

PIECES 프레임워크 중심의 요구사항 정제와 우선순위 결정 전략

전혜영*, 변정원**, 류성열***

An Efficient Dynamic Workload Balancing Strategy

HyeYoung Jeon *, JungWon Byun **, SungYul Rhew ***

요약

급변하는 웹과 모바일 환경에서 사용자 요구사항을 효율적으로 찾아내고 올바르게 시스템에 반영하는 것은 매우 중요하다. 본 연구는 모바일 프로그램에서 사용되는 댓글, Q&A, 불편사항 신고 등의 사용자 요구사항을 기반으로 웹과 모바일 프로그램을 변경하기 위한 요구사항을 정제하고, 정제된 요구사항의 우선순위를 결정하기 위한 전략을 제안한다. 요구사항을 정제하기 위하여 소프트웨어사업선진화포럼의 표준화 방안, 기 개발된 프로그램 구성도를 활용하여 요구사항을 그룹화 하였으며, 이를 PIECES 에 맵핑하여, 정제된 요구사항이 시스템에 유효하게 반영될 수 있는지 여부를 확인하고 정제하였다. 정제된 요구사항의 우선순위를 결정하기 위하여 첫째, 소프트웨어의 구조, 요구사항, PIECES 카테고리에 상대적인 가중치를 부여하고, 둘째, 각 요구사항에 대한 통합 점수를 구하여 부문 및 전체 점수의 상대적인 값을 구한다.

제안한 기법의 가능성을 검증하기 위해서 S 대학교에서 서비스하는 모바일 애플리케이션의 변경 요구사항을 15 명의 업무 관련 이해당사자에게 설문한 결과로 그 유효성을 입증하였다.

▶ Keywords : 요구사항 정제, PIECES, 요구사항의 상대적 우선순위

Abstract

Identifying user requirements efficiently and reflecting them on the existing system is very important in a rapidly changing web and mobile environments. This study proposes the strategies to refining requirements and to prioritizing those refined requirements for changing of web and mobile application based on user requirements (e.g. mobile application comments, Q&A, reported information as discomfort factors). In order to refining the user requirements, those requirements are grouped by using the advancement of the software business of the Forum of standardization and the existing configuration-based programs. Then, we mapped them onto the PIECES

• 제1저자 : 전혜영 • 교신저자 : 변정원, 류성열
• 투고일 : 2012. 08. 20. 심사일 : 2012. 08. 29. 게재확정일 : 2012. 09. 19.
* 숭실대학교 컴퓨터학과(Computer Science and Engineering)

framework to identifying whether the refined requirements are correctly reflected to the system in a way of valid and pure. To determine the priority of refined requirements, first, relative weights are given to software structure, requirements and categories of PIECES. Second, integration points on each requirement are counted to obtain the relative value of partial and overall score of a set of software structural requirements.

In order to verifying the possibility and proving the effectiveness of proposing technique in this study, survey was conducted on changing requirements of mobile application which have been serviced at S University by targeting 15 people of work-related stakeholders.

▶ Keywords : Requirement Refinement, PIECES, Relative Prioritization of Requirement

I. 서 론

웹과 모바일이 시스템의 핵심적 역할을 수행하게 되는 것은 사용자의 영향력이 더욱 확대되고 있다는 것을 의미한다. 웹과 모바일 등을 통해 사용자의 참여 수단이 다양해짐에 따라 사용자도 적극적으로 자신의 의견을 표현하고 있다. 이러한 환경에서 사용자의 요구사항을 시스템에 올바르게 반영하는 것은 매우 중요한 일이다[1].

웹과 모바일은 하드웨어 기기의 급격한 변화와 발전에 따라 소프트웨어도 이에 빠르게 순응하지 않으면 도태되거나 외면 받을 수밖에 없다. 이러한 환경 요인에 순응하기 위해서는 사용자 요구사항을 인식하고 실행을 위한 준비를 자동화 하는 것은 필수적인 사항이다.

또한 다양한 사용자와 개별적인 수많은 요구는 시스템의 성능을 저하, 관리 운용의 효율성 저하를 유발하며, 과도한 자료로 인해 판단을 흐리게 할 수도 있으며, 시스템의 본질적 목적을 망각하게 할 수 있으므로 시스템의 운용 관리에 적합한 요구사항을 우선순위에 따라 선별 및 선정할 필요가 있다.

본 연구는 이와 같은 요구사항의 자동적인 인식을 위하여 요구사항의 우선순위를 결정하는 전략적인 방안을 제시하고자 한다.

첫째, 사용자의 요구사항을 자동적으로 인식하기 위하여 웹과 모바일 프로그램에서 사용되고 있는 댓글, 불편사항, Q&A 등의 사용자 의견들을 추출, 정리, 구조화하여 요구사항으로 작성한다[2].

둘째, 요구사항은 중복되거나 추상화 정도가 서로 다르므로 표현 방법이 다양하기 때문에 표준화되고 같은 기능들로 그룹화되기 위한 정제 과정을 가진다. 정제의 방법으로 다양한 접근 방법이 존재하겠지만, 본 연구에서는 J. Wethebe가 주장한 문제와 기회를 찾기 위한 전략 계획 프레임워크인

PIECES를 활용한다.

셋째, 정제된 요구사항을 설계 및 구현하기 위한 우선순위를 결정하기 위하여 요구사항들의 상대적인 비교 평가가 이루어져야 한다. 본 연구에서는 요구사항의 우선순위를 정하기 위한 방안으로 요구사항을 소프트웨어 구조, PIECES 카테고리에서 부여한다. 그리고 소프트웨어 구조 및 카테고리의 상대적 가중치를 결정하며, 동등한 구조 및 카테고리에 포함된 요구사항들끼리의 상대적인 평가를 실시한다. 이를 종합하여 요구사항의 우선순위를 제시한다.

요구사항의 자동화된 인식 및 우선순위 결정에 대한 연구는 변화하는 웹과 모바일 환경에서 선택이 아닌 필수적인 연구이다. 이 연구의 부족한 점을 지속적으로 보완하여 완성한다면 웹과 모바일 응용프로그램의 발전에 크게 기여할 것이다.

