

유스케이스 점수 측정의 신뢰도 향상을 위한 단위기능 중심의 유스케이스 정제 방법

(An Elementary-Function-Based Refinement Method for Use Cases to Improve Reliability of Use Case Points)

허 령[†] 서 영 덕^{††} 백 두 권^{†††}
(Ryoung Heo) (Young-Duk Seo) (Doo-Kwon Baik)

요 약 소프트웨어의 규모측정 기법 중 하나인 유스케이스 점수(Use Case Points; UCP)는 고객 기능 요구사항에 기반하여 규모를 측정한다. 이 때 요구사항 분석자의 유스케이스 추상화 수준에 따라 유스케이스 수의 차이가 발생하고, 이는 UCP에 많은 영향을 미친다. 본 논문에서는 요구사항 분석자의 추상화 수준에 따른 유스케이스 수의 편차를 줄이기 위해 단위기능으로 분할하는 방법을 제안함으로써 측정자에 따른 UCP의 편차를 줄이고자 한다. 이 방법은 스텝, 트랜잭션, Narrative 중심으로 접근한 기존의 UCP 방법보다 측정자에 따른 UCP의 편차를 줄일 수 있어 신뢰성 있는 소프트웨어 규모측정이 가능하다.

키워드: 유스케이스 점수, 단위기능, 측정 신뢰성, 기능점수, 유스케이스 모델링, 유스케이스 크기

Abstract Use The Use Case Points method is a software estimation method that is based on user requirements. When requirement analysts elicit user requirements, they obtain different use cases because different levels of detail are possible for the Use Case, and this affects the Use Case Points. In this paper, we suggest a method to refine the level of detail of the Use Case by using the concept of an elementary function. This refinement method achieves the desired reliability for the Use Case Points because it produces less of a deviation in the Use Case Points for different requirement analysts than other methods that are based on the step, transaction, and narrative of the Use Case.

Keywords: use case points, elementary function, estimation reliability, function point, use case modeling, use case granularity

· 이 논문은 제41회 동계학술발표회에서 'Use Case Points 편차를 줄이기 위한 Use Case Granularity 정제 방법에 대한 연구'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 고려대학교 소프트웨어공학과
hvnpoet@korea.ac.kr

†† 학생회원 : 고려대학교 컴퓨터 전자통신공학과
seoyoungd@korea.ac.kr

††† 종신회원 : 고려대학교 컴퓨터전자통신공학과 교수
(Korea Univ.)
baikdk@korea.ac.kr
(Corresponding author)

논문접수 : 2015년 4월 1일
(Received 1 April 2015)
논문수정 : 2015년 7월 1일
(Revised 1 July 2015)
심사완료 : 2015년 7월 1일
(Accepted 1 July 2015)

Copyright©2015 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지 제42권 제9호(2015. 9)

1. 서론

IT서비스에 있어 범위와 시간, 비용의 판단기준으로 규모측정은 필수공정이다. 그러나 대부분의 기업이 과학적, 수학적 방법이 아닌 전문가판단이나 기존 사업경험, 사업전략에 따라 규모예측을 하고 있다. 이런 이유로 Standish Group의 2013 CHAOS Manifesto에 따르면 전체 프로젝트의 39%만이 일정과 예산을 지키고 있으며, 43%는 여전히 지연이나 예산초과, 18%는 완료 전에 취소되거나 완료되더라도 사용되지 않고 있다[1].

소프트웨어 규모측정 방식은 LoC(Line of Code), COCOMO, Function Point(FP), NESMA, COSMIC-FFP 등 오랫동안 많은 연구가 진행되었고, IT서비스의 경우 요구사항 기능 중심의 측정방법인 FP가 ISO표준이 되어 많은 국가와 기업에서 사용하고 있다. 이는 IT 중심적인 규모측정보다는 소프트웨어의 비용을 직접 지불하는 고객의 관점에서 자신들의 요구사항에 기반한 규모측정이 더 설득력이 있고, 협상하기에 용이하기 때문이다. 하지만 FP 방식도 몇 가지 단점이 있어 현실세계에 적용하기 쉽지 않은데 최근 McKinsey의 Article에서는 소프트웨어 규모측정을 하거나 추적 관리를 할 때 큰 오버헤드가 소요되고, 많은 소프트웨어 개발조직의 일하는 방식이 표준화되어 있지 않으며, 내부 직원들이 생산성 측정에 대한 거부감을 가지는 것을 원인으로 제시했다[2].

