

Study for Design of Analysis Tool for Improvement of Requirements Reliability and Satisfaction

Lee Eun-Ser[†]

ABSTRACT

Project success is depend on requirement analysis for in the software engineering. Requirements error have a effect in the whole system. As a result, the customer satisfaction will deteriorate. Therefore, we are need to tool that stakeholder's opinion exchange and modify for a accurate analysis in the requirement phase. In this paper, we are design that tool of the stakeholder's opinion exchange.

Keywords : Requirement Analysis, Customer Satisfaction, Customer Based on Design

요구사항 정의의 신뢰성과 만족도 향상을 위한 분석 도구 설계에 관한 연구

이 은 서[†]

요 약

소프트웨어 공학에서 요구사항 분석은 전체 시스템의 성공률을 좌우한다. 요구사항에서 발생하는 오류는 전체 시스템에 영향을 주게 되고, 그 결과 고객의 만족도가 낮아진다. 따라서 요구사항 단계에서 정확한 분석을 위하여 이해관계자 간의 의견을 교환하고 수정할 수 있는 도구가 필요하게 된다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 이해관계자 간의 의견을 교환할 수 있는 도구를 설계하고자 한다.

키워드 : 요구사항 분석, 고객 만족, 고객 중심 설계

1. 서 론

소프트웨어를 개발하며 고객의 요구를 만족시키는 시스템을 만들기 위해 고객의 요구사항을 알아내는 것은 시스템을 개발하는 관리자와 실무자에게 중요한 일이며 가장 큰 도전이다. 요구사항 분석의 오류를 줄이기 위해 프로젝트 초기인 요구사항 분석 단계에서부터 다양한 이해관계자를 프로젝트 개발 과정에 참여시키는 것이 중요하며, 이해관계자의 활발한 참여는 잘못된 요구사항을 밝혀내고 중요 요구사항 누락을 방지하는 데 필수적이다[1, 3, 7].

요구사항 단계는 분석과정을 거쳐서 시스템화 하기 위한 필수 단계이다. 따라서 요구사항의 중요성은 영역을 얼마나 이해하고 프로젝트를 시작하고 고객의 요구사항을 얼마나 파악하고 있는가에 따라서 시스템의 신뢰성과 고객의 만족도의 완성도가 달라진다.

고객 또는 프로젝트의 결과물을 사용할 사용자는 프로젝트의 이해관계자 중 가장 중요한 부류이다. 이들의 참여를 통해 시스템을 사용할 사람들이 누구이고 어떤 역할을 담당하며, 그 역할을 위해 어떤 업무를 수행하는지 이해하게 된다[1, 4, 8].

요구사항 분석 중 또 하나의 중요한 요소는 요구사항을 제시하거나 개발할 구성원 간의 의사소통으로서 서로가 의도하는 바를 명확히 이해하도록 하는 것이다. 이와 같은 과정은 프로젝트의 비전을 공유하며 전체적인 시스템의 연동을 가능하게 한다.

소프트웨어 공학의 업무는 인간의 노력이 집중된 것이라는 것을 잊어버리는 관리자는 프로세스 관리에 결코 성공할 수 없다. 프로젝트 초기에 이해관계자들 간에 의사소통을 하지 않는다면 해결책 자체는 좋으나 엉뚱한 문제에 대한 해결책을 제시할 수 있는 위험이 있다[2]. 따라서 본 논문에서는 요구사항의 신뢰성 있는 분석을 하기 위하여 신뢰성을 위한 요구사항 정의 모델을 제시하고 도구를 설계한다. 본 논문은 2장 기반연구와 3장 본론, 4장 적용 및 사례로 구성 되어 있다.

* 이 논문은 2015학년도 안동대학교 연구비에 의하여 연구되었음.

