

# 국방 소프트웨어의 가치창출을 위한 요구공학 방법론

고려대학교 김상수 · 인 호  
서강대학교 최순황 · 박수용

## 1. 요구공학의 중요성

요구사항은 소프트웨어 규모가 커지고 복잡화 됨에 따라 점차 소프트웨어 개발 성패의 중요한 요소로 자리 잡게 되었다. 소프트웨어는 개발 중 많은 오류를 수정해 나가며 개발되는데 이러한 오류는 소프트웨어 개발 생명주기의 여러 단계에서 일어날 수 있다. 오류가 요구사항 단계에서 발견된다면 큰 비용을 치르지 않고 수정될 수 있지만 코딩이나, 유지보수 단계 같은 생명주기의 후반부에서 발견된다면, 큰 비용을 치러가며 수정해야 한다. 미국 국방관련 프로젝트를 다수 수행하는 TRW 사의 경우 56%의 오류가 코딩이나 단위 테스트의 단계에서 발견되었고, 이들 중 대부분은 코딩 단계의 오류(9%) 보다는 요구사항이나 설계단계에 관련된 오류(45%)였다고 한다[1]. Standish Group의 CHAOS 보고서에 따르면 프로젝트 성공과 실패 요인 중 대부분(약 40~50%)이 요구사항과 관련된 요인이다[2]. 또한, 요구공학 분야의 권위자인 Alan Davis는 초기에 발견 가능한 많은 요구사항 에러가 프로젝트 후반부에 발견되어 많은 비용초과를 초래한다고 언급하였다[3].

국방 분야의 소프트웨어는 일반적으로 구조가 복잡하고, 규모가 크며, 대부분이 실시간 시스템이고 고도의 정확성과 높은 신뢰성 그리고 가용성이 요구된다. 따라서 사용자의 요구사항을 체계적으로 분석하고 개발에서 폐기 시까지 전 주명주기 간에 걸쳐 요구공학을 적용하여야 한다.

국방시스템의 특성과 중요성을 고려한다 하더라도 제한된 재원은 국방 획득상의 가장 중요한 문제가 되고 있다. 따라서 무기체계 및 국방정보체계의 획득 시에는 개발비용뿐만 아니라 사용자의 만족도, 즉 업무효율의 향상과 전투력의 향상에 기여한 정도가 고려되어 획득의 성공 여부가 평가되어야 할 것이다. 아무리 많은 비용을 들여 첨단 기술이 접목된 시스템을 개발하여 배치한다 하더라도 고객의 가치를 고려하지 않음

으로써 사용자의 만족도에 기여하지 못한다면 시스템의 사용률이 떨어지고 유사 업무와 타 조직 및 타 시스템으로 확대 적용이 되지 못하고 개발하여 배치한 결과로만 만족해야 할 것이다. 일반적으로 적용되고 있는 소프트웨어 공학 방법론과 요구공학 절차는 이러한 국방시스템에서 고려되어야 하는 비용적인 측면과 사용자의 만족도를 고려하지 않고 있다.

미국 USC의 CSE Berry W. Boehm은 이러한 문제를 해결하기 위하여 가치기반 소프트웨어 공학[4]에 관한 연구를 수행하고 있다. 하지만 요구공학의 구체적인 적용을 위한 프로세스는 제시하지 않았다. 우리는 고객의 요구사항으로부터 출발하여 가치혁신을 통해 새로운 가치를 창출하고 고객의 만족도를 높임으로서 새로운 시장을 개척하는 체계적이고 구체적인 프로세스인 가치혁신 요구공학 프로세스를 개발하였다. 본 고에서는 제한된 재원을 이용하여 전투력 향상과 경제적인 군 운영에 도움을 줄 수 있는 국방 소프트웨어(시스템)의 가치창출을 위한 방안을 제시하고자 한다.

이를 위해 2장에서는 가치기반 요구공학의 필요성과 주요 요소에 대하여 고찰하고, 3장에서는 가치혁신을 통해 고객을 만족시키고 비고객을 고객화 할 수 있는 가치혁신 요구공학 프로세스에 대하여 소개한다. 4장에서는 이러한 노력을 실제 국방에 적용하기 위한 방안으로 요구공학 센터의 필요성에 대하여 언급하고, 결론을 맺는다.

## 2. 가치기반 요구공학

### 2.1 가치기반 요구공학의 필요성

가치기반 요구공학은 요구공학 프로세스 동안에 경제적인 가치(economic value)를 개발하는 것이며 이것은 혁신적인 국방 소프트웨어 개발을 위한 요구공학을 수행하는 동안 매우 유용한 것이다. 국방 분야에 있어서 시스템 개발을 통하여 생성된 새로운 이윤을 찾는다는 것은 쉬운 일이 아닐 것이다. 하지만, 가치기반

소프트웨어 공학 방법론에 따라 제품을 개발 했을 때 잠재적인 이익이 생성되고 이것이 국방 분야에서는 획득비의 절감인 직접적인 이익과 업무효율의 증대와 함께 전투력의 상승이라는 잠재적인 이익을 달성할 수 있다. 이러한 가치기반 요구공학의 개념은 Berry W. Boehm이 주도적으로 연구하고 있는 가치기반 소프트웨어 공학의 근간이 된다[4].

