

기존 ViRE 프로세스 개선을 위한 Use Case 지향 요구공학

박보경*, 문소영*, 김기두**, 김보연*, 김영철*

*홍익대학교 컴퓨터정보통신공학과

**한국정보통신기술협회

e-mail:{bk, msy, yeon, bob}@selab.hongik.ac.kr, kdkim@tta.or.kr

Use Case Oriented Requirements Engineering for improving the previous ViRE's Process

Bokyung Park*, Soyoung Moon*, Kidu Kim**, Boyeon Kim*, R. Youngchul Kim*

*Dept. of CIC, Hongik University, Sejong Campus, Korea

**Telecommunications Technology Association

요 약

전통적인 소프트웨어 개발에서는 초기 요구사항을 정확히 판단하고 분석하는 것이 중요하다. 이를 위해 B.Boehm은 가치혁신 요구공학을 제안하지만, 이는 기존의 개발 방법 기반이다. 본 논문에서는 가치혁신 요구공학 개선을 위해, 유스케이스 기반의 요구공학 방법을 제시와, 요구사항과 유스케이스의 우선순위 도출 방법을 제안한다. 이를 위해 어려운 시스템 요소(하드웨어와 소프트웨어)를 기본 단위인 유스케이스 요소와, 유스케이스 점수(Use Case Point) 개념을 적용하여 유스케이스 중요도를 도출한다. 그 기본 단위 내의 분석을 통해 쉽게 요구사항 추출 및 우선순위화를 한다. 이는 테스트 단계에서 우선순위화된 테스트 케이스를 적용이 가능하다고 본다.

1. 서론

전통적인 소프트웨어 개발 방법은 요구사항 분석, 설계, 개발, 테스트, 유지보수 등의 순서로 개발한다. 대규모 프로젝트에서 유용하게 사용하는 방법이지만, 새로운 요구사항이 발생하고, 변경해야 할 경우 다시 작업하게 되는 일이 발생한다. 성공적인 개발을 위해서는 초기 요구사항을 정확히 판단하고 분석하는 것이 중요하다[8].

정확한 요구사항을 추출하기 위해 다양한 방법이 제시되고 있다. 그 중 B. Boehm은 새로운 고객의 가치를 창조하는 가치혁신 요구공학(Value-Innovative Requirements Engineering)을 제안하였다[1,2]. 이 방법은 고객의 가치를 분석하고, 체계적인 절차와 공리적 검증을 거쳐 요구사항을 추출한다[1,2]. 그러나 가치혁신 요구공학은 현재 사용되는 개발 방법보다는 기존의 구조적 방법을 기반으로 한다. 본 논문에서는 유스케이스 기반 개발 방법으로 적용을 통해 개선하고자 한다.

가치혁신 요구공학의 매트릭스 분석에서는 시스템 요소와 고객 요구사항의 관계를 분석한다. 여기서 시스템 요소(System Elements)는 하드웨어와 소프트웨어 요소로 구분한다. 기존 연구에서는 하드웨어와 소프트웨어 요소 분할 방법에 대해서 제안하였다[4]. 하지만 이 방법은 본 논문에서와 같이 소프트웨어 요소만을 대상으로 하지 않는다. 따라서 유스케이스를 적용하면 하드웨어와 소프트웨어 요소

로 분할할 필요 없이 하나의 요소로 대체 가능하다. 또한 기존 방법인 상대 비용(Relative Cost)을 유스케이스 점수(Use Case Point) 방식을 적용하여 유스케이스의 중요도를 추출한다. 본 논문에서는 유스케이스를 기반으로 가치혁신 요구공학을 개선하여 요구사항과 유스케이스의 우선순위 추출 방법을 제시한다.

2. Related Work - Use Case Point

G. Karner가 제안한 유스케이스 점수는 제품의 유스케이스를 양적으로 평가하여 소프트웨어의 크기를 평가한다[5]. 소프트웨어의 크기는 유스케이스 개수, 크기 및 복잡도에 의해 평가된다.

유스케이스 점수는 유스케이스와 유스케이스 명세서를 기반으로 계산한다. 계산 방식은 액터와 유스케이스 가중치를 결정한다. 이때, 액터의 형태에 따라, 유스케이스 트랜잭션 수에 따라 단순(1), 평균(2), 복잡(3)으로 분류하고 가중치를 부여한다. 유스케이스 명세서에서 트랜잭션의 수를 구할 때 일반, 대체, 예외 흐름을 각각 하나로 처리한다. 각각의 가중치를 합하여 총 조정되지 않은 유스케이스 점수(UUCP: Unadjusted Use Case Points)를 구한다.