[†] 종신회원 : 안동대학교 컴퓨터공학과 부교수

Manuscript Received : October 21, 2015

Accepted : November 16, 2015

* Corresponding Author : Lee Eun-Ser(eslee@anu.ac.kr)

2. 기반 연구

2.1 요구사항의 분석

소프트웨어 개발에 있어서 가장 어려운 부분은 무엇을 개발할 것인가를 정확히 결정하는 일이다. 설계와 구현, 테스트 등은 잘못되었을 때 적은 비용으로 개정할 수 있으나 요구 분석의 변경은 처음부터 다시 개발하는 비용이 들기 때문이다[5]. 요구사항 분석의 오류는 이와 같은 많은 위험 요소를 가지고 있다. 많은 내포된 오류들은 설계와 구현, 테스트 등 많은 개발 단계에 내재되어서 위험요소로 전이되어 프로젝트의 실패 요인이 된다.

요구분석은 크게 두 가지 단계로 구성된다. 현재의 상태를 파악하고 요구를 정의한 후, 문제 해결과 구현될 시스템의 목표를 명확히 도출한다. 이를 분석 단계라고 한다. 다음 단계는 명세서 작성 과정이다. 완성될 소프트웨어가 어떤 기능을 가져야 하는가를 정확히 기술하는 일이다[5, 9, 10].

요구분석은 문제를 분석하여 고객에게 확인하는 단계를 개발 단계에 추가하여 고객의 만족도를 높이려는 노력을 하고 있다. 따라서 고객의 요구사항의 정확한 분석과 만족도의 향상, 개발 시스템의 오류를 최소화 할 수 있게 된다.

2.2 요구사항 타당성 검증

요구사항 분석은 신뢰성을 검증해야 시스템의 완성도를 높일 수 있다. 따라서 요구사항에 대한 타당성 검증은 이와 같은 목적을 위하여 필요한 사항이 된다.

요구사항 타당성 검증 단계에서는 명세된 요구사항의 구현 가능성, 명세 표현의 정확성 및 완전성, 표준과의 일치성, 요구사항 간의 충돌, 기술적 결함 등에 대한 검증을 수행한다. 이 단계에서는 시스템 요구사항이 설계 기준에 따라 하드웨어 형상 항목, 소프트웨어 형상 항목 등에 적절하게 할당되고 안전, 보안 및 위험성과 관련된 소프트웨어 요구사항이 정확하다는 것을 보여줄 수 있어야 한다[6, 11].

요구사항의 타당성은 체계적이며 이해관계자와의 원활한 소통을 통하여 이루어져야 타당성 검증의 기준을 설정할 수 있다. 이와 같은 내용을 위하여 본 논문에서는 의사소통을 원활히 하여 고객의 요구사항을 이해하고 이를 기능화하기 위한 도구를 설계하였다.

2.3 결함 제거 효율(DRE: Defect Removal Efficiency)

프로젝트 수준과 프로세스 수준 모두에 유익한 품질 계량은 결함 제거 효율이다. 근본적으로 결함 제거 효율은 전반적인 프로세스에 걸친 품질 보증과 품질제어 행위에서 걸러내는 능력에 대한 측정이다. 프로젝트들은 총체적으로 고려할 때, 결함 제거 효율은 다음과 같이 정의된다[2].

$$DRE = E / E + D$$

E : 소프트웨어가 최종 사용자에게 배포되기 이전에 발견된 오류의 수
 D : 배포된 후에 발견된 결함의 수

이상적인 DRE의 값은 1이다. 소프트웨어에서 결함이 발견되지 않았다는 것을 의미하며 현실적으로 D는 0보다 크고 E가 주어진 D의 값에 따라 증가하므로 DRE의 값은 역시 1에 가까워진다. E의 값이 증가하면 D의 최종 값은 감소한다. 품질 제어 및 품질 보증 활동의 걸러내는 능력에 대한 지표로 계량(metrics)이 사용될 경우, DRE는 소프트웨어 프로젝트 팀이 소프트웨어의 배포 이전에 가능한 한 많은 오류를 걸러낼 수 있게 하는 기법을 제정하도록 유도한다.

