패키지 소프트웨어 시험 프로세스와 평가모듈의 개발

이 하 용[†] · 황 석 형[††] · 양 해 술[†††]

요 약

패키지 소프트웨어는 그 특성상 동일한 유형에 속하는 다수의 제품 중에서 구매자가 적합하다고 판단되는 제품을 식별할 수 있어야 한다. 패키지 소프트웨어 제품에 대한 구매자의 선택 능력은 객관적인 품질 시험 절차와 방법을 통해 정해진 기준에 부합되는가를 판단할 수 있는 체계를 갖추고 있는가에 달려 있다. 이러한 체계를 구축하기 위해 패키지 소프트웨어에 적용할 수 있는 표준으로서 <ISO/IEC 14598-4 : 구매자를 위한 품질평가 프로세스>와 <ISO/IEC 12119 정보기술-소프트웨어 패키지-품질 요구사항 및 시험>이 있다. 본 연구에서는 이러한 표준을 기반으로 패키지 소프트웨어에 대한 품질시험 프로세스를 구축하고 시험 메트릭과 적용 방법을 개발함으로써 구매자가 효과적으로 자신의 요구에 맞는 패키지 소프트웨어를 선택할 수 있는 체계를 구축하였다.

Development of Package Software Test Process and Evaluation Module

Ha-Yong Lee[†] · Suk-Hyung Hwang[††] · Hae-Sool Yang[†††]

ABSTRACT

Package software should have the feature that enables purchasers to discriminate a product suitable for them among a number of software belonging to the similar kind of product. Purchaser's ability to choose a package software depends on whether they can judge that a package software conforms to the relevant standard through an objective quality test process and method or not. There are the standards that can be applicable to the quality evaluation of package software, such as <ISO/IEC 14598-4 : Quality Evaluation Process for Acquirers> and <ISO/IEC 12119 : Information Technology-software package-Quality Requirements & Test>. This study developed a system with which purchasers can effectively select a package software suitable for their needs, building quality test process for package software and developing test metric and application method.

키워드 : 패키지 소프트웨어(Package Software), 품질 평가(Quality Evaluation), 메트릭(Metric)

1. Introduction

Due to the rapid spread of personal computers, a variety of package software for personal or office use have been developed, and consequently the liberty of choice has been broadened.

Package software should have the feature that purchasers can discriminate a product suitable for them among a number of software, which belong to the similar kind of product.

If we want to make a right choice for package software, we should consider whether a package software satisfies the established standard or not through objective quality test process and method.

For building this system, there are the standards that can be applicable to package software, such as <ISO/IEC 14598-4 : Quality Evaluation Process for Acquirer> and <ISO/IEC 12119 : Information Technology - software package - Quality Requirements & Test> [2, 3].

In case of ISO/IEC 12119, those can use it, such as software developers, organizations for authentication that intend to establish third party authentication, organization for approving authentication and test centers, and software purchasers [3].

This study developed the method that can contribute to quality improvement of package software by building the quality test process for package software based on this standard and developing test metric and application method [14].

ISO/IEC 9126 and ISO/IEC 12119 include quality evaluation guideline and methodology for general software and package software but have not concrete measurement way [12].

In this research, it's different to extract and integrate the except part of 12119 to be based ISO/IEC 12119 from 9126, . it make an evaluation module for package software evaluation to be adapted by evaluation module to be presented an configuration mode in ISO/IEC 14598 and make a point to

※ This research was supported by University IT Research Center.
† 종신회원 : 호서대학교 벤처전문대학원
†† 종신회원 : 선문대학교 컴퓨터정보학부 교수
††† 종신회원 : 호서대학교 벤처전문대학원 교수
논문접수 : 2003년 5월 29일, 심사완료 : 2003년 7월 21일

the test table mode with existing standards.

There need an evaluation module to be developed to matched by software pattern and the test table to apply a quality evaluation methodology to be managed by international standard essentially, and it's developed the way to evaluate a product to based an package software on the external viewpoint in this research.

The evaluation for package software is a conformance oriented evaluation to identify whether accomplished software product conform specification for product and detailed functions or not. In this study, we developed evaluation process and quality evaluation table to evaluate the conformance of package software.

This study introduces the present research state related to quality in Chapter 2 [15], and builds the test process for package software from the purchaser's viewpoint in Chapter 3.