지금까지 대부분의 소프트웨어 공학에 대한 연구와 실무적용은 가치중립(value neutral)적인 접근법을 적용하였다. 이것은 모든 요구사항, 유즈 케이스, 오브젝트, 테스트 케이스 그리고 결함 등의 중요성을 동등하게 고려하는 것이다. 과거에는 소프트웨어가 전체 시스템에서의 비용과 일정 그리고 가치가 차지하는 비중이 크지 않아 가치중립적인 접근법으로도 시스템을 개발하는데 아무런 문제가 되지 않았다. 하지만 현재 그리고 미래의 시스템, 특히 국방 시스템의 경우 소프트웨어가 차지하는 비용, 일정 그리고 가치의 비중이 적게는 30%(지상장비)에서 90%(침단유도무기) 까지 차지하고 있어 시스템 레벨에서의 소프트웨어는 의사결정의 결정적인 요소로 차지하고 있다.

또한, 가치중립적인 소프트웨어 공학 원론과 실제적인 접근법은 소프트웨어 실패의 주요원인을 해결하기 위해서 적합하지 않다. Standish Group의 CHAOS 보고서[2]에 따르면 대부분의 소프트웨어 프로젝트의 실패원인은 사용자 입력의 부족, 불완전한 요구사항, 요구사항의 변경, 재원의 부족, 현실적이지 못한 기대치, 목적의 불명확성 그리고 현실적이지 못한 일정 등과 같이 가치지향에 주력하지 않은데서 그 원인이 있다고 밝혔다.

가치중립적인 방법의 경우 개발 비용상의 책임과 관련된 의사결정이 어렵다는 것을 다음 예를 통해서도 알 수 있다:

총 사업비가 200억인 국방 자동화 정보체계를 개발하고 있다고 가정하자. 이때 공급자가 자동 시험 시스템(ATG: Automated Test Generation) 을 다음과 같이 제안하였다:

“우리의 장비는 소프트웨어 시험비용을 반으로 줄일 수 있는 장비이다. 시험 비용은 전체 개발비용의 50%가 소요된다. 즉 100억 원이 소요된다. 우리는 시험비용의 30% 가격인 30억에 장비를 공급하겠다. 당신은 장비를 사용하면 시험비용의 50%인 50억을 절약할 수 있으며, 결국은 20억의 이득을 보는 것이다.”

국방 획득관계자는 이러한 공급자의 제안을 수용해야 하는가?

이 질문에 대한 결론부터 말하면 “No”라고 해야 한다.

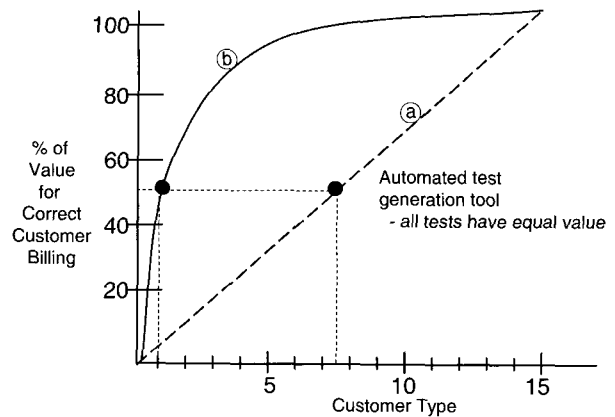


그림 1 테스트 케이스 가치의 20-80 Pareto 분포[4]

그림 1에서 보는바와 같이 가치중립 접근법의 경우 직선 a를 따르게 되므로 자동시험장비의 공급자가 제안한 내용을 만족하게 된다. 이것은 모든 요구사항, 테스트 케이스 그리고 결함들에 대해 가치중립적으로, 즉 동등하게 중요성을 부여했기 때문이다.

하지만, 대부분의 경우 그림 1의 곡선 b와 같이 Pareto 분포를 따르게 된다. 즉 80%의 시험 임무수행 효과는 20%의 소프트웨어 컴포넌트를 시험을 통해서 얻을 수 있다[5].

실제로 시스템 및 소프트웨어 개발에서는 비용 및 일정 그리고 특정 수준의 품질을 유지하기 위한 정책 때문에 이론적인 모든 시험을 수행하지 못한다. 또한 특정 시점 이상에서는 시험비용에 비해 얻어지는 이익(ROI: Return On Investment)이 극단적으로 줄어들기 때문에 가장 중요한 부분만을 골라 시험하게 되는데 이때에는 ATG 시험보다는 Pareto 시험이 효과적이라 할 수 있다.