또한 DRE는 프로젝트의 수행 중에 오류들이 다음 단계, 또는 다음 소프트웨어 공학 활동에 넘어가기 전에 찾아내는 능력을 평가하는 데에도 사용될 수 있다[12, 13].

$$DRE = E_i / E_i + E_{i+1}$$

E_i : 소프트웨어 공학 활동에서 i에서 발견된 오류의 수
 E_{i+1} : 소프트웨어 공학 활동에서 i+1에서 발견된 오류의 수

i+1 단계에서 발견된 오류들은 소프트웨어 공학 활동 i에서 발견되지 않은 오류라고 추적할 수 있다.

3. 본 론

3.1 요구사항 정의의 신뢰성과 만족도 향상을 위한 모델

요구사항 추출의 목적은 고객의 요구사항과 부합하는 기능을 정확히 구현하기 위함이다. 이와 같은 요구사항의 추출은 고객의 만족도와 밀접한 관계가 있다. 프로젝트 초기에 요구사항 추출을 위한 활동을 하지 않으면 전반적인 시스템의 완성도와 고객이 원하는 기능이 구현되지 않는다. 요구사항 추출은 그 자체가 완전한 형태가 아닌 것으로서 타당성에 대한 검증이 필수적으로 요구된다. 따라서 요구사항 정의를 위하여 요구사항 추출을 선행하여야 하며 이 과정에서 여러 사람의 의견을 반영하여 검증을 할 수 있어야 한다. 추출된 사항에 대하여 다른 이해관계자들이 완전히 이해를 해야 추출된 사항에 대하여 검증을 수행할 수 있게 된다. 따라서 검증된 결과에 의하여 요구사항을 정의할 수 있게 된다. 이와 같은 과정을 위하여 추출된 항목에 대하여 지속적으로 이해관계자들 간의 의견 교환이 필요하게 된다. 지속적인 의견 교환은 추출된 항목을 요구사항으로 정의할 수 있는 근거와 검증자료로 활용된다. 다음 Fig. 1은 신뢰성 있는 요구사항을 정의하기 위한 모델이다.

제안하는 모델은 전체 5개의 단계로 구성이 되어있다.

1) 아이템 제시 단계

아이템 제시 단계는 개발하려는 시스템이나 프로젝트에서 기능으로 구현되기 위한 요구사항을 제시하는 단계이다. 제시되는 아이터은 반드시 개발 목적과 목표를 이해한 후에 제시가 되어야 한다. 또한 아이터이 개발하려는 시스템의 구성요소에 부합되는 내용을 포함해야 한다.

2) 아이터 설명 단계

아이터 설명 단계는 제시된 아이터를 다른 이해관계자들

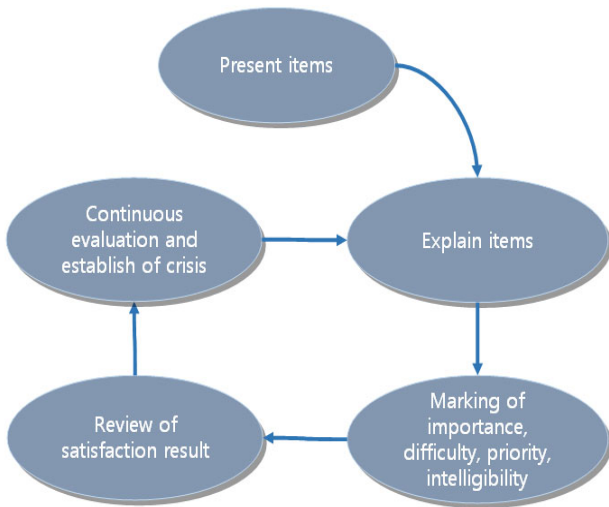


Fig. 1. Reliability for Model of Requirement Definition

에게 이해시키기 위하여 내용을 설명하는 단계이다. 아이템의 설명 단계를 통하여 다른 이해관계자들이 제시된 아이টে을 기능화할 때 필요한 핵심기능인지 부가적인 기능 인지를 판단할 수 있는 근거가 된다. 따라서 이 단계는 아이টে이 요구사항으로 정의하기 위한 중요한 단계가 된다. 이때 아이টে에 대한 의문사항은 질문을 통하여 모호성을 없애야 한다.