It introduces quality model for testing package software in Chapter 4, and describes the metric and evaluation module that was developed based on quality model in Chapter 5, and describes the test example in chapter 6, and finally describes the conclusion and study works after this.

2. Present state of related study

2.1 Foreign Trend

Foreign advanced countries in software are continuously trying to establish the standard for quality evaluation. They are on the way to standardize ISO/IEC 9126 as the standard on quality evaluation features and ISO/IEC 14598 as the standard on quality evaluation process [1, 2].

However, it is the actual circumstances that it is very rare they build the specific quality evaluation method and then actually apply it, based on the general contents on standard. And there is a case that they build the practical evaluation system about application, a part of quality features, and then utilize it.

2.2 Domestic Trend

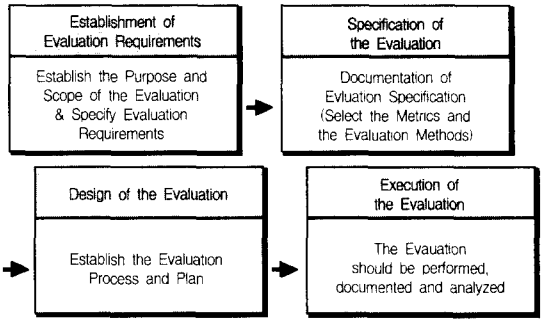
It can be said that now the domestic trend on quality evaluation & test technology has its weak basis on the whole [14].

The related standard for quality evaluation has not been established yet, and the authentication for software's quality system relies on foreign countries, and thereby we can see the basic study in domestic is very weak.

Even though domestic software industry regards technology for quality improvement and development of product evaluation technology as the urgent task, it has much difficulty in pushing technology development in itself.

3. Building test process for package Software acquirer

Having an eye to the point that package software are "off-the-shelf" software, this study introduced ISO/IEC 14598-4, the quality evaluation process for purchasers and built the test process for package software. The process of ISO/IEC 14598-4 can be summarized as shown in (Figure 1).

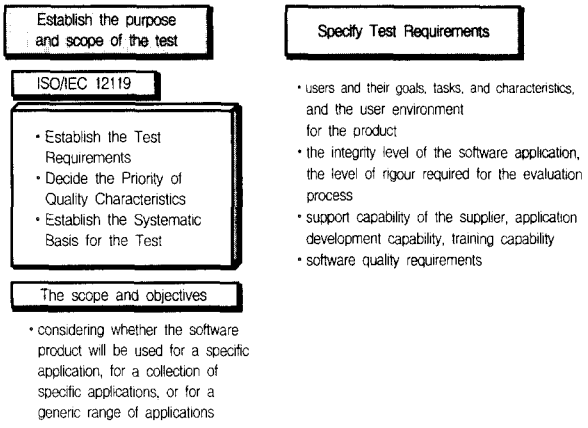


(Figure 1) ISO/IEC 14598-4 : Process for acquirers

This study built the test process like (Figure 1) that is applied to package software as follows.

3.1 Establishment of test requirements

This process means that it can be utilized at the stage of test by clearly defining the constitutional elements of package software such as product manual, user document and quality requirements for program & data.



(Figure 2) Establish the Test Requirements

In the process of test, software quality requirements, the priority of quality characteristics, the test criteria, the test scope and objectives should be set up.

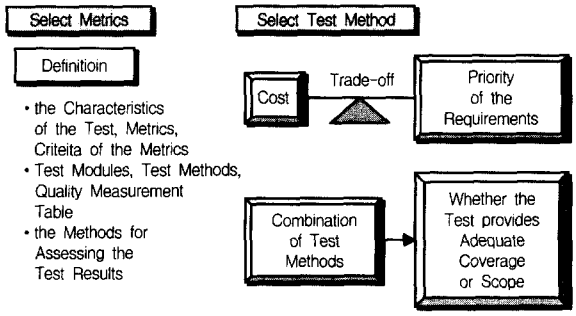
Specification of the test requirements includes users and their goals, tasks, characteristics, the user environment for

the product, the integrity level of the software application, software quality requirements.

3.2 Specification of the test

The test details should be documented in order for an appropriately qualified manager to repeat the test process with the repeatable results.

(Figure 3) shows the activities that are executed in the process of test specification in details.