이런 상황에서 과거의 IT 중심적인 LoC나 COCOMO로 회귀하지 않으면서 기능 중심의 소프트웨어 규모측정이 가능한 방법이 바로 UCP이다. 고객의 기능 요구사항을 도출하는데 유용한 유스케이스 모델링으로부터 소프트웨어의 규모를 측정할 수 있는 UCP는 기존 FP의 단점을 극복할 수 있는 대안으로 볼 수 있다. 고객이 시스템에 요구하는 기능인 유스케이스를 중심으로 규모산정을 하는 UCP는 External Input(EI), External Output(EO), External Inquire(EQ) 등의 단위 프로세스 도출, Internal Logical File(ILF), External Interface File(EIF)과 같은 저장소에 대한 사전분석이 필요한 FP보다 시스템을 이용하려는 유스케이스와 액터만 구별해 내면 되므로 규모측정에 투입되는 노력이 50% 감소되며, 별도의 규모산정을 위한 기간이 필요한 것이 아니라 기존의 기능 요구사항 분석 공정에 포함되어 있어 기존에 일하던 방식을 변화시키지 않으면서도 바로 적용할 수 있으며, 실제 투입되는 노력의 규모가 비교적 정확히 측정된다[3].

이런 장점에도 불구하고 UCP는 FP보다 광범위하게 사용되고 있지 않다. 이는 UCP 자체가 가지는 문제점 때문이 아니라 유스케이스 모델링에 대한 오해 때문인

데 많은 사람들이 유스케이스 모델링을 객체지향 프로그래밍에만 적합하다고 알고 있거나, 기능요구사항을 도출하는 보조도구로 알고 있기 때문이다. 또한 유스케이스 모델링을 하는 사람마다 도출한 유스케이스의 수가 다르기 때문에 '그것으로 규모측정이 가능한가?'에 대한 의문을 갖는 것도 하나의 원인이라고 볼 수 있다.

기존의 UCP를 통한 소프트웨어 규모측정 연구들은 유스케이스 모델링으로부터 도출된 UCP가 소프트웨어의 규모를 얼마만큼 정확히 반영하느냐에 중점을 두고 있다. UCP는 유스케이스의 수에 크게 영향을 받는다. 유스케이스의 수는 유스케이스 추상화 수준에 따라 달라진다. 유스케이스를 어느 수준에서 결정해야 하는가는 많은 연구가 이루어져 있는데 실무에서는 요구사항을 도출하는 도구 정도로 활용하고 있으므로 그 개념을 그대로 적용하기는 어렵다. 따라서 현재의 유스케이스 모델링 기법을 적용하면 요구사항 분석자에 따라 서로 다른 유스케이스 모델링이 그려지고, 그로 인해 서로 다른 크기의 유스케이스가 도출되어 UCP의 편차를 발생시키는 요인이 될 수 있어 측정된 소프트웨어 규모에 대한 신뢰를 보장하기 어렵다.

본 논문은 UCP가 높은 신뢰도를 가질 수 있도록 유스케이스를 정제하는 방법으로 '단위기능 중심의 유스케이스'를 제안한다. 최초 유스케이스 모델링 결과를 단위기능 중심의 유스케이스로 만들어가는 과정을 통해 동일한 요구사항을 기준으로 측정된 UCP의 편차가 줄어들 수 있는지를 실험한다. 이를 통해 전문가만이 측정할 수 있는 UCP가 아닌, 유스케이스 모델링을 할 줄 아는 측정자만으로도 실무에서 활용 가능한 UCP를 얻을 수 있다는 것을 증명하고자 한다.

2. 관련 연구

G. Karner는 유스케이스와 액터를 통해 시스템의 크기를 추정하고자 하였다. 즉, 시스템을 구성하는 유스케이스와 액터의 개수에 복잡도를 고려하고, 환경요인과 기술복잡요인을 고려하여 UCP를 구하였다. 이 때 유스케이스의 복잡도는 트랜잭션의 개수에 따라 1~3개이면 Simple(5), 4~7개이면 Average(10), 8개 이상이면 Complex(15) 3가지로 구별하였고, 액터의 복잡도는 액터의 유형에 따라 Simple(1), Average(2), Complex(3) 3가지로 구별하였다[4].

이후 M. Ochodek et al.은 유스케이스의 복잡도를 구하기 위해 트랜잭션보다 그것을 구성하는 스텝을 기반으로 복잡도를 산정하였고, Kasi Periyasamy et al.은 유스케이스를 구성하는 Narrative를 중심으로 접근을 하였으나 샘플규모가 작고 결과도 크게 개선되지는 않았다[5,6].