II. 관련 연구

1. PIECES

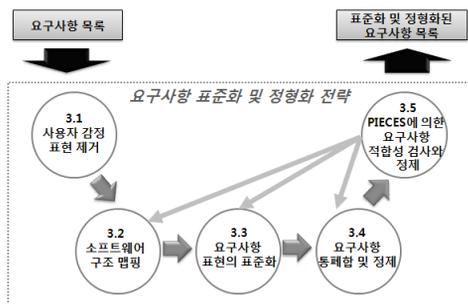
PIECES 프레임워크는 J. Wetherbe 에 의해 제안된 경영정보시스템의 계획 방법론 중 하나이다[2]. PIECES는 현재 존재하는 시스템의 문제점을 성능(Performance), 정보(Information), 경제(Economics), 통제(Control), 효율(Efficiency), 서비스(Service) 등의 측면에서 살펴보고 문제점을 개선함으로써 얻는 이득을 찾고자 하였다. 이를 각 카테고리의 첫 글자를 이용하여 PIECES 프레임워크라고 한다. PIECES 는 현 시스템의 문제나 기회를 각 카테고리의 측면으로써 바라볼 수 있게 한다. 이 카테고리는 경우에 따라 중첩될 수도 있지만, 각각은 서로 다른 관점을 제시한다. 카테고리는 서로 상호작용을 한다. 결과적으로, 특정 문제나 기회는 다수의 카테고리에 속할 수도 있고, 다수 문제가 하나의

카테고리에 속할 수도 있다. 본 연구에서 PIECES 프레임워크는 요구사항을 분류하는 기준으로써, 그리고 요구사항의 타당성을 판단하는 기준으로써 활용된다. 이는 본 연구에서 제시하는 전략의 핵심으로써 요구사항의 우선순위를 결정하기 위한 중요한 역할을 수행한다.

2. 요구사항 정형화 및 계층화

소프트웨어는 제품 및 서비스로서의 역할을 한다[3]. 사용자는 소프트웨어가 제공하는 다양한 이점을 얻기 위해 소프트웨어를 사용할 것이며, 개발/운영자는 이러한 이점을 소비자에게 제공하고 이를 통해 이윤을 창출하고자 소프트웨어를 개발/운영할 것이다[4]. 즉, 소프트웨어의 개발과 운영은 하나의 비즈니스적인 측면으로 바라봐야 한다. 생산자는 자신의 비즈니스를 유지하고 발전시키기 위하여 비전을 수립하고, 이에 적합한 범위를 설정하며, 소프트웨어는 이러한 비전과 범위를 준수해야만 한다[5].

요구사항의 정형화는 목표와 요구사항 사이의 관계를 설정하는 관점에서 볼 수 있다[6]. 이 관점은 요구사항을 목표에 부합하도록 정렬하는 것이다. 목표는 요구사항을 종합/통합한 개념이며, 목표를 근거로 요구사항이 추출된다는 주장을 뒷받침한다[7]. 요구사항의 정형화는 고품질의 요구사항을 명세하기 위한 사전 타스크로 볼 수도 있다. 이는 요구사항 명세 지침에서 제시한 좋은 요구사항의 특징을 반영하기 위한 타스크로 볼 수 있다[8].



(Figure 1) Requirements for standardization and formalization strategy

요구사항의 계층화를 위하여는 목표와 요구사항 사이의 텍사노미를 구축할 것을 제안하였다[9]. 이는 이후 Goal Oriented Requirements Engineering의 기반이 되었다[10]. 또 다른 방안으로 And/Or 그래프를 이용하여 요구사항을 계층화하는 기법을 제안하였다. 이 기법은 요구사항의 충돌을****는데 유용하였다. 이들 연구는 목표와 관련된 요구

사항의 기여도를 측정하는 다양한 연구의 기반을 제공하였다[11]. 그러나 Darimont는 요구사항의 분석 단계에서 추상성이 비교적 높은 요구사항을 수학/기호학 기반으로 작성하는 것이 분석가에게 요구사항 명세를 위한 추가적인 학습은 물론, 다른 이해당사자와의 의사소통을 저해한다고 주장하였다[12].

과거 모델링 기반의 정형화 및 계층화의 연구들은 근본적으로 요구사항의 명세 단계에 집중하고 있다. 요구사항을 명세화를 효과적으로 하기 위해서는 정형화된 요구사항을 추출해야 하는데 이때 요구사항과 목표 사이의 관계를 식별하는 방법에 초점을 두어야 한다.

3. 요구사항의 선정과 평가

요구사항을 선택하는 것은 한정된 자원을 투입하는 프로젝트를 통해 최대한의 가치를 창출하기 위해 중요하다[13]. 기존 연구들이 이해당사자의 합의를 통해 요구사항을 선택하는 기법을 제안하고 있으나, 이해당사자의 감정, 정치, 개인적 추측을 배제하고 의사소통할 수 있는 방안을 제시하고 있진 못하다. 이러한 의사소통을 위해 객관적인 정보와 합리적인 판단 기준이 필요하다.

소프트웨어의 개발과 운영은 프로젝트의 관점에서 보아야 한다. 이러한 프로젝트를 계획한 후 수행 이전에 프로젝트의 타당성 분석을 수행한다[14]. 요구사항 측면에서의 타당성 분석은 각 요구사항이 프로젝트에 적합한 요구사항을 선정 평가하는 과정이다[5]. 적합한 요구사항을 선정하기 위하여, 기존 연구들은 요구사항의 적합성을 측정하고 이를 기준으로 우선순위를 하였다[13].

요구사항의 품질 측정에 대하여, Alan D.는 다양한 척도를 종합하였다[15]. 본 연구의 범위 (목표와 타당성을 고려한 요구사항의 선정)에 적용 가능한 척도는 외적 일관성 척도이다. Alan 이 제시한 외적 일관성 척도는 요구사항 명세서와 다른 문서 사이의 일관된 정도를 비율로써 표현하는 척도이다. 그러나 개별 요구사항이 목표 및 타당성과의 관계 정도를 표현하는 척도를 제시하고 있지 못하다.

개별 요구사항 선택에 대한 다양한 기준들이 연구되었다. Covey는 선택 기준으로써 요구사항의 중요도와 위급도를 제시하였으며[16], Karlsson과 Ryan은 요구사항의 수용비용과 반영된 이후 얻어지는 가치를 제시하였다[17]. 그러나 이러한 기준은 객관적이고 정량적인 값으로 획득하기 힘든 한계점이 있다.

Liu는 다양한 기준을 고려하여 요구사항을 체계적으로 선정 평가할 수 있는 방법으로 SW-QFD를 제시하였다[18]. QFD의 중요한 요소인 HoQ를 소프트웨어 요구공학 측면으

로 개선한 이 연구는, 개별 요구사항에 다수의 기준을 적용하여 판단할 수 방안을 제시하였다. 그러나 요구사항의 개수가 많아질수록, 의사결정권자가 제시된 기준들을 일관되게 판단하기는 힘든 한계점이 존재한다.