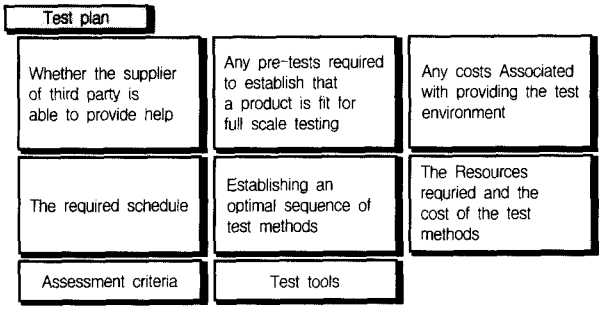
(Figure 3) Specification of the Test

In the process of test, we have to setup the characteristics of the test, metrics for the quality test, sufficient criteria for metrics to describe their acceptable range, any packaged test modules, quality measurement tables to be answered by the test, the methods for assessing the test results.

3.3 Design of the Test

Test design is the process to establish test procedure and plan, and it executes the activities as shown in (Figure 4).

The test plan should identify the required schedule, the resources required and the cost of the test methods, the procedures for developing and validating metrics, and for standardizing the test process, metrics and measures.



(Figure 4) Design of the Test

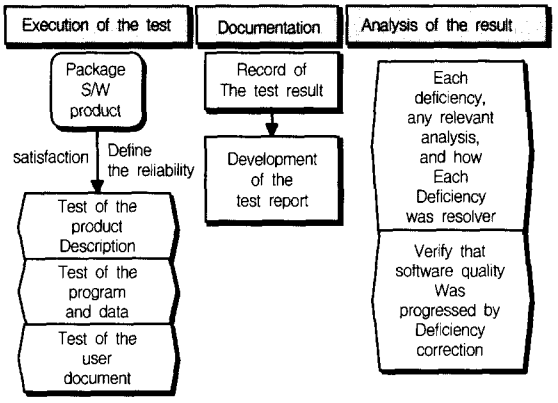
3.4 Execution of the Test

ISO/IEC 12119 defines the constitutional elements for package software like <Table 1>.

<Table 1> Constitutional Elements of Package Software

Constitutional elements	Concepts
Document for requirements	It is the document that includes recommendation, requirements or restrictions that package software should satisfy
Product Manual	This is a document that explains software's features, and it enables purchasers themselves to evaluate a product's suitability before purchasing. A product manual regulates the features or contents of a product, and provides information on user document, program and data as a part of package documents.
User document	It is provided for product use as the printed or non-printed all forms of documents, and it should include all information necessary for product use.
Program & data	It is an execution program provided by one or more media and the related data.

Test includes the execution processes of testing package software constitutional elements and documenting and analyzing it as shown in (Figure 5).



(Figure 5) Execution of the Test

3.4.1 Test record

This process includes the followings to be recorded.

3.4.2 Making up test report

It is the process of making up test report, and the test report should have the following structure.

- Identification of product
- Construction used(hardware & software)
- Document used
- Result of the test
- Problem list

3.4.3 Analysis of test result

It reviews each defect, comparative analysis and solution to each defect. The solution to defects includes the followings.

- One of the other test methods has provided assurance that the deficiency is not major
- A satisfactory "work-around" can be found to alleviate the impact of the deficiency
- The original requirement is not mandatory and the deficiency can be accepted
- The deficiency is acceptable provided that the use of the software product will be controlled by specific conditions or limitations

3.5 Test conclusion

The process of test conclusion includes the following executions.

- Verify that software product satisfy requirement and integrity level
- Decide whether buy it or not

4. Quality model

In order to apply ISO/IEC 12119 to package software test, the quality model, which each item consisting of package software is to be applied to, should be organized.

4.1 Quality model on product manual

Quality model on product manual among the constitutional elements of package software includes the items such as functionality, reliability, application, effectiveness, maintenance and graft, and those can be summarized as shown in <Table 2>.