또한 인간의 요구사항 추출 정확도가 높지 않다는 것에 착안하여 UCP에 Fuzzy이론을 적용한 연구도 진행되었다. 하지만 이를 적용하기 위해서는 요구사항 문서가 매우 정교하게 작성되어야 하는 전제조건이 필요하고, 문서가 매우 간단하거나 매우 복잡한 경우에만 비교적 정확한 규모의 측정이 가능했다[7].

그러나 이러한 UCP에 대한 기존 연구들은 유스케이스의 추상화 수준에 따라 유스케이스의 개수가 달라질 수 있다는 위험성을 내포하고 있다. 이것은 소프트웨어 규모를 측정하는데 있어 매우 중요한 부분으로 이것이 해결되지 않으면 측정된 값에 대한 신뢰를 얻을 수 없게 된다. FPA에서는 그것을 단위 프로세스(elementary process)라는 개념으로 정리하였다. 즉, FPA의 정의에 따르면 단위 프로세스는 User에게 의미를 줄 수 있는 최소한의 활동 단위를 말하는데 이것은 그것 자체로 비즈니스 기능을 완성시키는 단위이다. FPA에서는 단위 프로세스가 트랜잭션기능의 측정 단위인데 이를 통해 균질한 기능의 개수를 얻을 수 있고, 궁극적으로는 시스템의 규모를 균질하게 측정할 수 있게 되는 기초가 된다. 그러나 유스케이스는 다수의 트랜잭션으로 구성되어 있으므로 그것을 통해 단위 프로세스를 구별해 내기에는 충분하지 않다[8].

유스케이스의 추상화 수준을 권고하기 위해 Alistair Cockburn은 유스케이스를 User Goal Level로 정의할 것을 제시했다. 즉, User task를 sea level에 비유하여 Project의 목표는 Strategic Goals Level로, User가 시스템과 인터페이스하는 접점은 User Goal Level로, 그 User Goal을 수행하기 위한 하부 기능은 Subfunction Level로 구별하고, User Goal Level로 유스케이스를 도출하는 것이 올바른 유스케이스 접근법이라고 하였다[9].

하지만 현실 세계에서는 사람마다 User Goal을 해석하는 수준이 달라 유스케이스의 추상화 수준이 다르게 적용되어 유스케이스 Granularity의 편차가 크게 발생하고 있다. 이러한 문제점을 해결하기 위해서는 요구사항 분석과정에서 기능 세분화(functional decomposition)를 매우 높은 수준으로 해야 할 필요가 있으며, 요구사항 도출 시 기능의 완전성을 기하기 위해 unity criteria를

고려하여야 적당한 granularity를 가진 기능 요구사항 문서가 나올 수 있다[10-12].

3. 단위기능(elementary function) 중심의 UCP 측정

3.1 단위기능 중심의 유스케이스 정제 방법

먼저 본 논문에서 사용하는 용어에 대한 정의는 표 1과 같다.

‘user goal’은 A. Cockburn이 제안한 유스케이스의 추상화 수준을 의미하며, ‘function’은 비즈니스의 의미를 가지는 단위 기능의 집합을 의미한다. ‘elementary function’은 ‘function’의 한 종류로서 더 이상 하위 기능으로 분리될 수 없는 최소한의 비즈니스 의미를 가지는 기능을 의미한다. 이 개념은 FP의 transaction 기능인 EI, EO, EQ와 같은 elementary process보다는 큰 개념으로서 고객관점에서의 독립된 하나의 기능을 의미한다. ‘step’이란 ‘function’이나 ‘elementary function’ 내에 존재하는 조건이나 절차를 의미하며, 도출된 이후에는 테스트케이스로서 활용될 수 있는 요소이다. 마지막으로 ‘include/extended elementary function’은 다른 ‘function’에 의해서 반드시 혹은 선택적으로 참조되는 ‘elementary function’을 말한다.

UCP의 편차를 줄이기 위해서는 유스케이스의 추상화 수준을 수렴시키는 과정이 필요하다. 즉, UCP에서는 고객의 요구사항으로부터 유스케이스만을 도출하는 것이 아니라 User가 시스템을 사용하는 목적인 User Goal Level로 유스케이스를 찾아가는 과정이 반드시 있어야 함을 의미한다. 그러나 많은 사람들이 User Goal Level에 대해 각기 다른 시각을 가질 수 있으므로 보다 구체적인 방법이 필요하다. 본 논문에서는 그 방법으로 그림 1과 같은 단계를 제안한다.