요구사항 선정 평가 시 정량적인 정보를 활용하는 연구로써 Requirements Triage를 들 수 있다[13]. 이는 요구사항 선정 평가 시 고려해야하는 다양한 기준과 방법을 소개하였다. 특히, 다양한 기준을 고려하여 요구사항 선정 평가 시 측정되어야하는 값들이 미래에 대한 예측 값 또는 불확실성을 가진 값이라는 가정 하에 확률 이론을 도입하여 정량적인 정보를 획득하였다. 이러한 정량적인 정보를 이용하여 의사 결정하는 방법은 본 연구의 측정값의 해석과 의사결정 영역에 중요한 영향을 미쳤다.

III. 요구사항의 표준화와 정형화 전략

웹과 모바일 시스템에서 표현되는 요구사항은 사용자의 수준이나 시스템에 접근하는 관점의 차이로 인하여 요구사항 간의 의미가 모호하거나 중복되고 유사하여 표현 방식을 표준화하고 요구사항 계층 간의 수준을 정형화하는 요구사항 정제 작업이 필요하다.

웹과 모바일 시스템에서 자동으로 추출된 요구사항 자료들은 [Figure 1]와 같은 '휴리스틱에 기반 한 요구사항 정제 절차'를 통하여 정제된 요구사항으로 정리된다. 이 절차는 시스템을 이해하고 있는 시스템 분석가의 경험과 직관을 바탕으로 업무 수행할 것을 전제로 하고 있다.

요구사항 정제와 PIECES를 이용한 분류는 크게 5개의 타스크로 나누어진다. 각 타스크는 이후 절에서 자세하게 설명된다.

1. 사용자 감정 표현 제거

자동으로 추출된 요구사항 자료 중에서 요구사항으로 인식할 수 없는 사용자 목소리를 제거하는 단계이다. 실험적 연구에서 70% 정도에 해당하는 사용자 목소리가 감정을 표현하거나 너무 추상적인 부적격 자료이므로 이를 자동화할 수 있을 때까지 분석가의 휴리스틱한 제거에 의존한다.

예를 들어 '앱이 너무 예쁘지 않습니다. 예쁘게 디자인해주세요요'와 같은 감정적인 요구사항, '제가 원하는 기능을 서비스해주세요요'와 같은 너무 추상적인 요구사항은 요구사항 목록에서 제거의 대상이 된다.

2. 소프트웨어 구조에 맵핑

1차 정리된 요구사항들은 기 운영 중인 소프트웨어의 구조에 맵핑하여 분류한다. 소프트웨어 구조(기능)별로 분류된 요구사항들은 동일한 내용이 중복되어 표현되거나, 추상화 정도가 지나치거나, 상세화되어 있을 수도 있고, 새로운 기능을 요구할 수도 있다. 이때에도 2차적인 분석가의 경험이나 직관을 바탕으로 한 휴리스틱 정제가 요구된다.

예를 들어 '재학생들이 사용하는 모바일 앱에서 등록된 친한 친구의 시간표를 상대방의 공개 승인 후 공유하고 싶다.'와 같은 서비스는 소프트웨어 구조적인 측면에서 시간표와 관련된 "수강신청"에 해당한다.

3. 요구사항 표현의 표준화

다양한 사용자 의견을 요구사항으로 표현하기 위하여서는 표준 형식이 필요하다. 본 연구에서는 소프트웨어 사업 선진화 포럼에서 제시한 요구사항 명세화 표준[19] 중에서 [Figure 2]와 같이 3가지 요구사항 명세 형식만을 사용하여 표준화 한다. 경우에 따라서는 보다 많은 표준을 적용하여도 된다.

- ① 시스템은 <<목적>>을 위하여, <<유형>>의 <<이름>>을 <<작성 형식과 매체>>로 제공하여야 한다.
- ② 시스템은 <<규정>>을 준수하여야 한다.
- ③ <<사용자 유형>>는 시스템의 <<접근 방법>>을 이용하여 <<기능>>을 사용할 수 있어야 한다.

(Figure 2) Three requirements specification format

예를 들어 "모바일 앱은 학생들이 수강신청 시 학생들 간의 수강신청 정보를 공유할 수 있도록 제공한다." "시스템은 개인 정보 보호를 위해 서로 등록되고 승인된 학생들에게만 서로의 정보를 제공할 수 있다.", "모바일 앱은 학번으로 재학생 여부를 확인받은 후 사용 가능하다." 와 같은 형식으로 요구사항을 작성한다.

4. 요구사항의 통폐합 및 정제

요구사항의 표준화 과정에서 중복된 요구사항이 발생할 수도 있고, 동일한 타입에 유사한 단어가 발생할 수도 있으며, 추상화 수준의 차이에서 여러 개의 요구사항이 하나로 표현되기도 한다. 이 과정을 통하여 요구사항의 1차 표준화 과정이 완성된다. 이러한 요구사항 정제의 과정은 한 번에 완성되는 것이 아니다. 피드백을 통하여 지속적으로 요구사항 목록을

완성하여야 한다. 1차 표준화된 요구사항은 기존 소프트웨어의 수정/보완 항목과 신규 기능으로 분류된다.

예를 들어 ‘모바일 앱에서 재학생은 자신의 수강신청 정보를 관리할 수 있다.’와 ‘모바일 앱에서 학생들은 재학생 여부를 확인받아 수강신청, 수정, 삭제, 확인을 할 수 있다.’의 요구사항은 같은 의미로 볼 수 있다. 즉, 재학생과 학생들은 같은 의미이고 수강신청 정보 관리의 수강신청, 수정, 삭제, 확인과 같이 업무가 분할될 수 있으므로 두 요구사항은 통합되어 정제되어야 한다.

5. PIECES 에 의한 요구사항의 적합성 검사와 정제

표준화된 요구사항은 시스템의 문제와 새로운 기회를 얻기 위한 요구사항으로 적합하지 않을 경우 검증하여야 한다. 요구사항의 적합성 여부를 판단하기 위하여 J.Wetherbe 에 의해 제안된 경영정보시스템의 전략 계획 방법론 중 하나인 PIECES 프레임워크를 사용한다.

표준화된 요구사항을 [Figure 3] ‘PIECES 관점에서 요구사항의 분류’와 같이 점검한다. 각 요구사항은 PIECES 항목에 2개 이하로 체크될 수 있도록 추상화 수준을 조정한다. 이 때 요구사항의 PIECES 의 항목 체크는 장단점이나 이익의 개념 등이 상반된 관점으로 선택하도록 한다.