이러한 사례는 단지 ROI driven 접근법을 적용한 사례이며 가치기반 시험방법에는 고객 만족도, 고객 유지, 조직의 평판 그리고 브랜드 등과 같은 다른 이해관계자들과 관련된 사항을 확인하는 것도 포함된다. 이러한 가치중립성은 다른 모든 소프트웨어 공학 기술에도 동일하게 적용된다. 특히, 요구공학에서는 고객의 요구를 만족시키기 위한 효율적인 노력을 위해서는 가치를 기반으로 한 체계적이고 정량적인 분석이 필요한 것이다.

## 2.2 가치기반 요구공학의 주요 요소

가치기반 요구공학은 Berry W. Boehm의 가치기반 소프트웨어 공학에서 출발한다. 가치기반 요구공학의 목적은 요구공학을 통하여 어떻게 가치를 만들어 내는지를 제시해 주기 위한 것이다. 가치기반 요구공학 프레임워크는 VBSE의 7가지 요소들로 구성되어 있다 [4].

● 이익 실현 분석(BRA: Benefit Realization Analysis)

BRA에서 가장 많이 사용되는 것은 그림 2와 같은 BRARC (Benefit Realization Approach Results Chain)이다. 이것은 정보시스템 역설(The Information Paradox)[6]을 해결하기 위하여 DMR Consulting Group에서 사용한 모델이다. 이것을 통해 우리는 일반적으로 생각하는 “국방 정보체계를 구축하면 모든 군인 또는 업무 관련자가 사용할 것이다”라는 정보시스템 신드롬(syndrome)을 해결할 수 있을 것이다.

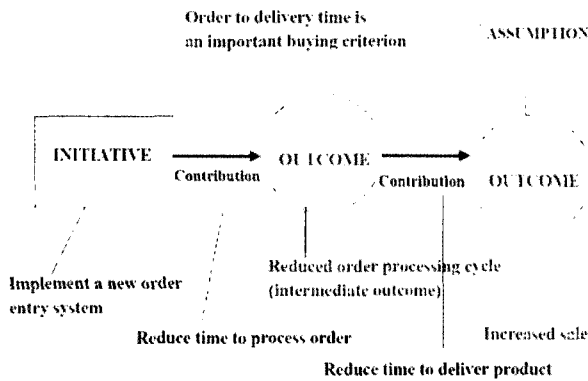


그림 2 BRARC 기본 구조[4]

● 이해관계자 가치 제안 추출 및 조정(Stakeholder Value Proposition Elicitation and Reconciliation)

시스템과 관련된 이해관계자들의 요구와 그들의 프로젝트에서의 역할이 각기 다르기 때문에 그들 간의 충돌이 발생하게 되며 반드시 해결해야만 한다. 그림 3은 이러한 충돌뿐만 아니라 시너지를 낼 수 있는 요소들을 잘 표현해주고 있다.

이해관계자들 간의 가치제안 조정을 위한 효과적인 접근법으로는 COCOMO 모델 등을 이용한 Expectation Management, Win-Win 협상모델을 적용한 그룹웨어, ROI 등 Business Case 분석 등이 있다.

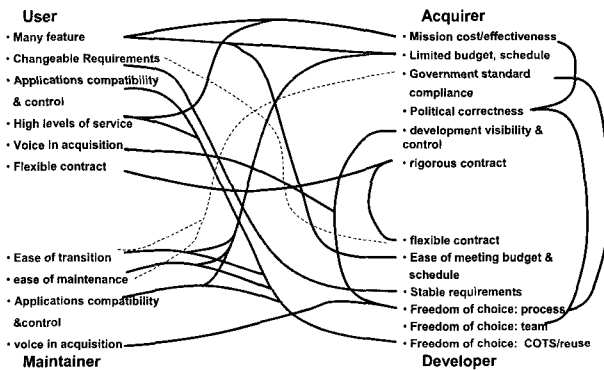


그림 3 이해당사자 간의 간의 충돌과 시너지를 표현한 Spider Web[4]

● 비즈니스 케이스 분석(Business Case Analysis)

비즈니스 케이스 분석의 가장 대표적인 예가 ROI 분석이다. 일반적인 요구공학 절차에서 타당성 분석(feasibility analysis)에 해당하는 부분이다. ROI 분석은 시스템개발에 따른 조직적 차원에서의 손익을 분석하기 위한 것이다. 국방 분야에서는 일반적으로 비용 대 효과분석(COEA: Cost Effectiveness Analysis)만을 수행하고 있다. 비용 대 효과 분석은 복수의 시스템 중 필요한 시스템을 선택하기 위해서 또는 단일 시스템의 도입에 따른 효과를 확인하여 시스템 도입을 결정하기 위해 적용한다. 비용의 경우는 산출이 쉬우나 이익의 경우 산출하기 어려운 것이 현실이다. 비용의 요소와 이익의 각 요소를 분석하고 시스템 또는 무기체계 도입시 군에 얼마나 이익이 있는지를 분석하는 작업과 다양한 옵션에 따른 의사결정을 지원하는데 ROI 분석에서 이루어질 수 있다.