기술적 요인은 구축하고자 하는 시스템에 얼마나 영향을 미치는가에 따라 가중치를 부여한다. 유스케이스 점수에서 제시하는 총 13개의 기술적 인자를 참고한다. 13개의

1) 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2012-(H0301-12-3004))과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구 결과임.

기술적 인자 가중치(Wi)와 상수(C1,C2)는 Albrecht에 의해 제안되었다[5]. Fi는 0에서 5까지의 규모로 평가한다. 0은 중요치 않은 것을 나타내며, 5는 필수적인 것을 의미한다. 식(1)과 같이 기술적 인자들에 가중치를 각각 계산하여 기술 복잡성 요인을 평가한다.

$$TCF = C_1 + C_2 \sum_{i=1}^{13} F_i \times W_i (C1: 0.6, C2: 0.01) \quad \text{식(1)}$$

또한 생산성에 영향을 미치는 8가지 환경적 인자들을 참고하여 가중치를 부여한다.(Environmental Factor: EF) 기술 복잡성 요인과 같은 형식으로 계산한다.

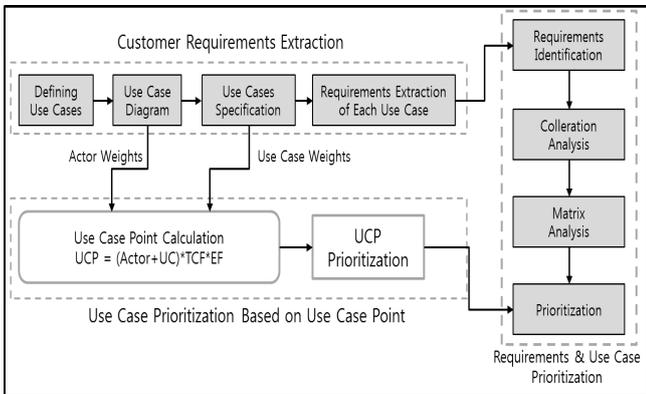
$$EF = C_1 + C_2 \sum_{i=1}^8 F_i \times W_i (C1: 1.4, C2: -0.03) \quad \text{식(2)}$$

총 조정되지 않은 유스케이스 점수(UUCP), 기술 복잡성 요인(TCF) 및 환경 요인(EF)을 곱해서 유스케이스 점수를 구한다.

$$UCP = UUCP * TCF * EF \quad \text{식(3)}$$

3. ViRE를 개선한 유스케이스 기반 요구공학

가치혁신 요구공학의 목적은 새로운 고객의 요구를 창출하는 것이다. 하지만 본 논문은 유스케이스를 기반으로 가치혁신 요구공학을 개선하는데 그 목적이 있다. 따라서 이러한 목적에 맞는 개선된 프로세스를 제시하고, 자동차 물품관리 시스템에 적용한다. 그림 1은 유스케이스 지향 요구공학 프로세스이다. 이 방법은 총 5단계로 구성된다.



(그림 1) Use Case 지향 요구공학 프로세스

Step1: Customer Requirements Extraction

이 단계에서는 유스케이스 방법론을 기반으로 고객 관점의 요구사항을 추출한다. 여기서 고객이란 시스템 개발을 의뢰한 사람을 말한다. 유스케이스를 정의하고, 유스케이스 다이어그램을 작성한다. 자동차 물품관리 시스템에서는 총 22개의 유스케이스를 추출하였다. 유스케이스 명세서는 일반, 예외, 대체흐름을 포함하여 작성한다. 이 부분은 추후 유스케이스 점수(Use Case Point) 계산에 사용된다. 또한 여기서 정의된 유스케이스는 시스템 요소를 대신하여 연관관계 분석과 매트릭스 분석에 사용된다. 마지막으로 각각의 유스케이스 별로 관련된 요구사항을 매핑한다. 이러한 과정을 거쳐 고객 관점의 요구사항을 추출한다.

Step2: Requirements Identification

요구사항 식별 단계에서는 도출된 요구사항을 우선순위를 화하고, 이에 따른 가중치 값을 결정한다. 고객과의 협의를 통해 고객 요구사항에 대한 우선순위를 평가한다. 우선순위 결과에 따라 고객 중요도(CI: Customer Importance)가 결정되는데, 1~5점으로 가중치를 부여한다.

Step 3: Correlation Analysis

요구사항을 식별한 후, 각각의 요구사항과 유스케이스와의 연관관계를 분석한다[1,2,4]. 각각의 요구사항이 어떠한 유스케이스와 연관되어 있는지 분석한다. 또한 유스케이스와 요구사항 간의 연관 정도(Correlation Degree)를 측정한다. 이를 위해 가치혁신 요구공학의 방법을 적용하여 강한 관계는 9점, 보통은 3점, 약한 관계는 1점을 부여한다.