이 과정에서 요구사항으로 적합하지 못한 항목들은 제거되거나 통폐합 및 정제 단계를 다시 거쳐 요구사항 정제 작업이 완성된다.

마지막으로 요구사항의 중복 제거, 통폐합, 추상화 조정에 따라, 요구사항 식별자의 표기법이 필요하다. 식별자는 요구사항의 내용을 명확하게 알지 못할지라도 분석된 결과를 식별자만 통해서 알 수 있게 한다. 본 연구에서 제안하는 요구사항의 식별자는 “R” + “일련번호” + {“소프트웨어 구조 약어” + “PIECES 관련 카테고리 (+- 순으로)” + “버전”} 의 형태이다. 만약, 초기 요구사항으로써 소프트웨어 구조에 배정되지 않은 경우 식별자에서 소프트웨어 구조 약어는 제외될 수 있으며, 동일한 방식으로 PIECES 관련 카테고리가 정해지지 않은 경우, 초기 버전인 경우 제외가 가능하다. 이러한 제의 가능함을 { }를 이용하여 표현하였다. 만약 요구사항이 통합 및 정제된 경우 이는 식별자에서 버전으로써 표기한다. 본 연구에서 버전은 수정날짜 + a ~ z 로 표기하며, 요구사항 추적 관리 테이블의 활용을 권장한다. 요구사항 추적 관리 테이블은 본 연구의 범위에선 제외한다. 요구사항 식별자 표기법의 예시에서, [Figure 3]의 첫 번째 요구사항인 “시간표 공유”의 경우 초기에는 R01 으로 시작한다. 만약 이 요구사항이 “수강신청” 소프트웨어 구조에 할당되면 R01-수강신청

으로 변경되며, PIECES 측면에서 정보에 +, 통제에 - 가 배정된 경우, R01-수강신청+정보-통제으로써 표현된다. 만약 이 요구사항이 다른 요구사항과 통폐합되거나 추상화 수준의 조정에 따라 정제되면, R01-수강신청+정보-통제-120817a 로써 표현한다. 이러한 식별자의 형식은 조직에 따라 변경될 수 있다. 예를 들어 수강신청의 경우 01로 변경될 수도 있고, PIECES 의 경우 영문 약어로 사용될 수도 있다. 버전 역시 조직 고유의 표기법에 따라 변경될 수도 있다.

소프트웨어 구조		정제된 요구사항 목록					
		상호	정보	관계	통제	유용	서비스
수강신청 (01)	R01_01 (시간표 공유)		+		-		
	R03_01 (과목별 교수 평가 열람)		+		-		
	R04_01 (수업 과목 버튼 열람)	+		-			
수업 (03)	R05_03 (수업 정보 기본 SNS 연계)				-		*
	R06_03 (수업 과제 공지 실시간 알림)				-		*
학교생활 (05)	R13_05 (외부장학금 정보 맞춤 서비스)		+				
	R14_05 (SNS를 통한 지도교수 상담)				-		*
도서관 (07)	R21_07 (도서관 추천도서 알림)	-	+				
	R23_07 (열람실 예약)					-	*
...	...						

(Figure 3) PIECES from the point of view of the classification requirements

R01_01의 요구사항은 모바일 앱에서 재학생이 수강신청 시 재학생들끼리 시간표를 공유하고 싶다는 요구사항이다. 친한 친구들끼리 시간표를 공유하여 수강신청 시 정보를 제공할 수 있다는 장점은 있지만 친한 친구라도 원하지 않는 개인 정보를 제공함으로써 보안에는 좋지 않은 영향을 제공할 수 있다. 그래서 R01의 요구사항은 정보에 + 영향이 있지만 통제에 - 영향이 될 수 있다.

IV. 요구사항의 우선순위 결정

제 3장에서 요구사항은 정제되었으며, 소프트웨어 구조와 PIECES에 의해 분류되었다. 요구사항의 선택을 위하여 구조 및 PIECES와의 관계를 고려하여 요구사항의 우선순위를 결정해야 한다.

요구사항의 우선순위를 결정하기 위하여, 요구사항이 소프트웨어에 기여하는 정도에 대한 상대적인 또는 정량적인 분석이 요구된다. 요구사항에 대한 여러 척도를 이용하여, 요구사항이 소프트웨어에 기여하는 정도를 측정할 수도 있지만, 본 연구의 경우 우선순위를 결정하는데 초점을 두기 때문에 정량적인 분석이 아닌 상대적인 분석을 수행한다.

요구사항의 기여 정도를 분석하는 것은 매우 시간 소모적인 일이다. 그러므로 요구사항에 대한 기여 정도 분석 이전에 요구사항에 대한 선별 작업을 수행해야 한다. 이러한 선별 작업

은 PIECES의 주요 개념에 따라 요구사항이 특정한 문제를 해결하고 이를 통해 기회를 획득하는 관점에서 살펴보아야 한다.

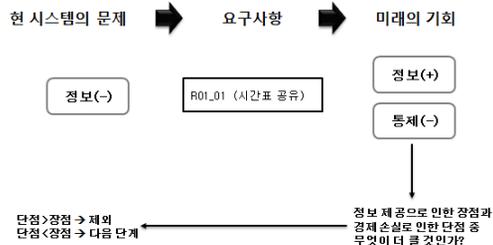
[Figure 4]는 요구사항의 우선순위를 결정하기 위하여 본 연구에서 제안하는 2가지의 주요 타스크를 설명한다.



(Figure 4) Requirements prioritization

1. 문제와 기회를 고려한 요구사항의 타당성 분석

본 연구에서 활용하는 PIECES의 주요 본질은 시스템의 문제와 기회를 식별하는 것이다. 요구사항은 식별된 문제를 해결하고 이를 통해 기회를 얻기 위한 솔루션으로써 볼 수 있다. 그러나 사용자의 목소리를 통해 획득된 요구사항은 이러한 문제와 관련이 없거나 또는 문제를 해결하기 위해 너무 많은 노력이 필요하여 효율성이 저해되는 경우가 존재한다. 요구사항의 우선순위 결정에서 수행해야 하는 첫 번째 타스크는 바로 이러한 요구사항을 식별하여 제거하는 것이다. 문제와 기회를 고려한 요구사항 타당성 분석에 대한 개념은 [Figure 5]와 같이 표현할 수 있다.