- (1) 초기 유스케이스 도출 : 요구사항 분석자의 관점에서 고객의 요구사항으로부터 유스케이스를 도출한다.
- (2) 각 유스케이스의 스텝 도출 : 각 유스케이스를 구성하는 스텝을 도출한다.
- (3) 유스케이스 단위기능 도출 : 각 유스케이스를 구성하는 스텝 단위로 최소한의 비즈니스 의미를 가지는 단위기

표 1 용어
Table 1 Terms

Term	Description
user goal	the object of use case from user to the system
function	the unit function or the set of elementary functions with business perspectives
elementary function	the function with business perspectives which doesn't separate the sub-function
step	the condition or procedure which is in the function
included elementary function	the unit function which is called mandatorily by other function
extended elementary function	the unit function which is called optionally by other function

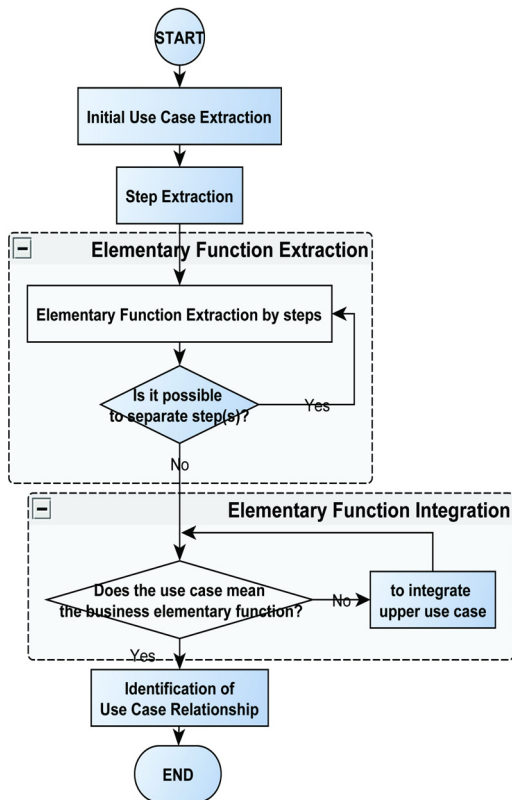


그림 1 단위기능 중심의 유스케이스 정제 방법

Fig. 1 Refinement Method for the Use Cases Based on the Elementary Function

능을 도출한다.

- (4) 유스케이스 단위기능 통합 : 앞 단계에서 도출된 최소한의 비즈니스 의미를 가지지 않은 단위기능은 상위기능으로 통합한다.
- (5) 유스케이스 관계 지정 : include와 extended 유스케이스의 관계를 지정한다.

3.2 단위기능 중심의 유스케이스 정제 단계

다음은 쇼핑몰 요구사항 기술서를 기반으로 유스케이스를 정제하는 각 단계에 대한 설명이다. 주요 기능으로는 회원가입, 로그인, 물건 구매, 배송 확인으로 제시가 되었다.

그림 2는 User Goal Level로 초기 유스케이스를 도출한 상태이다. 위에 제시한 쇼핑몰의 초기 유스케이스는 'join to be a member(사용자 가입)', 'log in the shoppingmall(로그인)', 'purchase products(상품 구매)', 'get delivery information(배송 확인)' 등이 있다. 여기까지는 추상화 수준이 요구사항 분석자마다 다를 수 있다. 이렇게 도출된 유스케이스는 고객의 비즈니스 관점



그림 2 초기 유스케이스 도출

Fig. 2 Initial Use Case Extraction

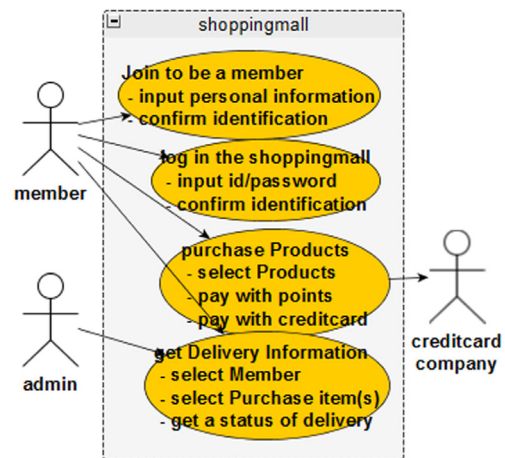


그림 3 각 유스케이스의 스텝 도출

Fig. 3 Step Extraction of Each Use Case

에서 시스템에 요구하는 기능으로 볼 수 있는데 이 기능은 그것을 구성하는 하위 기능으로 분리할 수 있다.