〈Table 2〉 Quality Model about Product Manual

Quality Model	Concept
Functionality	Summary of functions, region value, security information
Reliability	Information for data storing process
Usability	User interface form, knowledge for product usage, identification of usage condition
Efficiency	Response time, processing rate
Maintainability	Explanation about maintainability
Portability	Explanation about Portability

4.2 Quality model on user document

〈Table 3〉 Quality Model about User Document

Quality model	Concept
Completeness	Product usage information, region value, installation-maintenance manual
Correctness	Correctness of document information, clearness of expression
Consistency	Integrity between documents, terms consistency
Understandability	User group have to understand
Easy summary	Easy summary about user document

Quality model on user document among the constitutional elements of package software includes the items such as perfection, exactness, consistency, understanding and easy summary, and those can be summarized as shown in <Table 3>.

4.3 Quality model on program and data

Quality model on program and data among the constitutional elements of package software includes the items such as function, reliability, application, effectiveness, maintenance and graft, and those can be summarized as shown in <Table 4>.

〈Table 4〉 Quality Model about Program and Data

Quality model	Concept
Function ality	• can Install according to the manual • similar to all explanation in other document • not conflict with other documents • must be executed as specification
Reliability	• always controllable • data is not destructed
Usability	• understandability about all information of program • adequacy of error message information
Efficiency	• the explanation about efficiency is suitable
Maintainability	• the explanation about maintainability is suitable
Portability	• the explanation about portability is suitable

5. Development of package software evaluation metrics

Evaluation metric for package software has the basis of ISO/IEC 12119, and this study abstracted the metric items that are applicable to package software from ISO/IEC 9126-2, 3, and modified and supplemented them. The details of developed Metric items are as shown in <Table 5>.

〈Table 5〉 Development Specification of Package Software Test Metrics

Type of Metrics	The number of Metrics	Remark
General requirements	10	Metrics about Identification and order
Product manual	20	Metrics about functionality, reliability, usability, efficiency, maintainability, portability
User document	12	Metrics about completeness, correctness, consistency, understandability, easy summary
Program & data	61	Metrics about functionality, reliability, usability, efficiency, maintainability, portability

5.1 Metric Index Table

This study built the Metric Index Table by product element consisting of package software, as shown in <Table 6>. The Metric Index Table on general requirements for package software is shown in the Table.

〈Table 6〉 an Example of Index Table

Characteristics	Metric index	Type	Reference
Identification and order	1.1 identification of product manual	Y/N	ISO/IEC 12119
	1.2 identification of product	Y/N	ISO/IEC 12119
	1.3 Specification of supplier	Y/N	ISO/IEC 12119
	1.4 Specification of Work	Y/N	ISO/IEC 12119
	1.5 Document for adequacy requirements	Y/N	ISO/IEC 12119
	1.6 Required system	Y/N	ISO/IEC 12119
	1.7 Product configuration	Y/N	ISO/IEC 12119
	1.8 Installability	Y/N	ISO/IEC 12119
	1.9 Support	Y/N	ISO/IEC 12119
	1.10 Maintainability	Y/N	ISO/IEC 12119

5.2 Construction of Metric Table

An example of Metric that is developed for the purpose of testing package software by product element is as shown in <Table 7>. The example of metric on suitability in quality characteristics is shown in the Table. Metric Table was developed, based on ISO/IEC 12119, and it was modified as suitable one for package software test by introducing some relevant items from ISO/IEC 9126-2, 3.

〈Table 7〉 a Sample of Metrics Table about Suitablity in Functionality

Metric		exception processing for input which get out of boundary value
Boundary Value Processing Rate		
Measurement Item	A	the number of item have to be examined boundary value
	B	sum of the each test case success rate
Calculation Equation	• Boundary Value Processing Rate = B/A • $B = \sum_{i=0}^{A} \frac{Success\_TC_i}{Total\_TC_i}$ • Success_TC : the number of successful test case which is executed to examine boundary value processing function • Total_TC : the number of test case which is executed to examine boundary value processing function	
Result Region	0 ≤ Result ≤ 1	Result Value
Problem		

5.3 Decision of the evaluation marks level and judgment standard on Metric value

If the result value intends to have the meaning, it needs to decide the evaluation marks level on Metric value.

A : excellent : satisfy all requirements
B : good : satisfy almost requirements
C : fair : not satisfy a part of requirements
D : poor : not satisfy requirements

First, we define the evaluation marks level by deciding the number of range that Metric value has. The following example shows the case that defines 4 evaluation marks levels.

We can decide the range corresponding to evaluation marks level on each Metric value as follows.