(Figure 5) Requirements for the validity of the analysis considering the problems and opportunities

예를 들어, [Figure 5]에서 현 시스템의 문제로서 사용자는 모바일 앱을 통해 현재 상영작에 대한 정보를 얻길 바라지만 그렇지 못하고 있다고 가정하자. 이러한 경우, R01_01 "시간표 공유" 이란 요구사항은 사용자에게 어떠한 정보를 제공함으로써 기회를 얻을 수 있을 것이다. 하지만 이러한 요구사항을 구현하기 위한 노력과 비용이 요구된다. 그렇다면, 시스템 분석가는 이 요구사항을 구현함으로써 인해 얻을 수 있는 정보 제공의 장점과 경제 손실의 단점을 비교하여 단점이 큰 경우 요구사항을 제외해야 할 것이며, 장점이 클 경우 요구사항 우선순위 결정 타스크로 요구사항을 전달해야 한다.

이러한 분석은 초기에는 휴리스틱한 분석에 의해 수행될 것이다. 하지만 요구사항 우선순위 결정 과정을 통해 정량화된 수치를 이용하여 제거하여도 된다. 단, 중요한 것은 요구사항의 개수가 너무 많은 경우 요구사항에 대한 기여 정도를 수치화하는데 많은 노력이 들기 때문에, 가능한 적당한 크기로 (즉, 주어진 예산과 시간을 고려하여 수용 가능한 요구사항) 정리해야 할 필요성이 있다.

2. 요구사항의 우선순위 화와 상대적 기여 정도

소프트웨어 구조	정제된 요구사항 목록	성능 (0.3)	경량 (0.7)	경제 (1.0)	동적 (0.5)	효율 (0.3)	서비스 (0.8)
수강신청 (0.7)	R01_01 (시간표 공유)		+1.0		-0.7		
	R03_01 (과목별 교수 평가 열람)		+0.8		-1.0		
	R04_01 (수업 과목 빠른 열람)	+1.0		-1.0			
수업 (1.0)	R06_03 (수업 정보 기존 SNS 연계)				-1.0		+0.6
	R06_03 (수업 과제 공지 실시간 알림)			-1.0			+1.0
학교생활 (0.9)	R13_05 (외부강학금 정보 맞춤 서비스)	+1.0			-1.0		
	R14_05 (SNS를 통한 지도교수 상담)			-1.0			+1.0
도서관 (0.6)	R21_07 (도서관 추천도서 알림)	+1.0	+1.0				
	R23_07 (열람실 예약)					-1.0	+1.0

시스템 분석가에 의해 휴리스틱하게 결정

(Figure 7) Requirements heuristic decision

요구사항의 상대적 기여 정도를 계산하는 것은 요구사항의 우선순위를 결정하기 위해 매우 중요한 작업이다. 요구사항의 상대적 기여 정도를 계산하기 위해 고려해야 할 기준은 무엇인가?

소프트웨어 구조	정제된 요구사항 목록	PIECS
수강신청	R01_01 (시간표 공유)	문제(-) 정보(+)
	R03_01 (과목별 교수 평가 열람)	문제(-) 정보(+)
	R04_01 (수업 과목 빠른 열람)	경제(-) 성능(+)
수업	R06_03 (수업 정보 기존 SNS 연계)	문제(-) 서비스(+)
	R06_03 (수업 과제 공지 실시간 알림)	경제(-) 서비스(+)
학교생활	R13_05 (외부강학금 정보 맞춤 서비스)	효율(-) 정보(+)
	R14_05 (SNS를 통한 지도교수 상담)	문제(-) 정보(+)
도서관	R21_07 (도서관 추천도서 알림)	성능(-) 정보(+)
	R23_07 (열람실 예약)	효율(-) 서비스(+)

무엇이 가장 우선순위가 높은가?

- 정량화 기준
- 소프트웨어 구조의 중요성
 - PIECS 항목의 중요성
 - PIECS 축면의 요구사항 기여 정도

(Figure 6) First quantify the ranking for the required

앞선 타스크들을 수행하여 [Figure 6]과 요구사항의 상대적 기여 정도를 계산하기 위해 고려해야 할 기준으로 소프트웨어 구조를 고려한 요구사항 그룹화, PIECES 프레임워크에서 영향을 미치는 카테고리 확인, 확인된 카테고리의 점수화

등으로 요구사항에 따라 상대적으로 기여도를 확인하여 계산하였다. 여기서 R01_01 과 R03_01 의 경우 동일한 PIECES 카테고리에 해당하지만 기여 정도가 다르다. Figure과 같이 R03_01의 경우 R01_01 에 비해 정보 측면에서 기여하는 바는 크지만, 통계적인 측면의 단점이 더 많은 편이다. 이러한 경우 무엇을 선택해야 하는가? 또 다른 예시로 R21_07 과 R01_01의 경우, 서로 다른 PIECES 의 카테고리에 해당한다. 이러한 경우 R21_07 과 R01_01 중 무엇이 더 중요한 것인가? 마지막 예시로, R01_01과 R14_05 의 경우 동일한 PIECES 카테고리에 해당하며, 비슷한 기여 정도를 가지지만, 각 요구사항이 속한 소프트웨어 구조에는 차이가 존재한다. 과연 R01_01과 R14_05 중 무엇이 우선순위가 더 높을 것인가? 이러한 물음에 답하기 위하여 본 연구에서는 우선순위를 위한 3가지 기준으로써, “소프트웨어 구조별 중요성”, “PIECES 카테고리별 중요성”, “PIECES 측면에서 요구사항 기여 정도”를 제시한다.

각 기준으로 상대적인 중요성을 계산해야 한다. 상대적 중요도는 가장 중요한 것을 1.0 으로 가장 중요하지 않은 것을 0.0 으로 하며, 나머지는 0.0 ~ 1.0 사이의 분석가의 휴리스틱한 분석에 따른 수치를 가지게 된다. 예를 들어, S 대학교 모바일 앱의 경우 소프트웨어 구조에 “수강신청”, “수업”, “학교생활”, “도서관”의 소프트웨어 구조가 존재하며, 시스템 분석가는 주어진 상황을 고려하여 수업의 중요도를 가장 중요하며 (중요도 1.0), 나머지 3개 구조에 상대적 중요도를 결정하였다 (0.9, 0.7, 0.6). 이러한 방식은 PIECES 카테고리별 중요성, PIECES 측면에서 요구사항 기여 정도에도 동일