● 지속적인 위험 및 기회 관리(Continuous risk and Opportunity Management)

위험이란 “발생 시에 긍정적 또는 부정적인 영향을 미치는 불확실한 사건”을 말한다[4,7]. 시스템 개발 시에는 이러한 위험들이 따르게 되며 위험을 식별하고 관리하는 일은 시스템 개발의 특정 단계뿐만 아니라 전 수명주기(life cycle) 간에 이루어져야 하며 특히 요구사항을 분석하는 단계에서는 위험을 분석하는 일을 수행해야 한다[4,8].

● 동시공학을 적용한 시스템 및 소프트웨어 개발 (Concurrent System and Software engineering)

동시공학은 소프트웨어 공학을 적용하는 노력의 가치를 최대화 할 수 있는 접근법이다. 동시공학적 접근법을 위한 다양한 프로세스 모델이 개발되어 사용되고 있다. 대표적으로 MBASE(Model Based (system) Architecting and Software Engineering), XP (Extreme Programming), RUP(Rational Unified Process) 등을 예로 들 수 있다. 동시공학적 개발 프로세스는 일반적으로 개발기간을 단축하여 사용자에게 가치를 빠르게 제공할 수 있다[4].

● 가치기반 감시 및 통제(Value Based Monitoring and Control)

가치를 기반으로 한 프로젝트의 통제 방법으로는 EVMS (Earned Value Management System) 기법과 BSC (Balanced Score Card) 방법 등이 있다. EVMS 기법은 미 국방 분야에서는 DoD 획득관리 규정에 성문화되어 있으며 우리나라의 경우에도 건설 분야에서 적용되고 있다. EVMS는 현재까지 개발된, 즉 획득된 가치를 평가하여 일정과 비용을 분석하는 접근

법이다. 프로젝트 진행시 현재의 시스템 개발의 일정과 비용을 분석하고 개발 완료시의 비용 등을 분석함으로써 시스템 개발간의 의사결정에 결정적인 역할을 한다.

● **요구사항 변경관리(Change of Opportunity)**

국방 시스템의 경우 요구사항의 변경이 어려운 경직된 절차를 따르고 있다. 잦은 요구사항의 변경은 시스템 개발비용을 증가시키고 개발기간을 지연시키는 문제점을 내포하고 있다. 따라서 요구사항의 정의단계부터 체계적인 분석 방법이 적용되고 있다.

하지만, 시스템 개발 시에 변경은 반드시 고려되고 관리되어야 한다. 또한 이러한 변경은 잘못된 요구사항을 바로 잡거나 새로운 기술의 적용을 통하여 시스템의 품질을 향상시키는 기회로 작용된다.

이상에서 소개한 가치기반 요구공학의 기본 요소들은 국방 시스템의 개발 수명주기 간에 적용되어야 할 중요한 요소들이다. 이러한 가치기반 기술이 국방 시스템 개발 시에 어떻게 적용될지에 대한 구체적인 연구가 미미하다. 따라서 본 고에서는 가치기반 기술을 국방 소프트웨어에 적용하는 프로세스를 제안하고자 한다. 특히, 국방시스템 사용자의 요구사항을 반영하여 시스템의 효과를 극대화하기 위한 방안으로 가치혁신 요구공학을 제시하였다.

**3. 국방 소프트웨어의 가치창출을 위한 요구공학 프로세스**

**3.1 국방 시스템 가치창출 사례**

제한된 재원으로 최대의 효과를 내기 위한 노력은 국방 획득관리자의 가장 큰 관심사이다. 첨단 시스템과 자동화된 무기체계로 무장하는 것이 국방에 도움이 된다는 것은 자명한 사실이다. 하지만 주어진 예산 범위 내에서 성능을 결정해야 한다. 무기체계의 성능을 높일 경우 비용이 증가하고 비용을 낮출 경우 필요한 성능의 무기체계를 획득할 수 없는 것이 일반적인 사실이다.

이러한 문제점을 해결하기 위한 방안으로 개발된 가치혁신 요구공학(ViRE: Value-Innovative Requirements Engineering) 프로세스(9)를 소개하고자 한다. ViRE 프로세스는 고객의 요구사항과 비교객의 요구를 분석하여 ERRC 분석, 즉 시스템에서 불필요한 부분을 제거하고(Eliminate), 필요이상으로 과도하게 높은 성능은 감소시키고(Reduce), 필요하지만 부족한 부분의 성능은 높이고(Raise), 필요하지만 없거나 고려되지 않은 기능을 새롭게 생성(Create)하여 새로운 고객의 가치를 창출함으로써 기존의 비 고객을 고객으로 끌어들이므로써 블루오션을 창출해 내는 접근법이다. 이것

은 국방시스템 획득 시 일반적으로 성능을 높이면 비용이 증가하고 비용을 낮추면 원하는 성능을 획득할 수 없는 비용과 성능의 절충(trade-off) 관계를 극복하여 군에서 원하는 성능을 유지하면서 비용을 줄일 수 있는 가치혁신(Value Innovation)[10]적인 접근법이라 할 수 있다.