3) 중요도, 난이도, 우선순위, 이해도에 대한 의견 제시 단계
이 단계는 아이টে 설명 단계의 결과를 활용하여 제시된 아이টে이 전체 시스템에서 차지하는 중요도와 개발 난이도, 기능의 우선순위와 기능을 이해관계자들이 얼마나 이해하고 있는지를 확인하는 단계이다. 이 단계를 통하여 기능의 중요도, 난이도, 우선순위, 이해도에 대한 합의를 도출할 수 있게 된다.

각 항목에 대하여 백분율을 대상으로 가중치를 곱하여 정량적인 만족도를 산출하도록 한다. 전체 점수는 백분율을 기준으로 수행하며 가중치는 전체 시스템의 비중을 1로 산정하여 시스템에서 차지하는 비중을 할당한다. 가중치는 각 항목의 비중을 고려하여 상중하로 구분한다. 상은 1, 중은 0.5, 하는 0.3으로 할당하였다. 상, 중, 하를 기준으로 하여 각 항목에 적합한 가중치를 할당한다. 따라서 모든 항목의 가중치의 합은 1로 산정한다. 상, 중, 하의 기준은 다른 항목과 비교하여 가중치를 입력하게 되는 척도로 활용할 수 있다. 가중치에 대한 할당 기준은 개발하려는 내용의 목표에 따라서 가중치를 할당하게 된다.

중요도에 대한 가중치의 척도

- 상 - 해당 항목이 핵심 기능에 해당되고 다른 항목과 연동이 되는 경우
- 중 - 해당 항목이 핵심 기능이지만 다른 아이টে과 연동 기능이 없는 경우
- 하 - 해당 항목이 핵심 기능이 아닌 부가적인 기능에 해당하는 경우

난이도에 대한 가중치의 척도

- 상 - 다른 항목 또는 컴포넌트와 연동되고 제어 명령을 주고받는 경우
- 중 - 다른 항목 또는 컴포넌트와 연동되는 경우
- 하 - 다른 항목 또는 컴포넌트와 연동되지 않고 독립적인 기능을 수행하는 경우

우선순위(기능이 전체 시스템에서 차지하는 우선순위)

- 상 - 해당 항목이 핵심 기능에 해당되고 다른 항목과 연동이 되는 경우
- 중 - 해당 항목이 핵심 기능이지만 다른 항목과 연동 기능이 없는 경우
- 하 - 해당 항목이 핵심 기능이 아닌 부가적인 기능에 해당하는 경우

이해도(투표하는 사람이 해당 아이টে을 이해하고 있는지에 대한 정도)

- 상 - 해당 항목의 구현에 필요한 연동 아이টে과 구조 및 필요한 데이터를 완전히 이해하고 있음.
- 중 - 해당 항목 자체의 내용과 구조만 이해하고 있음
- 하 - 해당 항목의 내용만 이해를 하고 있음

4) 만족도 결과의 검토 단계

중요도, 난이도, 우선순위, 이해도에 대한 의견 제시단계의 결과를 활용하여 전반적인 만족도에 대한 결과를 도출하여 내용을 검증해야 한다. 만족도 결과를 도출하기 위하여 정량적인 산출 방식을 활용한다.

개발하려는 프로젝트의 특성에 따라서 기준을 중요도, 난이도, 우선순위, 이해도를 산출하게 된다.

각 항목과 만족도에 대한 산정은 다음과 같다.

$$\text{중요도} = 100 \times \text{가중치}$$

$$\text{난이도} = 100 \times \text{가중치}$$

$$\text{우선순위} = 100 \times \text{가중치}$$

$$\text{이해도} = 100 \times \text{가중치}$$

$$\text{만족도} = \text{중요도} + \text{난이도} + \text{우선순위} + \text{이해도}$$

산출된 만족도의 수준은 상, 중, 하로 구분하였다. 상은 80-100, 중은 60-79, 하는 59 이하인 경우로 구분하였다. 만족도의 각 수준의 의미는 핵심기능의 경우 완전히 만족함을 기준으로 하여 다음과 같이 구분하였다.