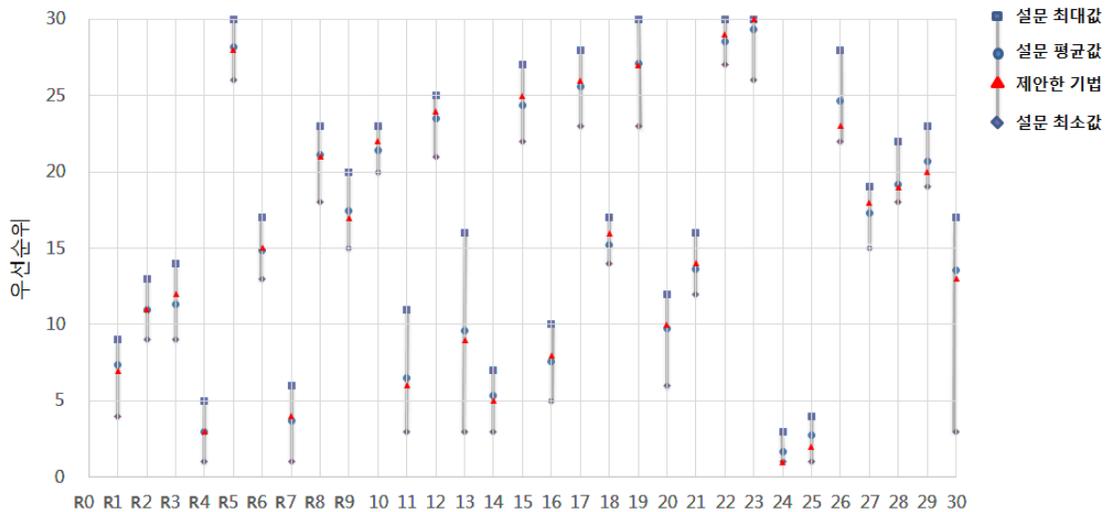
하게 적용된다.

PIECES 측면에서 요구사항 기여 정도를 결정할 때에는 추가적인 분석 조건이 존재한다. 상대적 기여 정도의 중요한 가정 중 하나는 동일한 그룹이 존재한다는 의미이다. 즉, 동일한 그룹 내에서 상대적 기여 정도를 계산하는 것이다. 예를 들어, 요구사항 R01_01의 정보 측면에서의 상대적 기여 정도를 계산하기 위한 후보는 동일한 소프트웨어 그룹 내에 있는 정보 측면을 다루는 요구사항들 (R01_01, R03_01)을 의미한다. R13_05은 동일한 소프트웨어 구조 내에 존재하지 않으며, R04_01 은 정보 측면을 다루고 있지 않다.

그리고 앞서 요구사항은 PIECES 측면에서 +, - 로 구분되었다. 상대적 기여 정도는 동일하게 계산되며, 요구사항의 단점 중 가장 큰 영향을 미치는 것을 -1.0, 그렇지 않은 것을 -1.0 ~ -0.0 사이에 값을 가질 것이다.

요구사항의 상대적 기여 정도를 계산을 위하여, 3가지 기준에 대한 상대적 기준이 모두 결정되면, 시스템 분석가는 요구사항의 상대적 기여 정도를 계산한다. 계산식은 $\sum(\text{소프트웨어 구조의 중요도} * \text{PIECES 카테고리의 중요도} * \text{PIECES 측면에서 요구사항 기여 정도})$ 이다. 예를 들어, R01_01 의 상대적 기여 정도는 $(0.7 * 0.7 * 1.0) + (0.7 * 0.5 * -0.5) = 0.31$ 이다.

모든 요구사항의 상대적 기여 정도가 계산이 되면 다음과 같은 해석으로 우선순위 화 할 수 있다. 상대적 기여 정도는 -1.0~1.0 사이의 값을 가진다. 1.0 에 가까울수록 요구사항이 시스템에 미치는 상대적 기여 정도는 문제를 해결하여 얻는 이득이 이를 위한 손실보다 크다. 반면에 -1.0 에 가까울



(Figure 8) Requirements for distribution analysis graphs

수록 문제를 해결하여 얻는 이득보다 손실이 큼을 의미한다.

모든 요구사항의 상대적 기여 정도가 계산이 되면 다음과 같은 해석으로 우선순위 화 할 수 있다. 상대적 기여 정도는 -1.0~1.0 사이의 값을 가진다. 1.0 에 가까울수록 요구사항이 시스템에 미치는 상대적 기여 정도는 문제를 해결하여 얻는 이득이 이를 위한 손실보다 크다. 반면에 -1.0 에 가까울수록 문제를 해결하여 얻는 이득보다 손실이 큼을 의미한다.

V. 사례 연구

(Table 1) 30 requirements of the "S University Bachelor of system" for priority results

		이해당사자				제한한 기법
		S1	...	S15	평균	
요구 사항	R1	7	...	9	7.3	7
	R2	11	...	13	10.9	11

	R30	13	...	3	13.5	13

본 연구의 타당성을 보이기 위하여 사례 연구를 수행한다. 사례 연구는 "S 대학교 학사 시스템" 을 대상으로 하였다. 시스템에 대한 30개의 요구사항을 추출하고, 시스템과 관련된 15 명의 이해당사자를 선별하여 요구사항에 대한 우선순위를 결정하도록 요청하였다. 그리고 시스템 분석가는 본 연구에서 제안한 방법에 따라 우선순위를 결정하도록 요청하였다. 우선 순위 결정과정은 모두 동영상 촬영을 하여 우선순위 결정 방법을 기록하였다. 우선순위 분석이 충분히 고려되도록 15 명의 이해당사자에게 자신들이 사용하는 방법을 토의하여 유사한 방법을 사용하는 이해당사자들끼리 분석 과정이 올바르게 수행되었음을 상호 검증하는 방식으로 분석의 신뢰성을 높였다.

본 연구의 타당성 분석은 15 명의 이해당사자가 결정한 각 요구사항의 우선순위 결과와 제안한 방법을 통한 우선순위 결과에 대한 통계 분석으로 이루어진다. 이해당사자가 저마다 결정한 우선순위와 제안한 방법을 통해 결정된 우선순위의 상관 분석을 통해 본 연구의 효과성을 검증할 것이다. 또한 15명이 우선순위를 부여하기 위해 투입된 시간과 본 연구를 통해 우선순위가 부여된 시간을 비교하여 본 연구의 효율성을 검증할 것이다.