Measurement value 0 <= X <= 1
X < 0.6 : rating level D
0.6 <= X < 0.7 : rating level C
0.7 <= X < 0.8 : rating level B
0.8 <= X : rating level A

Since the range of Metric measurement value is not always fixed, we decide it by considering the range of measurement value on each Metric.

In this way, we can score evaluation marks according to evaluation marks level on each Metric, and if it acquires a certain level of evaluation marks, we get the final result by deciding the criterion to pass or fail. For example, supposing that they decide to purchase if the number of Metric, of which the evaluation marks level is B or above, is 95% or more, and if the test is applied to the several software as objects, we can decide to purchase the software that acquired the best result.

5.4 Construction of Evaluation Module

〈Table 8〉 Construction of Evaluation Module

Items	Detailed Items	Concept
Outline	Concept of metric	Fundamental concept of evaluation module
	Measurement object	What you want to get by measurement of evaluation modules
	Metric category	Position of metric
	Term explanation	Term explanation related to metrics
Application scope	Application object and necessary resources	objects such as document or program
	Method	Applicable test method
	Consideration facts in applying	Related information to be considered in applying evaluation modules
Reference documents	Documents related to metrics	
Metrics	Measurement items	Data items to be measured
	Measurement method	Description of concrete measurement methods for measurement items constructing metric
	Calculation expression	Definition of calculation expression using data items
Application process	Detailed process	
Analysis and report of result	Mapping of measurement value	Region of metrics result value (Y, N, NA or value)
	Analysis of measurement result	Proposal of guideline for analysis method of measurement result
	Report facts	Specification of report facts using document for measurement result

An example of Evaluation Module that is developed for the purpose of testing package software by product element

is as shown in <Table 8>.

5.5 Test Case Table

Test case table is list of test case to measure for measurement items using quality measurement table. For example, test case table of measurement items for metric <boundary value processing ratio> in quality measurement table of <Table 8> is like <Table 9>.

〈Table 9〉 An Example of Test Case Table

No	Boundary Value	Explanation	Boundary Value Processing Ratio	Note
1	Transmission interval of packet	Input only recommendation item value 110, 220, 330	NA	
2	Maximum scale of packet	Default value 22050, minimum value 512, maximum value 30720	Y	
3	The number of screen division	It is possible to set to 100	Y	
4	Search time of one man	Recommendation item value 60, 80	Y	
5	Waiting time for login	Default value 20	Y	
No. of Y			4	
No. of N			0	
Result				

5.6 Test Result

〈Table 10〉 An example of test result

Product Description and User Document			
Quality Characteristics	Subcharac-teristics	metrics	Measurement value
Functionality	Suitability	Function information present	0.7
		Function implementation completeness	0.75
		Boundary value information present	0.85
		Boundary value processing ratio	0.77
	Correctness	Function implementation correctness information present	0.93
		Function implementation correctness	0.86
	Interoerability	Data exchange information present	Y
		Data exchageability	1.00
	security	Access control information present	Y
		Accessibility	0.50
		...	...
.	...	...	...

If measurement is implemented for quality measurement table using test case of test case table, measurement result of each metric can be produced. These results is documented to test result such as <Table 10>.

5.7 Certification Assessment

Test report which is draw up based on the test result is transmitted to the certification committee meeting. Certification committee inspect test report and various certification factors and decide to certify or not.

It is necessary to set and specify certification factors and document for application method for objective certification.

6. Quality Measurement and Evaluation Example

In this evaluation example, we implemented evaluation for package software of A company and described plembles and improvement method by evaluation example.

6.1 Measurement Result for Metrics

〈Table 11〉 Evaluation Result of Internal Characteristics about Reliability

Characteristics	Subcharacteristics	Metrics	Result
Functionality	Suitablity	Function Informaition Provision	0.92
		Function Implementation Completeness	0.95
		Boundary Value Information Provision	0.20
		Boundary Value Processing Rate	1.00
	Correctness	Function Implementation Correctness Information Provision	0.77
		Function Implementation Correctness	0.92
	Interoperability	Data Exchange Information Provision	Y
		Data Exchangeability	1.00
	Security	Access Control Information Provision	N
		Access Controllability	0.50
		Access Audit Information Provision	Y
		Access Auditability	0.50
	Conformance	Function Standard Conformance Information Provision	NA
		Function Standard Conformance Rate	NA
		Interface Standard Conformance Information Provision	Y
		Interface Standard Conformance Rate	1.00

We implemented the evaluation for functionality, reliablity, usability, efficiency, maintainability, portability, showed measurement result for functionality in <Table 11>. we can see result of each metric by measurement result, know weak characteristics relatively.