미 국방성의 JSF(Joint Strike Fight) 획득사업은 비교객을 고객화하여 새로운 시장을 개척함으로써 성공한 사업으로 일컬어지는 대표적인 사례이다[10].

전통적으로 미국의 공군, 해군 그리고 해병대는 이상적인 전투기에 대한 기본 요구사항이 달라 독자적인 사업을 통해 전투기를 설계 개발하였다. 기존에는 이러한 각 군의 요구사항의 차이점은 극복하기 힘든 당연한 것으로 여겨왔으며 공동개발 운용이라는 개념은 생각지도 않고 있었으나, F-35 전투기 개발 사업, 즉 JSF 획득사업은 이러한 개념을 정면으로 도전한 사례라 할 수 있다.

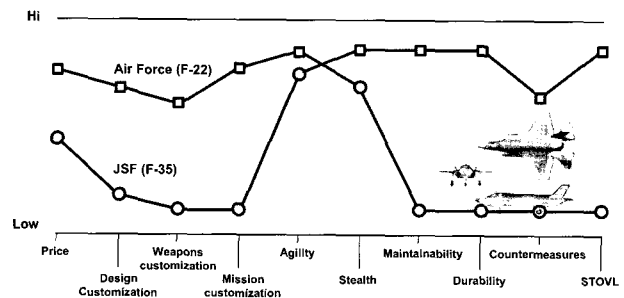


그림 4 F-22와 F-35의 전략 캔버스[10]

공군의 주요 요구사항은 전술적 민첩성과 스텔스 기능을 가진 전투기를 개발하는 것이었으며, 해군의 주요 요구사항은 항공모함에서 운용하기에 적합하도록 유지보수가 쉽고 내구성이 강한 전투기를 개발하는 것이었고 해병대는 수직 이착륙이 가능한 단거리 전력 구사용 정찰기를 원했다. JSF 사업에서는 이러한 각 군의 주요 요구사항을 수용하면서 기존의 각 군의 전투기가 보유하고 있던 추가적인 기능들을 성능을 낮추거나 과감하게 제거함으로써 그림 4와 같이 전략 캔버스를 새롭게 완성함으로써 각 군이 원하는 전투기를 성공적으로 공급할 수 있게 되었다.

JSF는 각 군의 기존 고객 개념과 달리 비교객을 고객화하고 각 군별로 분리되어 있던 수요를 집결시킴으로써 성공적인 사업의 결과를 기대하고 있으며 각 군 모두가 만족하며 지지하고 있다.

비고객을 고객화하는 것은 김위찬 교수와 르네 마보안 교수가 그들의 저서인 블루오션 전략(Blue Ocean Strategy)[10]에서 제시한 블루오션 전략의 하나이다. 우리는 이러한 블루오션 전략을 구체화하고 블루오

선 전략에서 제시하지 못하는 실제 시스템 또는 소프트웨어 개발에 적용할 체계적인 프로세스인 가치혁신 요구공학 프로세스를 개발하여 국제청 정보체계 등 다양한 프로젝트에 적용하여 성공적인 결과를 얻어내고 있다.

다음 절에서는 국방 소프트웨어의 가치혁신을 통하여 블루오션을 창출함으로써 제한된 재원으로 국방력을 효율적으로 증대시키는데 도움이 될 수 있는 방안을 제시하고자 한다.

### 3.2 가치혁신 요구공학 프로세스

#### 3.2.1 가치혁신 요구공학

가치혁신 요구사항(Value-innovative Requirements)은 고객의 새로운 가치를 포함하는 요구사항을 말한다. 가치요소는 새로운 잠재적인 시장을 창출할 수 있는 제품을 이끌어 낸다. 가치혁신 요구사항은 가치혁신 요구공학 프로세스에 의해서 만들어진다[9].

가치혁신 요구공학(ViRE)이란 고객의 가치를 분석하고, 가치창조 절차를 명세화 하고, 고객가치가 적절하게 반영되었는지를 평가하는 등의 반복적인 절차를 통하여 가치혁신 요구사항을 개발하는 프로세스를 말한다. 가치혁신 요구공학은 다음의 경우에 사용될 수 있다:

- 블루오션을 창출할 수 있는 신제품 개발 시
- 새로운 시장 공간을 이끌어 낼 수 있는 소프트웨어(시스템)의 요구사항 재정의 시
- 고객 가치를 증가시키기 위하여 2개 이상의 기존 시스템 요구사항을 결합할 때

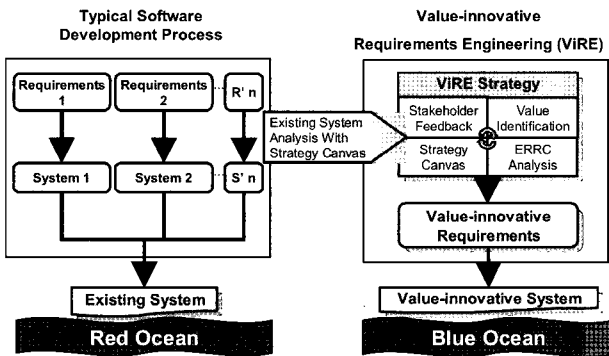


그림 5 VIRE를 이용한 시스템 개선

그림 5는 n개의 기존 시스템의 결합을 통하여 새로운 잠재적 시장인 블루오션을 창출할 수 있는 시스템을 개발하기 위한 개념을 나타낸다.