- 상 - 고객의 요구사항 정의서의 내용을 완전히 만족하고 추적성을 유지하는 경우
- 중 - 고객의 요구사항 정의서와 내용을 만족하는 경우
- 하 - 고객의 요구사항 정의서와 내용을 만족하지 못하는 경우

산출된 만족도를 기준으로 전반적인 고객의 요구사항 완성도를 측정할 수 있으며 구현되려는 시스템의 오류와 결함도 줄일 수 있게 된다.

5) 지속적인 평가와 재설정 단계

지속적인 평가와 재설정 단계는 이해관계자 간의 합의가 잘 이루어졌는지의 결과를 활용하여 각 단계들을 평가하여 합의를 도출할 수 있도록 평가와 아이টে에 대한 재설정, 중요도, 난이도, 우선순위, 이해도에 대한 의견 제시를 다시 하도록 한다. 이와 같은 단계는 합의가 이루어질 때까지 수행하도록 하여 전반적인 만족도를 높이고자 한다. 합의가 이루어지는 동안 지속적으로 평가와 재설정을 하지만 반복되는 사이클은 전반적인 프로젝트의 요구사항 정의 단계의 일정을 고려하여 수행하도록 한다. 따라서 프로젝트의 전반적인 일정에 차질을 발생하지 않도록 한다.

4. 적용 및 사례 연구

4.1 도구 설계

3장의 이론을 기반으로 하여 사례를 적용하고자 한다. 사례 적용을 위하여 유즈케이스를 분석하였다. 설계하려는 도구(이하 POLL이라고 함)의 유즈케이스를 위한 기술서는 다음과 같다.

- 사용자는 프로젝트의 기능에 대한 투표를 하려 한다.
- 투표를 하려면 입력을 해야 한다. 입력엔 투표자입력, 아이টে입력, 기능 설명이 있다.
- 투표를 할 땐 난이도 투표, 중요도 투표, 아이টে이해도 투표가 있다.
- 입력한 내용과 투표한 내용은 데이터베이스에 저장된다.
- 아이টে에 관한 이해가 부족할 때 아이টে 제안자에게 질문할 수 있는 메시지기능이 있다. 메시지기능은 아이টে에 관한 질의응답을 가능하게 하는 기능이다.
- 마지막으로 결과값이 산출되고, 사용자에게 보여준다.

Fig. 2는 유즈케이스 기술서를 기반으로 유즈케이스를 분석한 것이다.

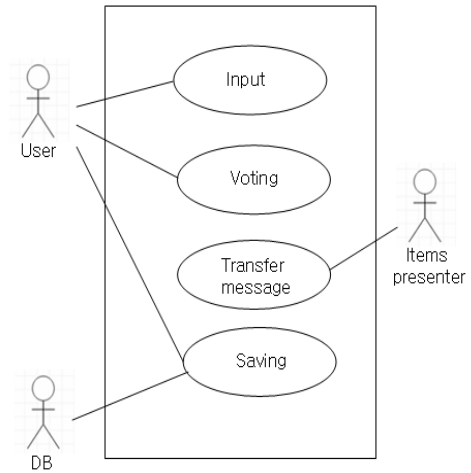


Fig. 2. Use Case Diagram

Fig. 3은 유즈케이스를 기반으로 아키텍처를 설계한 것이다. Fig. 4는 분석된 내용을 기반으로 사용자 인터페이스를 설계한 것이다. 다음은 각 항목에 중요도, 난이도, 이해도, 우선순위를 입력하는 화면이다.