1. 우선순위 결과 비교를 통한 제안한 기법의 효과성 검증

15명의 이해당사자들은 4명 사용자, 4명 운영자, 3명 관리자, 4명 개발자로 구성되었다. 각기 30개의 요구사항에 대하여 우선순위를 결정하였다. 우선순위 결정 방법은 개인이 알고 있는 방법을 사용하였다. 방법에 대한 비교는 일부 이해당사자의 경우, 요구사항의 우선순위를 위해 필요한 지식을 갖추지 않아서 소프트웨어 구조 중 판단 가능한 분야에 대해 우선적으로 우선순위를 결정하게 하였으며, 이후 이러한 이해당사자들끼리 협의를 거쳐 남은 요구사항에 대한 우선순위를 결정하게 하였다. 마지막으로 시스템 분석가는 동일한 요구사항에 대한 본 연구에서 제안하는 방법으로 우선순위를 결정하였다. 그 결과는 (Table 1)와 같다.

이해당사자 S1 ~ S15 는 각 요구사항에 우선순위를 결정하게 하였다. 우선순위는 1 ~ 30 으로 결정되었으며, 1이 우선적인 요구사항, 30은 그렇지 않은 요구사항을 의미한다. 또한 동등한 우선순위를 가지지 않도록 요청하였다. 모든 이해당사자의 우선순위 평균도 계산하였으며, 제안한 기법을 적용하여 결정된 우선순위 역시 (Table 1) 의 마지막 컬럼에 표현되었다.

설문 결과에 대한 (Figure 8)와 같이 분포 분석을 수행하였다. 각 요구사항에 대해 15명의 이해당사자의 우선순위를 최소, 최대, 평균값으로 표현하였으며, 제안한 기법으로 결정된 우선순위도 표현하였다. 분포에 영향을 미치는 우선순위의 표준편차는 대부분 0.9 ~ 1.5 의 범위를 가졌으며, R30 의 경우 3.5 정도의 분산이 계산되었다. 이는 대부분의 우선순위의 결과가 95% (2 시그마) 의 범위에서 -3 ~ +3 이내에 존재함을 의미한다. 이는 (Figure 8)에서 확인할 수 있는 것처럼 평균을 중심으로 오차의 범위가 3 이내에 존재함을 의미한다. 또한 분포 분석 결과 본 연구에서 제안한 기법이 평균과 매우 유사한 값을 가짐을 알 수 있었으며, 최소/최대 범위 내에 존재함을 알 수 있다.

더욱 자세한 분석을 위해 통계분석을 수행한다. 15명의 이해당사자의 우선순위 결과에 대한 일관성 분석 결과는 Cronbach's Alpha 분석으로 수행하였다. 이는 15명의 이해당사자의 결과가 일관성을 가지고 올바르게 분석을 수행하였음을 판단하기 위함이다. 일관성 분석 결과 0.998 의 Alpha 값이 나오으로써 15명의 우선순위가 일관성을 가지고 있음을 알 수 있다. 이는 15명의 이해당사자가 30개의 요구사항에 대해 유사하게 인식하였음을 의미한다. 즉, R01 에 대해 누군가 높은 우선순위를 결정하였다면, 다른 사람들도 비슷한 수준의 우선순위를 가졌음을 의미한다.

(Table 2) First ranking results for reliability analysis
신뢰도 통계량

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	항목수
.998	.998	15

15 명의 이해당사자의 우선순위 결과와 제안한 기법의 우선순위와의 일치 정도를 평가하기 위해 상관분석을 수행하였다. 상관분석 결과 99% 이상의 유의수준에서 매우 높은 상관관계 (상관계수 0.7 이상) 를 가짐을 알 수 있다. 이는 이해당사자가 결정한 우선순위와 본 연구에서 제안한 우선순위가 매우 높은 일치 정도를 가짐을 의미한다.

(Table 4) Comparison of methods for prioritizing

구분	가능한 인원 수	평균 비교 횟수	평균 비교 시간
직관적 비교를 통한 우선순위화	15/15	387	33 분
구조적 비교를 통한 우선순위화	15/15	112	49 분
수학적 비교를 통한 우선순위화	4/15	30	66 분
제안한 전략을 통한 우선순위화	15/15	58	12 분

2. 우선순위 결정 시간의 비교를 통한 제안한 기법의 효율성 검증

본 연구에서 제안한 기법에 따라 우선순위를 결정하였을 때 필요한 노력량과 15명이 저마다 우선순위를 결정하기 위해 필요한 시간을 비교함으로써 본 연구의 효율성을 검증하고자 한다.

효율성 검증을 위해 15명의 이해당사자가 우선순위를 결정하기 위한 방법에 대해 질문하였다. 방법은 크게 직관적, 구조적, 수학적 비교로 구분 가능하였다. 직관적 방법은 30개의 요구사항 중 2개의 요구사항을 비교하여 우선순위를 정하고 모든 요구사항의 우선순위가 정해질 때까지 비교를 하는 방법이다. 전수 비교로 볼 수 있으며 이해당사자의 직관에 따라 비교의 횟수가 줄어들었다. 그러나 다른 방법에 비교하여 직관에 의한 빠른 비교가 가능하였다. 15명의 이해당사자에게 이 방법을 설명한 결과 모든 사람이 저마다 기준이 다를 뿐 적용 가능한 방법이었다. 두 번째 방법은 구조적 비교이다. 30개의 요구사항에 대해 상위 하위 요구사항을 구분하고 동등 요구사항끼리 비교하는 방법이다. 본 연구에서 제안하는

방법 역시 구조적 비교의 일종으로 볼 수 있다. 이 방법의 경우 비교의 횟수가 크게 줄어드는 장점이 있으나, 이해당사자에 따라 요구사항의 구조가 크게 달라지며, 구조를 만들기 위해 많은 시간이 필요하다는 단점이 존재하였다. 세 번째 방법은 수학적 비교이다. 요구사항의 척도를 이용하여 우선순위를 하는 방법이다. 다른 요구사항과 비교가 아닌 요구사항별로 특정한 점수가 나오는 장점이 있으나, 척도에 따라 요구사항에 필요한 요소를 찾기 위해 많은 시간이 필요하며, 일부 요소의 경우에 접근이 일부 이해당사자에게는 불가능하였다. 또한 수학적 인 척도를 이해할 수 있는 사람만이 가능한 단점이 존재한다.

본 연구의 경우 구조적인 방법을 이용하는 기법으로써 대부분의 이해당사자들이 제안하는 기법을 이해하고 적용할 수 있었다. 평균 비교 횟수의 경우 구조적 방법에 비해 PIECES에 해당하는 요구사항만 비교하기에 좀 더 적은 비교 횟수로 우선순위가 가능하였다. 마지막으로 평균 비교 시간의 경우 매우 빠른 시간 내에 요구사항의 우선순위를 결정할 수 있었다. 상대적으로 적은 비교 횟수에 이해당사자의 직관을 이용한 빠른 비교는 본 연구의 장점이라고 할 수 있다.