Boundary value processing rate, data exchangeability and interface standard conformance is showing good result. Boundary value information provision, access controllability and access auditability is showing low value. Function standard conformance information provision and function standard conformance rate can not be applicable by document unpreparedness.

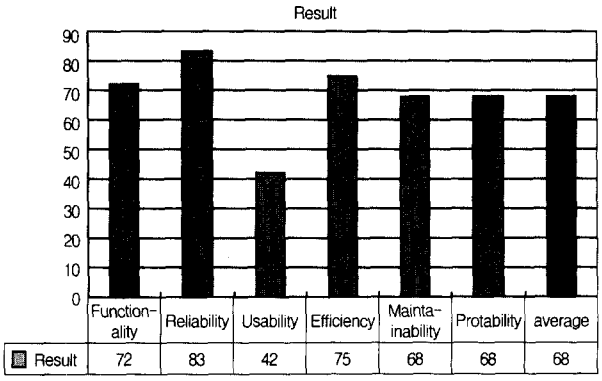
6.2 The Result Summation of Quality Subcharacteristics and Quality Characteristics

<Table 12> is the summation result of quality subcharacteristics. The summation of quality subcharacteristics is an average of sum of metric values for each subcharacteristic from the result of <Table 11>. In the metrics which is mesaured in Y/N, we regard Y as 1 and N as 0 and NA as 0, because NA means unpreparedness of document. We can see the subcharacteristics with weak result by result.

〈Table 12〉 Summation Table for Quality Subcharacteristics

Characteristics	Subcharacteristics	Result
Functionality	Suitablity	0.77
	Correctness	0.85
	Interoperablity	1.00
	Security	0.50
	Conformance	0.50
Reliability	Maturity	0.75
	Error tolerance	0.95
	Recoverability	0.87
	Conformance	0.75
Usability	Understandabiltiy	0.43
	Learnability	0.40
	Operabiltiy	0.18
	Preference	0.34
	Conformance	0.75
Efficiency	Time Behavior	1.00
	Resource Utilization	0.50
	Conformance	0.75
Maintainability	Analyzability	0.40
	Changeablity	0.81
	Testability	0.75
	Conformance	0.75
Portablity	Adaptability	0.50
	Installability	0.42
	Alternity	0.75
	Coexistence	1.00
	Conformance	0.75

(Figure 6) is the percentage for the quality characteristics of <Table 12>.



(Figure 6) Summation Chart for Quality Characteristics

6.3 Presentation of Problems

Because an object of quality evaluation is the quality improvement by problems analysis, we showed problems of software at <Table 13> in quality characteristics level.

〈Table 13〉 A Part of Problems

Contents of Test Result	
Test Object : Product description, User document	
Functionality	• lack of information about function • implementation method of error for data transmission
Reliability	• methods to solve problems • lack of information about problems
Efficiency	• lack of information for effect of environment setting
Usability	• Description of systematic work implementation method for every function • absence of information have to know to use product in advance
Maintainability	• detailed explanation for environment setting method • lack of information for facts to be changed by environment setting change
Portability	• offer of information related to the installation • detailed explanation for supportable OS & specification • lack of detailed explanation related to the installation

7. Conclusion

This study built quality test process for package software and developed Metric for testing package software by attempting to graft product evaluation process for purchasers in ISO/IEC 14598-4 into the standard of quality test for package software in ISO/IEC 12119, considering the features of package software.

If we firmly build evaluation system for package software

with basis of the process for purchasers in ISO/IEC 14598-4, it is considered that we can build the effective evaluation basis for package software types that are made by many development organizations.

Regarding the study works after this, it needs to specify measurement methods on measured items of test Metric for package software, and push to develop effective quality test through tools.

References

[1] ISO/IEC 9126, "Information Technology-Software Quality Characteristics and metrics-Part 1, 2, 3," Nov., 1997.

[2] ISO/IEC 14598, "Information Technology-Software product evaluation-Part 1, 2, 3, 4, 5, 6," Nov., 1997.

