinions*, KORMS, Vol. 19, No. 3, pp. 41-51, Dec. 1994.

만들어진 테스트케이스에 수치적으로 우선순위를 부여하기 위해 AHP(Analytic Hierarchy Process) 방법론을 활용한다. 이렇게 우선순위가 결정되면 순서대로 테스트케이스가 구성되며 테스트의 통과 개수에 따라 테스트케이스가 얼마나 수행되었는지 판단이 가능하다. 이를 통해 충분한 테스트가 공식적으로 수행되었는지 고객이 확인할 수 있도록 고객에게 전달하기 전에 s/w 상품의 품질을 수치화 할 수 있다. 따라서 TQM을 사용하면, 프로젝트가 끝나고 소프트웨어를 평가하는 대신 프로젝트 진행 중에 소프트웨어 품질 수준이 결정될 것으로 기대한다.

## 의료 소프트웨어 품질 정량화를 위한 TQM 기법

박아영<sup>1</sup>, 송기원<sup>2</sup>

<sup>1</sup> 건양대학교 의료공학과

<sup>2</sup> 건양대학교 의료IT공학과

### 요 약

본 논문은 개발 주기 동안 의료 s/w의 QCD(Quality Cost Delivery)를 보장하기 위한 요구사항이 제대로 구현되었는지 테스트를 하기 위한 모델을 제시한다. 대부분, 고객의 요구사항은 프로젝트 수행 과정에서 변경될 가능성이 높다. 특히 의료 분야에서는 테스트를 하지 않고 의료 소프트웨어를 상품화할 경우 훨씬 더 많은 결함이 발견될 수 있다. 특히 의료분야의 결함은 치명적이다. 이러한 문제를 해결하기 위해, GQM(goal question metric)방법론을 사용한다. 목표를 설정하고 설문지를 구성하고 전문가의 답변을 받는다. 그리고 답변을 반영한 요구사항을 추출한다. 이렇게



**A Young Park** received the bachelor's degree in the Department of Medical Information Technology Engineering from the Konyang University in 2016.

She studied the M.S. degree in the Department of Medical Engineering from Konyang University since 2016.

E-mail address: parkayoung92@naver.com



**Ki-Won Song** received the bachelor's degree in the Department of Computer Engineering from the Konyang University in 2002.

He received the M.S. degree and the Ph.D. degree in the Department of Software Engineering from Chung-ang University in 2002 and 2004, respectively. From 2004 to 2007. He has been a professor in the Department of Medical Information Technology Engineering at Konyang University since 2015.

E-mail address: kiwonsong@konyang.ac.kr