각 유스케이스의 하위 기능을 도출하기 위해서는 먼저 그 유스케이스를 구성하는 스텝을 도출하여야 한다. 스텝이란 유스케이스를 수행하기 위한 일련의 절차이며, 실제 구현이 되고, 테스트가 되어야 하는 대상이고, 이는 그림 3과 같다. 'join to be a member'를 구성하는 스텝으로 'input personal information(개인정보 입력)', 'confirm identification(본인 인증)'을 도출한다. 이렇게 도출된 스텝은 단위기능으로 도출할 수 있는 후보가 된다.

앞 단계에서 도출된 스텝은 각각이 하나의 단위기능으로 분리되어 또 다른 유스케이스로 분화된다. 즉, 그림 3에서의 스텝 도출과정을 반복하여 각 기능이 비즈니스 의미를 가지는 더 이상 쪼갤 수 없는 수준까지 도출한다. 이 과정을 유스케이스 단위기능 도출 단계라고 하고, 이는 그림 4와 같다.

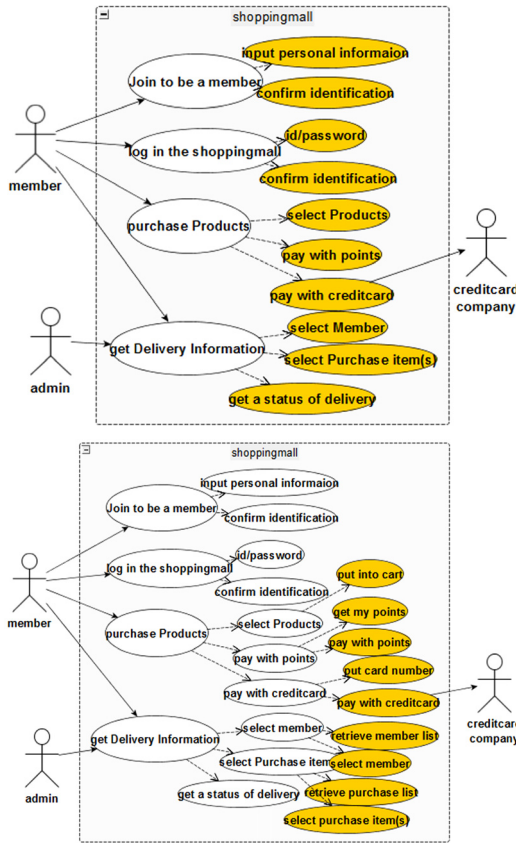


그림 4 유스케이스 단위기능 도출

Fig. 4 Extraction of the Elementary Function of the Use Case

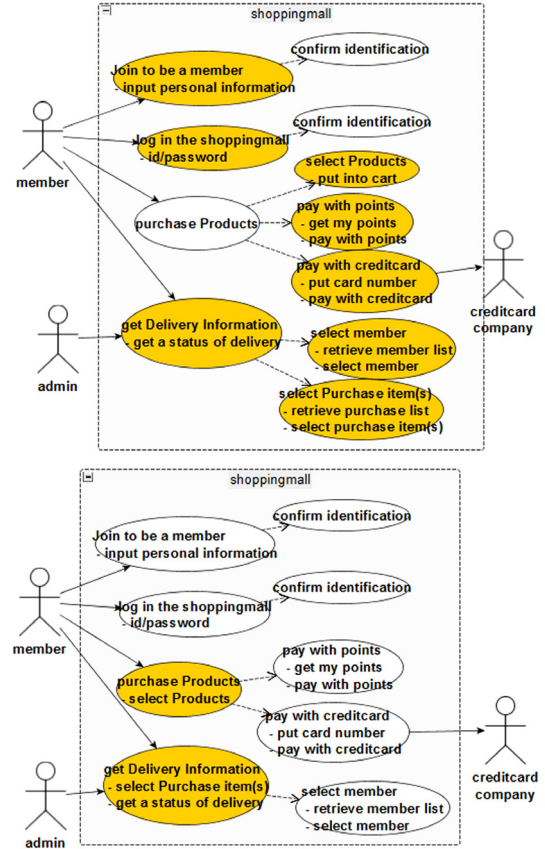


그림 5 유스케이스 단위기능 통합

Fig. 5 Integration of the Elementary Function of the Use Case

앞 단계에서 고객 관점의 비즈니스 기능으로 볼 수 없는 단위기능은 상위기능으로 통합한다. 즉, Sub Function으로 정의가 가능한 기능은 더 이상 유스케이스가 아니라고 판단하고, 앞 단계의 유스케이스를 구성하는 스텝으로 정의하여 유스케이스를 통합한다. 이 과정을 유스케이스 단위기능 통합 단계라고 하고, 이는 그림 5와 같다. 'put into cart'는 비즈니스 단위기능이라 볼 수 없으므로 'select Products'와 통합한다.

유스케이스 단위기능 분리와 통합 단계를 통해 정제된 유스케이스는 고객 관점의 단위기능으로 볼 수 있다. 고객 관점의 단위기능이란 비즈니스 최소단위로 볼 수 있는 일련의 기능을 의미한다. 이 기능은 고객이 합의할 수 있는 수준이어야 하고, 고객이 합의하지 않는다면 합의할 수 있는 수준으로 조정해야 한다. 이러한 과정에서 분리와 통합 단계는 여러 차례 반복이 될 수 있다. 또한 이 단위기능 사이에는 include와 extends 관계가 있을 수 있는데 이 관계까지 수행하면 그림 6과 같이 UCP를

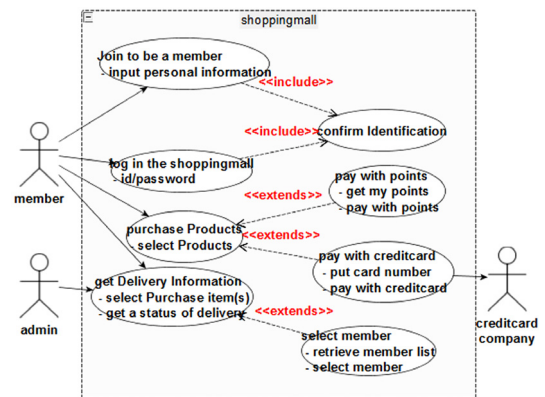