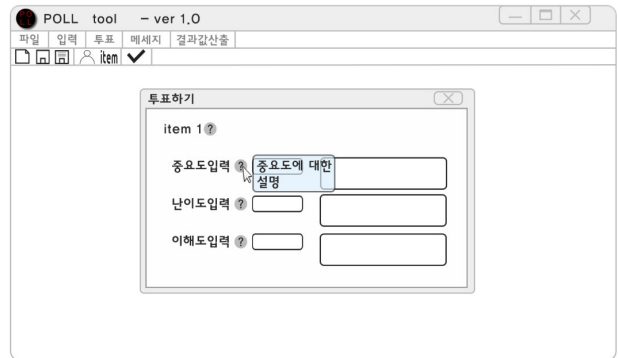


Fig. 4. Design of Importance, Difficulty, Intelligibility

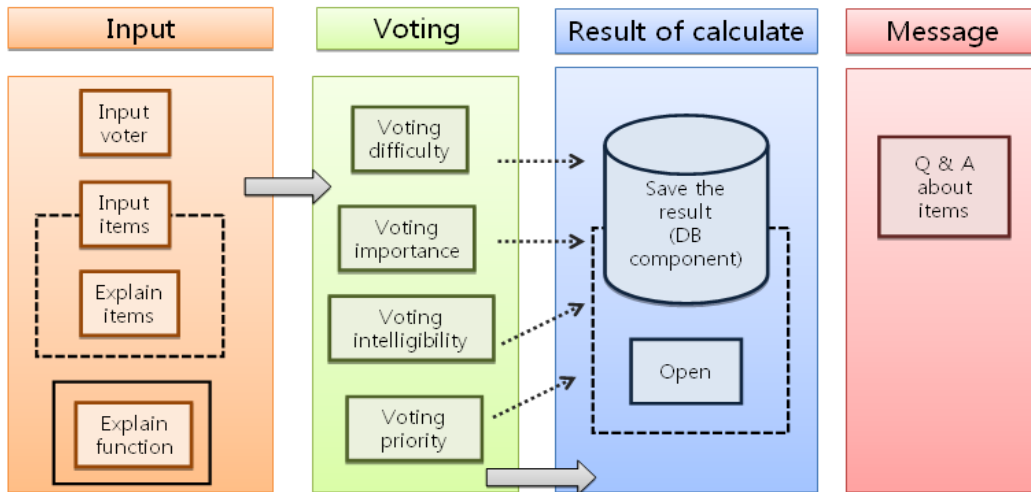


Fig. 3. Design of Architecture

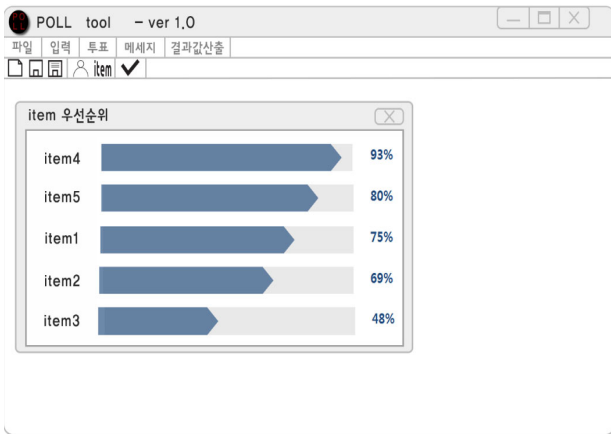


Fig. 5. Design of Priority

4.2 검증

4.1절에서 설계된 도구를 적용하여 효율성을 검증하고자 한다. 검증 방법은 2.3절에서 소개한 결함 제거 효율(DRE) 방법을 활용하였다.

검증 방법은 설계된 도구를 적용하기 전의 DRE와 도구를 적용한 후의 DRE 결과를 비교하고자 한다.

적용 분야는 다음과 같은 내용을 기반으로 수행하였다.

1) 개발 배경

- ① 스마트폰 사용자가 점점 증가하면서 노년층과 유아층 사용자도 함께 증가하고 있다. 그러나 이와 같은 기능의 사용자 앱은 찾기 어렵다.
- ② 소외계층의 사용자에게 카카오톡, 네이버 등을 설치하고 추가적으로 사용할 수 있는 앱을 제작하고자 한다.