시스템 개선 프로세스는 먼저 기존 시스템의 요구사항들의 중복(redundancy)과 충돌(conflict)을 확인하고, 현재 시스템과 시장상황을 분석하여 전략 캔버스를

작성한다. 그 후 전략 캔버스를 블루오션 전략을 적용하여 새로운 시스템의 전략 캔버스와 그에 맞는 새로운 요구사항을 추출한다.

가치혁신 요구공학의 핵심은 ERRC 분석이다. 이를 통해 새로운 전략캔버스를 그리고 소프트웨어(시스템)의 새로운 요구사항을 만들어 내는 것이다.

#### 3.2.2 가치혁신 요구공학 프로세스

가치혁신 요구공학 프로세스는 그림 6에서 보는바와 같이 5단계로 구성되며, 전체 단계는 최소한 3번의 반복(iteration)을 권장하며 각각의 반복 과정에서는 중점적으로 초점을 맞추어야 하는 목표를 설정하였다. 첫 번째 반복 시에는 요구사항의 재정의의 중점 목표로 하고, 두 번째는 프로토타입, 세 번째는 제품 개발단계에서 이루어지는 사용자가 원하는 제품을 만들어 내는데 초점을 맞추어야 한다.

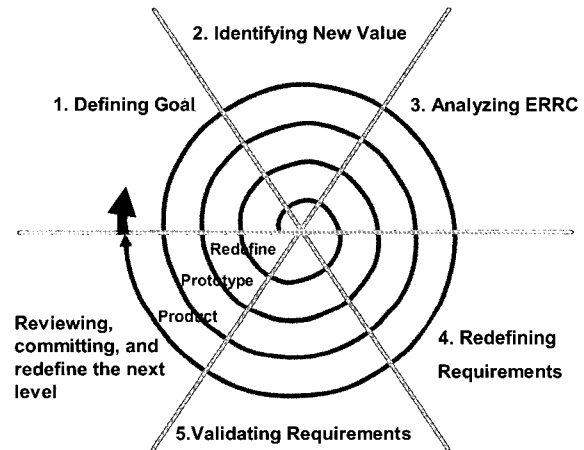


그림 6 가치혁신 요구공학 Spiral 모델

각각의 단계는 모두 입력(Input), 출력(Output), 제약사항(Constraints) 그리고 지원(Enabler)을 가지며 전 단계의 출력은 다음 단계의 입력으로 들어 갈 수 있다.

##### • 1단계: 목표 설정

프로젝트의 목표와 시스템의 범위를 정하고, 만들고자 하거나, 개량하고자 하는 시스템의 가치요소를 명확하게 한다.

##### • 2단계: 새로운 가치 식별

과거에 고려하지 못했던 다양한 가치요소를 찾아내기 위하여 고객의 잠재적인 요구를 추출하고, 고객의 가치요소를 식별하고 우선순위를 결정한다.

##### • 3단계: ERRC 분석

고객 가치요소를 재설정하기 위하여 고객의 주요 요구사항을 시스템에 할당하고 ERRC 분석, 협상 등의 과정을 거쳐 새로운 요구사항을 이끌어낸다.

•4단계:요구사항 재정의

앞의 세 단계를 통해 추출된 가치요소들을 토대로 전체 요구사항을 재정의하고 새로운 시스템에 대한 전략적 캔버스를 작성한다.

•5단계:요구사항 확인

재정의된 요구사항이 새로운 시스템을 구현하기 위해 적절하게 정의되었는지를 Axiomatic 접근법 등을 이용하여 검증한다.

3.3 가치혁신 요구공학 구현

3.3.1 ERRC 분석

전형적인 ERRC 분석은 몇 가지 단계를 거쳐 전개된다. 우리는 다음과 같은 3개의 단계를 적용한 방법을 논문 [9]와 그와 관련된 논문에서 소개한다.

- 1단계 : 고객 가치의 우선순위화
- 2단계 : 요구사항과 시스템요소를 매핑
- 3단계 : 고객 가치를 결정

이상의 단계들은 ViRE 프로세스의 1단계 후에 수행된다. 다시 말하면 프로젝트의 목표가 결정되고, 주요 이해관계자와 고객이 식별된 후에 이루어질 수 있다. 또한 고객 만족을 위한 요구사항이 미리 식별되어야 한다.