그림 6 유스케이스 관계 지정

Fig. 6 Identification of the Use Case Relationship

도출할 수 있는 유스케이스 모델이 완성된다. 이 과정을 유스케이스 관계 지정 단계라고 한다. 이렇게 유스케이스

스 모델이 완성이 되면 요구사항 분석자에 따른 유스케이스 개수의 차이가 작아지게 되고, 그것은 신뢰할 수 있는 UCP를 얻을 수 있는 기반이 된다.

이후 각 유스케이스를 구성하는 스텝의 개수에 따라 복잡도를 부여하여 조정된 유스케이스 점수(Unadjusted Use Case Points; UUCP)를 구하고, 기술복잡요인(TCF)와 환경요인(EF)를 고려하여 UCP를 구할 수 있다.

4. 측정자에 따른 UCP 편차에 대한 실험 결과 및 비교 분석

본 논문은 유스케이스 추상화 수준을 결정하는 방법을 통해 유스케이스 Granularity를 고르게 하기 위한 연구로서 UCP의 요소인 기술복잡요인과 환경요인을 고려하지 않은 UUCP에 대해서만 검토한다. UUCP는 Unadjusted Actor Weight(UAW)와 Unadjusted Use Case Weight(UUCW)의 합으로 얻을 수 있는데 본 논문에서는 유스케이스를 스텝 중심으로 접근한 M. Ochodek et al.의 방법을 준용하여 액터의 유형에 따라 1~3점을, UUCW는 유스케이스를 구성하는 스텝의 수에 따라 5~15점을 부여한다.

$$\text{Unadjusted Actor Weight(UAW)} = \sum (\text{Actor Weight})$$

$$\text{Unadjusted Use Case Weight(UUCW)} = \sum (\text{Use Case Weight})$$

$$\text{Unadjusted Use Case Points(UUCP)} = \text{UAW} + \text{UUCW}$$

$$\text{Use Case Points(UCP)} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

실험은 3.2절에서 제시한 쇼핑몰 요구사항 기술서를 기반으로 전자상거래 이용경험이 있는 학부졸업생에서부터 소프트웨어공학 분야의 6~15년 경험자로 구성된 8명의 측정자에게 의뢰하였다. 이들로부터 스텝 수에 따른 UUCP(UCP-S), 트랜잭션 수에 따른 UUCP(UCP-T), 유스케이스 Narrative를 모두 포함한 UUCP(UCP-N), 그리고 본 논문이 제안한 단위기능 중심의 UUCP(UCP-F) 등 4가지 방법의 UUCP를 측정하였다. 측정된 각각의 UUCP는 다음과 같다.