Step 4: Matrix Analysis

연관관계 분석 내용을 토대로 매트릭스 분석에 적용한다. 매트릭스의 세로축은 고객 요구사항 전체를 나타낸 것이며, 가로축은 1단계에서 추출한 유스케이스를 적용한 것이다.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18	UC19	UC20	UC21	UC22	CI/RI	
CR1	○																							1
CR2	●																							5
CR3	●																							1
CR4	●																							2
CR5	●	●																						1
CR6	●	●	○																					2
CR7	●	●	○																					2
CR8	○																							2
CR9	●	●	●	●																				3
CR10	●	●	●	●																				1
CR11	●	●	●	●																				2
CR12	●	●	●	●																				1
CR13	●	●	●	●																				3
CR14	●	●	●	●																				3
CR15	●	●	●	●																				4
CR16	●	●	●	●																				3
CR17	●	●	●	●																				3
CR18	●	●	●	●																				5
CR19	●	●	●	●																				4
CR20	●	●	●	●																				5
CR21	●	●	●	●																				4
CR22	●	●	●	●																				5
RI	73	54	164	108	36	57	129	45	60	171	186	185	49	123	45	168	339	138	328	342	144	36		
UCP	79.75	55	110	82.5	30.8	63.8	115.5	33	74.25	126.5	104.5	84	55	121	38.5	126.5	208	72	120	234	68	27.5		

(그림 2) Matrix Analysis

매트릭스 분석단계에서는 요구사항 중요도(RI: Requirements Importance), 유스케이스 점수(UCP)를 계산하여, ViRE 방법인 Hybrid 방법에 적용한다. 그림 3은 자동차 물품관리 시스템에서 추출된 요구사항과 유스케이스를 매트릭스로 나타낸 것이다. 연관관계를 분석한 데이터는 요구사항 중요도(RI) 평가에 사용된다. 이 연관정도(Ri,j)는 각각의 유스케이스와 요구사항 간의 연관관계를 말하며, ●(9점), ◎(3점), ○(1점)으로 평가한다. 요구사항 중요도를 계산하는 방법은 식(4)를 이용한다.

$$RI = (CI_1 \times R_{1,j}) + (CI_2 \times R_{2,j}) + \dots + (CI_i \times R_{i,j})$$

$$= \sum_{i,j=1}^n (CI_i \times R_{ij}) \quad \text{식(4)}$$

그림 3에서 UC1은 총 4개의 요구사항과 연관되어 있다.(CR1, CR2, CR3, CR4) 각각의 연관정도(Ri,j)와 고객 중요도(CI) 값을 곱한다. 이 값들을 더해서 UC1에

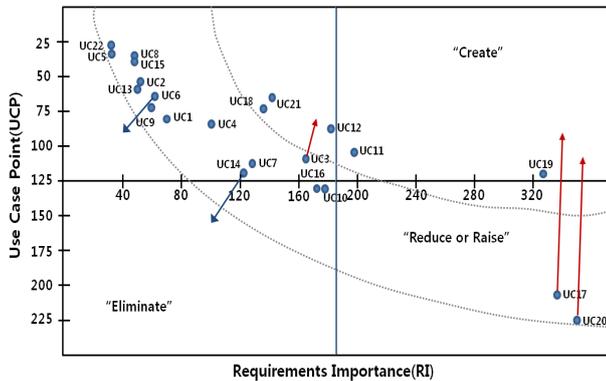
대한 요구사항 중요도(RI) 값을 구한다.

UC₁의 RI

$$= ((CI_1 \times R_{1,1}) + (CI_2 \times R_{2,1}) + (CI_3 \times R_{3,1}) + (CI_4 \times R_{4,1}))$$

$$= (1 \times 1) + (5 \times 9) + (1 \times 9) + (2 \times 9) = 1 + 45 + 9 + 18 = 73$$

요구사항 중요도를 구한 후, 각각의 유스케이스 점수를 구한다. 액터 가중치, 유스케이스 가중치, 기술 복잡성 인자, 환경적 인자를 개별적으로 평가하여 유스케이스의 점수를 구한다. 이를 비교하여 유스케이스의 우선순위를 추출한다.