[3] ISO/IEC 12119, Requirements for Quality of Commercial Off-the-Shelf Software Product(COTS) and Instructions for Testing, July, 2002.

[4] Moller, K. H. and Paulish, D. J., "Software Metrics," Chapmen & Hall(IEEE Press), 1993.

[5] Wallmuller, E., "Software Quality Assurance A practical approach," Prentice Hall, 1994.

[6] 水野幸男, "ソフトウエアの綜合的品質管理", 日科技連出版, 1993.

[7] 吉澤. 東. 片山, "ソフトウェアの 品質管理と生産技術", 日本規格協會, 1990.

[8] N. F. Schneidewind, "Methodology for Validating Software Metrics," IEEE Trans. on SE. Vol.18, No.5, May, 1992.

[9] Moller, K. H. and Paulish, D. J., "Software Metrics," Chapmen & Hall(IEEE Press), 1993.

[10] Hae-Sool Yang, Ha-Yong Lee, "Design and Implement of Quality Evaluation Toolkit in Design Phase," KISS Paper(C), Vol.3, No.3, Jun., 1997.

[11] Hae-Sool Yang, Quality Assurance and Evaluation of Hanjin Shipping New Information System, Hanjin Shipping co., 1998.

[12] Hae-Sool Yang, Development of Software Product Evaluation Supporting Tool, ETRI Computer Software Technology Institute, 1999.

[13] J. Boegh, S. De Panfilis, B. A. Kitchenham, A. Pasquini : A Method for software Quality Planning, Control, and Evaluation, IEEE Software, Vol.16, No.2, Mar./Apr., 1999.

[14] Hae-Sool Yang, Study on Enhancement of Software Test Evaluation Modules, ETRI Computer Software Technology Institute, 2001.

[15] Hae-Sool Yang, Study on Development of Embedded Software Quality Evaluation Model, Telecommunications Technology Association, 2002.

[16] Hae-Sool Yang, Development of Evaluation Criteria for Medical S/W, Korea Food & Drug Administration, 2002.

이 하 용

e-mail : lhyazby@hotmail.com

1993년 강원대학교 전자계산학과(이학사)

1995년 강원대학교 대학원 전자계산학과 소프트웨어공학 전공(이학석사)

1995년~2002년 한국S/W품질연구소 선임연구원

1996년~1997년 경희대학교 전자계산공학과 강사

1996년~2000년 경원대학교 전자계산학과 강사

2001년~2002년 선문대학교 컴퓨터정보학부 강사

2001년~현재 호서대학교 공과대학 강사

2002년~현재 호서대학교 벤처전문대학원 박사과정

관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 객체지향 프로그래밍, 객체지향 분석과 설계, CBD S/W개발방법론 및 품질평가)

황 석 형

e-mail : shwang@sunmoon.ac.kr

1991년 강원대학교 전자계산학과(이학사)

1994년 일본 오사카대학교 대학원 정보공학과(공학석사)

1997년 일본 오사카대학교 대학원 정보공학과(공학박사)

1997년~현재 선문대학교 컴퓨터정보학부 부교수

2001년~현재 국방대학교 국방정보화사업 관리과정 외래강사

2001년~현재 일본 OGIS-RI Co. LTD. Certified UML Engineer

관심분야 : 객체지향 소프트웨어 시스템의 재구성 및 재이용, UML, Design Pattern, Adaptive Programming 기법, Formal Method 등

양 해 술

e-mail : hsyang@office.hoseo.ac.kr

1975년 홍익대학교 전기공학과(학사)

1978년 성균관대학교 정보처리학과 정보처리전공(석사)

1991년 일본 오사카대학교 정보공학과 소프트웨어공학전공(공학박사)

1975년~1979년 육군중앙경리단 전자계산실 시스템 분석장교

1986년~1987년 일본 오사카대학교 객원연구원

1980년~1995년 강원대학교 전자계산학과 교수

1995년~2002년 한국소프트웨어품질연구소(INSQ) 소장

1999년~현재 호서대학교 벤처전문대학원 교수

2000년~현재 한국정보처리학회 부회장

2003년~현재 미국 ACIS학회 Vice President

관심분야 : 소프트웨어공학(특히, S/W품질보증과 품질평가, 품질감리, 품질컨설팅, OOA/OOD/OOP, CASE, SI), 컴퍼넌트 기반 개발방법론, IT 품질 경영