표 2의 측정결과에서는 UCP-N의 편차가 가장 작는데 이는 측정의 단위가 달라서 기인한 결과이다. 즉, UCP-N을 제외한 방법은 유스케이스의 가중치를 Simple 5점, Average 10점, Complex 15점으로 부여하나 UCP-N은 각 Narrative 요소마다 0.1~0.2점을 부여하고 있으므로 점수의 편차가 작을 수 밖에 없다. 따라서 공정한 비교를 위해서는 단순히 UUCP 편차만의 비교는 적당하지 않다.

비교단위를 일치시키는 보정을 위해 G. Karner가 제시했던 UCP-T 방식을 기준으로 시스템을 구현하기 위한 총 M/H를 구하면 다음과 같다[4].

표 2 측정된 UCP-X의 UUCP

Table 2 UUCP of each UCP-X

PERSON	UCP-S	UCP-T	UCP-N	UCP-F
A	58	43	15	58
B	73	73	21	58
C	58	48	15	48
D	78	68	19	68
E	73	58	19	63
F	58	43	15	53
G	73	58	17	53
H	68	53	16	53
AVERAGE	67.38	55.50	17.09	56.75
STDEV	8.21	11.02	2.50	6.41

표 3 UUCP별 예상 M/H

Table 3 Estimated M/H of each UUCP

PERSON	UCP-S	UCP-T	UCP-N	UCP-F
A	1,337.77	1,204.00	1,364.16	1,588.23
B	1,683.74	2,044.00	1,937.10	1,588.23
C	1,337.77	1,344.00	1,336.87	1,314.40
D	1,799.06	1,904.00	1,764.31	1,862.06
E	1,683.74	1,624.00	1,691.55	1,725.15
F	1,337.77	1,204.00	1,318.68	1,451.31
G	1,683.74	1,624.00	1,582.42	1,451.31
H	1,568.42	1,484.00	1,436.91	1,451.31
AVERAGE	1,554.00	1,554.00	1,554.00	1,554.00
STDEV	189.37	308.54	227.72	175.49

$$\text{UCP 평균값} \times 28\text{M/H} \quad (1 \text{ UCP} = 28\text{M/H})$$

예상 M/H를 기준으로 각 UCP 방법별로 1 UCP를 구하고, 이를 통해 환산한 각 UUCP의 예상 M/H는 표 3과 같다.

비교단위를 일치시키고 나니 단위기능 중심의 UCP-F가 기존의 스텝, 트랜잭션, Narrative 중심의 UUCP보다 예상 M/H의 편차가 가장 작음을 보여준다. 이것은 UCP-F의 UUCP 편차가 가장 작다는 것을 의미한다고 볼 수 있다.

이로써 유스케이스의 추상화 수준을 수렴시키기 위한 단위기능 도출단계와 통합단계가 요구사항 분석자들로 하여금 편차가 작은 UUCP를 얻을 수 있도록 작동한다고 볼 수 있으며, 서로 다른 교육이나 경험에 의존적이지 않다는 것도 같이 증명이 된다고 볼 수 있다.

5. 결론 및 향후 연구

소프트웨어 규모측정의 핵심성공요소는 정확도와 신뢰도이다. 측정자가 기준에 맞게 정확하게 측정했다고 해도 측정자에 따라 점수의 편차가 크게 되면 그것을 믿고 활용할 수 없게 된다. 또한 그 편차를 줄이기 위하

여 전문교육이나 측정시간을 늘리는 방법은 현실 세계에서 마땅한 해결책이 될 수 없다.

그 동안의 연구가 UCP가 소프트웨어의 규모를 정확히 측정할 수 있는가에 대해 접근을 하였고, 본 논문처럼 측정자에 따른 UCP의 편차가 크지 않도록 하여 UCP의 신뢰도를 높이는 방식으로 접근을 한 논문은 없었다. 이 부분은 매우 실무적인 측면에서의 접근일 수 있으나 UCP가 표준화가 되기 위해서는 반드시 검토해야 할 부분이라고 판단된다.