VI. 결론

본 연구는 사용중인 모바일 애플리케이션에 대한 변경 요구사항을 정제하고 정제된 요구사항의 우선순위를 결정을 위한 전략을 제시하였다. 특히 휴리스틱한 분석을 기반으로 PIECES 프레임워크의 가중치를 이용하여 요구사항 우선순위화는 기존의 직관적/구조적/수학적 우선순위화 하는 방법과 유사한 결과를 얻었지만 투입되는 시간, 비교횟수 등은 본 연구가 우수함을 확인할 수 있었다.

그러나 본 연구는 소프트웨어의 구조가 이미 결정되어 있으며, 이해당사자는 의사결정을 위한 충분한 지식과 경험을 가지고 있다고 가정하고 있다. 만약 새로운 소프트웨어에서, 소프트웨어 구조가 확정되지 않았으며, 이해당사자 역시 충분한 지식과 경험을 가지고 있지 않다면, 본 연구에서 제안하는 전략을 활용하기 힘들 것이다.

향후 연구에는 휴리스틱한 분석을 반영할 수 있도록 분석틀을 제안해야 한다. 이러한 틀은 이해당사자의 지식과 경험이 반영된 것으로써, 이 틀을 적용한다면 누구나 유사한 정제 결과 및 우선순위 결과를 얻을 수 있을 것이다. 이는 급변하는 환경에서 사용자의 요구를 자동으로 인식하여 수용하는 기반을 제공할 것이다.

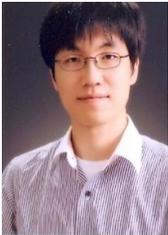
참고문헌

- [1] Ivy F. Hooks & Kristin A. Farry, "Customer-Centered Products - Creating Successful Products through Smart Requirements Management," Amacom, 2001.
- [2] James C.Wetherbe & Nicholas P.Vitalari "Systems Anlysis and Design : best practices" West Publishing Company, 1994.
- [3] Pressman, R. S., "Software Engineering - A Practitioner's Approach," McGraw-Hill, 2010.
- [4] Boehm, B., & Sullivan, K., "Software economics: status and prospects," Information and Software Technology, Vol.41, No.14, pp.937-946, 1999.
- [5] Karl W., "Software Requirements," Microsoft Press, 2003.
- [6] Darimont, R., & Van Lamsweerde, A., "Formal refinement patterns for goal-driven requirements elaboration," ACM SIGSOFT Software Engineering Notes, Vol.21, No.6, pp.179-190. ACM, 1996.
- [7] A.M. Zaremski and J. Wing, "Signature Matching: A Tool for Using Software Libraries", ACM Trans. on Software Engineering and Methodology Vol.4, No.2, pp.146-170, 1995.
- [8] IEEE, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications," IEEE Computer Society, 1998.
- [9] Dardenne, A., van Lamsweerde, A., Fickas, S., "Goal-Directed Requirements Acquisition", Science of Computer Programming, Vol.20, No.1-2, pp.3-50, 1993.
- [10] Van Lamsweerde, A., Goal-oriented requirements engineering: a guided tour. Proceedings Fifth IEEE International Symposium on Requirements Engineering, pp.249-262, 2001.
- [11] B. Potter, J. Sinclair and D. Till, "An Introduction to Formal Specification and Z," Prentice Hall, 1991.
- [12] Darimont, R., & Van Lamsweerde, A., "Formal refinement patterns for goal-driven requirements elaboration," ACM SIGSOFT Software Engineering Notes, Vol.21, No.6, pp.179-190, 1996.
- [13] A. Davis, "Just Enough Requirements Management," Dorset House Publishing, 2005.
- [14] J.C.Wetherbe, N.P.Vitalari, "Systems Analysis and Design," West Publishing Company, 1994.
- [15] Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., et al., "Identifying and measuring quality in a software requirements specification," Proceedings First International Software Metrics Symposium, pp.141-152, 1993.
- [16] Covey, S.R., "The 7 habits of highly effective people," Rockefeller Center, 1989.
- [17] Karlsson, J., and Ryan, K., "A cost-value approach for prioritizing requirements." IEEE Software, Vol.14, No.5, pp.67-74, 1987.
- [18] Liu, X. F., "Software quality function deployment. Ieee Potentials," Kluwer Academic Publishers, 2000.
- [19] B.G. Lee, M.S. Hwang, Y.B. L, H.H. Lee, J.M. Baik, C.K. Lee, "Design and Development of a Standard Guidance for Software Requirement Specification", Journal of KISS: Software and Applications, Vol.36, No.7, pp.531-538, 2009
- [20] Jong-Cheol Shin, "Improving Requirements Specification to extend Requirements Management over the Development Life Cycle", Journal of The Korea Society of Computer and Information, v.6, no.4, pp.30-37, 2001.12.
- [21] Kyu-Jang Chung, "A Study on Data Modeling Technoques for Control Requirements of SPICE Reference Model", Journal of The Korea Society of Computer and Information, v.9, no.3, pp.1-6, 2004.9.

저 자 소 개



전 혜 영
 2005 : 수원대학교 컴퓨터학과 공학사
 1991 : 숭실대학교 정보과학대학원
 소프트웨어학과 공학석사
 2004 : 숭실대학교 일반대학원
 컴퓨터학과 공학 수료
 현 재 : 부천대학교 공동연구지원센터 팀장
 관심분야 : 소프트웨어 요구공학,
 소프트웨어 아키텍처
 Email : jeonhye0@korea.com



변 정 원
 2007 : 숭실대학교 미디어학부 공학사
 현 재 : 숭실대학교 컴퓨터학과 석박사
 통합과정
 관심분야 : 소프트웨어 요구공학,
 소프트웨어 유지보수,
 소프트웨어아키텍처/프레임워크
 Email : jimi01@ssu.ac.kr



류 성 열
 1976 : 숭실대학교 전산학과 공학사
 1980 : 연세대대학교 전산학과
 공학석사
 1996 : 아주대학교 전산학과 공학박사
 현 재 : 숭실대학교 컴퓨터학부 교수
 관심분야 : 소프트웨어 요구공학,
 소프트웨어 유지보수,
 오픈소스 소프트웨어
 Email : syrhew@ssu.ac.kr