(그림 3) Hybrid Approach

유스케이스 점수와 요구사항 중요도를 ViRE의 하이브리드 방법에 적용한다. ViRE에서는 투자수익 방법(ROI)을 이용하여 각각의 시스템 요소를 정렬한다. 여기서 ROI는 각각의 BI(Business Importance)값을 RC(Relative Cost) 값으로 나눈 결과이다. 정렬된 값을 통해 시스템 요소를 포함하거나 제외할 임계값을 결정하게 된다. 하지만 본 논문에서는 RC 값 추출이 불가능하기 때문에 유스케이스 점수를 적용한다. 추출된 유스케이스의 임계값을 결정하고, 점선으로 표시한다. 유스케이스가 임계값보다 크면 생성하고, 아니면 제거한다. 단, 일부의 유스케이스는 이해관계자와의 협의를 통해 늘리거나 줄일 수 있다. UC17, UC20, UC3은 협의를 통해 증가되었으나, UC6, UC14는 감소한다. 따라서 UC6과 UC14 유스케이스를 제거한다.

Step 5: Prioritization

마지막으로 전체 요구사항에 대한 최종 우선순위를 도출한다. 이를 위해서는 요구사항 중요도 값을 개별적으로 고려한다. 고객 중요도(CI)와 연관정도를 곱하여 계산한다. 이때 요구사항 중요도(RI)와 같이 관련된 모든 값을 합하는 것이 아니라 각각 계산한다. 이를 비교하여 정리하는데, 동일한 값들이 발생한다면 제일 큰 값을 제외하고 나머지 값들을 고려하지 않는다. 표1은 요구사항의 우선순위 최종 결과이다.

4. 결론

본 논문에서는 유스케이스를 기반으로 가치혁신 요구공학을 개선하여 요구사항과 유스케이스의 우선순위 도출 방법을 제안한다. 유스케이스는 시스템 요구사항 및 시스템과 액터 간의 상호작용 순서를 파악하는데 용이하다. 또한 가치혁신 요구공학의 시스템 요소는 하드웨어와 소프트웨어 요소로 분할한다. 하지만 유스케이스를 적용한다면 이를 분할할 필요 없이 하나의 요소로 대체할 수 있다. 각각의

유스케이스 점수를 계산하여 우선순위화 하고, 기존 방법을 이용하여 요구사항에 대한 우선순위를 도출한다. 또한 하이브리드 ERRC 결정 방법을 이용하여 시스템을 구현하는데 불필요한 유스케이스를 제거한다. 하지만 이 방법은 고객의 요구를 정확히 반영했는지에 대한 검증이 부족하다. 향후 연구에서는 이를 검증할 수 있는 방법을 연구할 것이다. 또한 고객 요구사항 뿐만 아니라 사용자의 요구를 반영한 요구사항 추출 및 우선순위화에 대해서도 연구할 것이다.

<표 1> Requirements Prioritization

Priority	FRs No	GI	Priority	FRs No	GI
1	CR02	45	4	CR26	27
	CR17	45		CR04	18
	CR19	45		CR06	18
	CR20	45		CR07	18
	CR28	45		CR08	18
	CR32	45		CR11	18
2	CR15	36	5	CR24	18
	CR18	36		CR03	9
	CR21	36		CR05	9
	CR25	36		CR10	9
	CR30	36		CR12	9
	CR31	36		CR14	9
3	CR09	27	6	CR23	9
	CR13	27		CR27	9
	CR16	27		CR29	9
	CR22	27		CR01	1

참고문헌

[1] Sangsoo Kim, Hoh Peter In, "ViRE: Sailing a Blue Ocean with Value-Innovative Requirements" Software, IEEE, pp.80-87, Vol.25, Jan-Feb, 2008

[2] Sangsoo Kimm, "Value-Innovative Requirements Engineering", Dissertation for the Degree of Doctor, Department of Computer Science and Engineering Graduate School, Korea University, 2007

[3] 이노우에 타츠키, "다이어그램으로 쉽게 배우는 UML", 한빛 네트워크

[4] 김능희, 박지용, 김상수, 인호, "고객 가치를 반영한 HW/SW 분할 방법론", 한국컴퓨터종합학술대회논문집, Vol.34, No.1(A), 2007

[5] Karner. G, "Resource Estimation for Objectory Projects, Objective System SFAB, 1993

[6] Bokyung Park, R. Youngchul Kim, "Extraction & Prioritization of User Preference Requirements through User Needs" The 37th KIPS Spring Conference, 2012

[7] 안성빈, 김동호, 서채연, 김영철, 정지홍, "사용자 행위 분석 기반 요구추출 방법에 대한 연구", 소프트웨어공학회, Vol.12, No.1, 2010

[8] 박보경, 문소영, 김동호, 서채연, 김영철, "Goal 지향 유스케이스 기반의 요구사항 추출에 관한 연구, 소프트웨어공학회, 2012

[9] 박보경, 문소영, 서채연, 김영철, "Goal 지향 유스케이스 기반의 요구사항 추출 및 우선순위화", 한국스마트미디어학회, Vol.1, No.1, 2012