2) 개발 목표

- ① 사전 준비 : 프로젝트 범위 및 세부일정 수립(애자일 방식)
- ② 사전 준비 : 현재 앱 마켓에서 구할 수 있는 유아용 앱 분석
- ③ 안드로이드 기본학습 : 프로그래밍 경험
- ④ 제품 패키지 및 테스트 수행 : 실무적 품질 관리 체험

먼저 도구를 적용하기 전의 요구사항 분석 단계와 설계 단계의 결함 개수를 산정하였다.

Table 1. Defect Number of General

	Ei	Ei+1
Requirement phase	10	10
Design phase	20	11

결함 개수를 기반으로 DRE를 산출하면 다음과 같다.

요구사항 분석 단계 : $10/10+10 = 0.5$

설계 단계 : $20/20+11 = 0.645$

다음은 같은 과제를 대상으로 POLL 도구를 활용한 후의 결함 개수를 산정하였다.

Table 2. Defect Number of Apply to POLL

	Ei	Ei+1
Requirement phase	10	3
Design phase	20	5

결함 개수를 기반으로 DRE를 산출하면 다음과 같다.

요구사항 분석 단계 : $10/10+3 = 0.769$

설계 단계 : $20/20+5 = 0.8$

결과를 분석하면, 도구를 적용한 경우에 요구사항 분석 단계의 DRE는 0.269 향상되었다.

설계 단계의 DRE는 0.155 향상되었다. 따라서 도구를 활용하여 결함을 줄일 수 있게 되었다.

5. 결론 및 향후 연구 방향

본 논문에서는 요구사항 정의의 신뢰성과 만족도 향상을 위하여 요구사항을 중요도, 난이도, 우선순위, 이해도로 구분하여 의사결정을 하였다. 따라서 고객의 만족도를 향상시키고 시스템의 완성도를 향상시키고자 했다.

향후 연구로는 설계된 도구를 안드로이드와 IOS와 연동할 수 있도록 개발할 필요가 있다.

References

- [1] 윤청, “이해하기 쉬운 소프트웨어 공학, 생능출판,” pp.406, 2014.
- [2] 김성규 외 6명, “소프트웨어 공학 실무적 접근 7판,” 도서출판 한산, pp.655, 2011.
- [3] Shari lawrence, “Software engineering forth edition,” Pearson, pp.208, 2010.
- [4] Sommerville, “Software engineering, Pearson,” pp.83, 2011.
- [5] 최은만, “소프트웨어 공학,” 정익사, pp.156, 2011.
- [6] 한혁수, “소프트웨어 공학의 소개,” 홍릉과학출판사, pp.117, 2011.
- [7] John Gaffney, “Some models for software defect analysis,” Lockheed martin, 1996.
- [8] L. hatton, “Is modularization always good idea,” *Information and Software Technology*, Vol.38, Issue.11, pp.719-721, 1996.
- [9] B. compton and C. withrow, “Prediction and control of ADA software defects,” *Journal of Systems and Software*, Vol.12, pp.199-207, 1990.
- [10] R. Balzer, N. Goldman, and D. Wile, “Informality in program specifications,” *IEEE Transactions on Software Engineering*, Vol.4, No.2, pp.94-103, 1978.

- [11] L. A Zadeh, "Test-score semantic as a basis for computationa approach to the representation of meaning," *Literacy & Linguistic Computing*, Vol.1, pp.24-35, 1986.
- [12] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, Vol.11, No.6, pp.48-56, 1994.
- [13] N. Juristo and A. M. Moreno, "Basics of Software Engineering Experimentation," in *Annex I*, Shari Lawrence Pfleeger, Kluwer Academic Publishers, 2001.



이 은 서

e-mail : eslee@anu.ac.kr

2001년~현 재 ISO/IEC 15504 국제 선임
심사원

2004년 중앙대학교 컴퓨터공학과(박사)

2004년~현 재 임베디드 산업협회 전문
위원

2004년~현 재 한국정보통신기술협회 위원

2015년~현 재 안동대학교 컴퓨터공학과 부교수

관심분야 : CBD, Formal method, Quality model, SPI(Defect
Analysis)