		SEs (How)					Importance Degree (CI)		
		SE1	SE2	SE3	SEj	SEn	Extremely important	5	
							Very important	4	
							Important	3	
							Little important	2	
							Not important	1	
							CI		
CRs (What)	CR1						Correlation	Importance	
	CR2								
	CR3								
	CRi				R <sub>ij</sub>				
	CRn								
ERRC Decision (ED <sub>j</sub> )	BI				BI <sub>j</sub>		Correlation Degree (R <sub>ij</sub> )		
	RC				RC <sub>j</sub>				
	ED				ED <sub>j</sub>				
Eliminate	X						Strong	●	9
Reduce	-						Moderate	⊙	3
Raise	+						Weak	○	1
Create	o								

그림 7 ERRC 결정 매트릭스

ERRC 결정 매트릭스 구조는 그림 7과 같다. 초기의 매트릭스는 Customer Attribute(CA), Customer Requirements (CR)등 고객 요구와 관련된 요소와 Correlation, SEs, Importance, ERRC Decision 등으로 구성된다. 그림 7에서 보는 바와 같이 ERRC 결정 매트릭스 1단계의 'How'가 2단계의 'What'으로

옮겨 감으로써 고객의 요구사항이 관련된 시스템 요소 (SE)에 대한 요구사항으로 구체화된다.

2단계의 ERRC 결정 매트릭스에 대한 5개의 주요 요소들은 다음과 같이 설명할 수 있다:

- Customer Requirements(CR<sub>j</sub>) : 1단계에서 정의된 Customer Attribute(CA)를 만족시킬 수 있는 고객 요구사항
- System Elements(SE<sub>i</sub>) : CRs를 만족하기 위해 필요시스템 요소
- Customer Importance(CI<sub>i</sub>) : 2단계에서 고객 요구사항의 우선순위화를 통해서 결정된 고객 중요도
- Correlation(R<sub>ij</sub>) : 고객 요구사항과 시스템 요소들 간의 관계
- ERRC Decision(ED<sub>j</sub>) : 시스템 요소들의 Business Importance (BI<sub>j</sub>)와 Relative Cost (RC<sub>j</sub>) 그리고 시스템 요소의 Eliminateing (X), Reducing (-), Raising (+), Creating (O) 등과 같은 제안되는 ERRC 행위를 나타낸다.

ERRC 결정 매트릭스는 제안된 ERRC 행위를 통해 새로운 시스템 요소들의 묶음을 재결정하는데 목적이 있다. 시스템 요소들에 대한 ERRC 행위를 결정하기 위하여 먼저 다음과 같이 BI<sub>j</sub>를 계산해야한다:

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

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

둘째, SEs의 Relative Cost (RC<sub>j</sub>)를 다음의 식을 통해 계산하여야 한다:

$$RC_j = (Cost\ of\ SE_j / Cost\ of\ total\ system) \times 100$$

셋째, BI<sub>j</sub>와 RC<sub>j</sub>를 고려하여 최종적으로 ERRC 행위가 제안된다. 이를 위해서는 특별하게 정해진 알고리즘은 없다. 하지만 국방분야의 가치를 창출하기 위한 시스템을 개발하기 위한 목적을 달성하기 위해서 ROI, WinWin 접근법 그리고 이 두 가지를 모두 고려한 Hybrid 접근법을 적용 할 수 있다.

3.3.2 Axiomatic 요구사항

Axiomatic 방법은 요구사항의 충돌을 찾아내고 검증하기 위한 구조를 제공하는 과학적인 이론을 기반으로 한 체계적 접근법이다. 이 접근법에서는 요구사항과 설계 프로세스 중 고객 가치창출 요소와 고객의 요구사항을 포함하는 고객 도메인과 시스템의 요구사항을 포함하는 시스템 도메인을 포함한다. 각각의 도메인에

서는 요구사항을 계층적으로 하위단계까지 분해하여 식별하게 되는데 그림 8과 같이 지그재그 형식으로 계층적으로 식별해 나가며 매핑하게 된다[11].

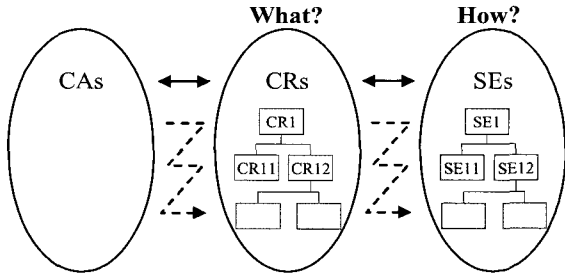


그림 8 CA와 SE 간의 매핑

Axiomatic 방법에서 가장 중요한 것은 매핑 프로세스 동안 2가지는 만족되어야 한다는 것이며, 2가지 axiom은 다음과 같다[11]:

- **Axiom 1**: The Independence Axiom(기능 요구사항은 독립성을 유지해야 한다.)
- **Axiom 2**: The Information Axiom(정보 콘텐츠를 최소화해야 한다.)