본 논문에서 제안한 단위기능 중심의 유스케이스 정제 방법은 요구사항 분석자에 따라 크게 차이가 나지 않는 UCP를 얻을 수 있도록 제시하고, 실험 결과로 기존의 측정방법보다 UCP의 편차가 작았음을 보여준다. 측정자에 따른 편차가 작은 만큼 측정치에 대한 신뢰도가 높아져 실제 소프트웨어 규모 측정을 하는데 널리 이용될 수 있는 기반이 될 수 있다. 그리고 유스케이스 모델링에 기반하여 고객과 합의할 수 있는 수준의 단위기능을 도출하는 과정은 별도의 교육이나 추가 투입비용이 필요한 것이 아닌 요구사항을 상세화하는 활동의 일환이므로 기존 활동의 연장선에서 쉽게 도입할 수 있는 장점이 있다. 이러한 점은 실무에서 활용가능한 소프트웨어 규모측정의 방법으로 UCP 도입을 결정하는데 도움이 될 것으로 기대한다.

다만 본 평가 실험은 하나의 요구사항 기술서만을 중심으로 측정한 것으로써 상대적으로 비슷한 규모의 점수를 얻을 수 있었다는 한계가 존재한다. 업종별, 도메인별, 규모별로 다양한 측정을 통해 UCP 편차의 비교 분석이 필요하다. 이번 연구는 UUCP에 대해서만 진행을 하였으나 실질적인 UCP를 활용하기 위해서는 기술 복잡요소(TCF), 환경요소(EF)에 대한 연구가 같이 이루어져야 할 것이다. 그리고 완전화, 적응, 수정, 예방보수 시 발생하는 변경량에 대한 UCP를 연구하여 소프트웨어 유지관리 분야에도 적용할 수 있도록 하고자 한다.

References

- [1] Standish Group international, "CHAOS Manifesto 2013 : Think Big, Act Small," *The Standish Group International Inc.*, 2013.
- [2] M. Huskins, J. Kaplan, and K. Krishnakanthan, "Enhancing the efficiency and effectiveness of application development," Article, McKinsey, 2013.
- [3] J. R. Schofield, A. W. Armentrout, and R. M. Trujillo, "Function Points Use Case Points Story Points: Observations from a Case Study," *The Journal of Defense Software Engineering*, Vol. 26, No. 3, pp. 23-27, 2012.
- [4] G. Karner, "Metrics for Objectory," Diploma Thesis, University of Linköping, Sweden, No. LiTH-IDA-

Ex-9344:21, 1993.

- [5] M. Ochodek, J. Nawrocki, and K. Kwarcia, "Simplifying effort estimation based on Use Case Points," *Information and Software Technology*, Vol. 53, No. 3, pp. 200-213, 2011.
- [6] K. Periyasamy, and A. Ghode, "Cost estimation using extended use case point (e-UCP) model," *International Conference on Computational Intelligence and Software Engineering*, 2009.
- [7] W. Fan, Y. Xiaohu, and Z. Xiaochun, "Extended use case points method for software cost estimation," *International Conference on Computational Intelligence and Software Engineering*, pp. 1-5, 2009.
- [8] Iorio T., "IFPUG Function point analysis in a UML framework," *Proc. of SMEF04*, Rome, Italy, 2004.
- [9] C. Alistair, "Writing effective use cases," Addison-Wesley, 2001.
- [10] K. Vinsen, D. Jamieson, G. Callender, "Use Case Estimation - The Devil is in the Detail," *Proc. of the requirements engineering conference, 12th IEEE International*, pp. 10-15, 2004.
- [11] A. Gonzalez, S. Espana, O. Pastor, "Unity criteria for business process modelling," *Research Challenges in Information Science*, 2009. RCIS 2009. Third International Conference on. IEEE, pp. 155-164, 2009.
- [12] S. Espana, et al., "Evaluating the completeness and granularity of functional requirements specifications: a controlled experiment," *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International. IEEE*, pp. 161-170, 2009.



허 령

1996년 건국대학교 전자계산학과(공학사)
2015년 고려대학교 소프트웨어공학파(석사). 관심분야는 요구공학, 소프트웨어 모델링, 소프트웨어 규모측정



서 영 택

2012년 고려대학교 컴퓨터통신공학부(공학사). 2012년~현재 고려대학교 컴퓨터전파통신공학파(석·박통합과정). 관심분야는 소셜 네트워크 서비스, 자가적응소프트웨어

백 두 권

정보과학회논문지
제 42 권 제 2 호 참조