먼저 고객 요구사항이 정의되면 시스템 파라미터가 식별되고, 고객 도메인과 시스템 도메인 사이의 매핑 프로세스는 다음과 같은 설계 방정식으로 나타낼 수 있다.

$$\{CRs\} = [R]\{SEs\} \quad (1)$$

설계 매트릭스  $R_{ij}$  는 다음과 같다.

$$R_{ij} = \frac{\partial CR_i}{\partial SE_j} \quad (2)$$

ERRC 분석단계에서 그림 8에 정의된 ERRC 결정 매트릭스의 Correlation 매트릭스를 CR들 간의 충돌과 부적절한 SE를 점검하기 위해 사용한다. 초기의 정리되지 않은 Matrix를 요소들의 위치를 변경함으로써 Diagonal Matrix, Triangular Matrix 그리고 그 외의 Matrix 등 3가지 형태로 변경할 수 있다.

Independence Axiom을 만족하기 위해서는  $[R]$ 이 Diagonal Matrix 또는 Triangular Matrix를 만족해야 한다.  $[R]$ 이 Diagonal Matrix인 경우를 Decoupled Design이라 하고,  $[R]$ 이 Triangular Matrix인 경우를 Uncoupled Design이라 한다.

$$\begin{Bmatrix} CR_1 \\ CR_2 \\ CR_3 \end{Bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ 0 & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \times \begin{Bmatrix} SE_1 \\ SE_2 \\ SE_3 \end{Bmatrix} \quad (3)$$

$$\begin{Bmatrix} CR_1 \\ CR_2 \\ CR_3 \end{Bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \times \begin{Bmatrix} SE_1 \\ SE_2 \\ SE_3 \end{Bmatrix} \quad (4)$$

$$\begin{Bmatrix} CR_1 \\ CR_2 \\ CR_3 \end{Bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \times \begin{Bmatrix} SE_1 \\ SE_2 \\ SE_3 \end{Bmatrix} \quad (5)$$

디자인 매트릭스가 식(3) 또는 식(4)를 따르지 않고 식(5)와 같을 때 이것을 Coupled Design이라고 한다 [11]. Coupled Design의 경우 Axiomatic 접근법을 만족하지 못하고 Axiom 1에 위배된다. 이것은 시스템의 요소가 잘못 선정되었거나 고객 요구사항 또는 시스템 요소 간의 충돌을 일으키는 경우이다. 따라서 CR과 SE를 재조정해야 한다. Diagonal Matrix(4) 또는 Triangular Matrix(5)를 만족하도록 하게 함으로써 요구사항 간의 충돌을 해결하는 것이다.

#### 4. 국방 요구공학 센터의 필요성

이상에서 소개한 내용들은 가치를 기반으로 한 요구공학 절차를 국방 소프트웨어 개발에 적용하여 전투력 향상과 경제적인 군 운영에 도움을 주기 위한 방안이다. 하지만 기존의 국방 획득시스템과 인력을 통해서 체계적으로 적용하기에는 한계가 있을 것이다.

기존 인력의 전문화와 제도의 정비 및 아웃소싱을 통하여 이러한 한계점을 극복할 수 있다. 하지만 민간 분야 기술의 비약적인 발전과 새로운 기술의 접목을 통한 프로세스의 개선과 체계적인 분석을 위해서는 특화된 연구센터의 운영이 가장 효율적일 것이다.

국방 소프트웨어는 점점 복잡해지고 대형화되고 있으며 전체 시스템에서의 소프트웨어 비중이 점차 높아지고 있다. 또한 개발과 운용 환경이 다양화, 분산화, 멀티미디어화되고 있고 국방정보화 사업에 있어서 사용자 요구사항의 명세가 더욱 어려워지고 있다. 따라서 국방 소프트웨어 개발 시 효과적으로 요구사항을 수집, 분석, 명세, 검증, 관리하기 위해서는 요구공학 분야의 연구를 전담할 수 있는 요구공학 연구센터를 지정하여 운영하는 방안이 필요하다.

요구공학 연구센터는 국방대학교, KIDA, ADD와 같은 군 유관기관에 설치하거나 민간 연구센터를 지정하여 설치할 수 있다. 하지만 요구공학 연구센터를 군 유관기관에 설치하는 것은 바람직하지 않은데 그 이유는 요구공학 관련 전문인력이 군 유관기관에 부족하고, 참가자들에 대한 인센티브가 존재하지 않아 활성화되기 어렵기 때문이다. 과거 요구공학 연구실을 국방대에 비편제로 잠시 운용한 사례가 있으나 전문인력

부족과 참가자들의 참여부족으로 폐지된 사례가 있다. 민간 연구기관에도 군 관련 정보체계, 조선, 원자력, 발전소, 항공기 제작 등 대형 프로젝트를 수행하면서 요구공학 관련 업무를 수행한 경험이 있는 인력과 기관들이 있으므로 이들을 활용하여 국방 소프트웨어 요구공학 연구센터를 설립하는 것이 바람직하다[12].

민간의 국방 요구공학 센터는 업체 혹은 국내 대학의 국방 분야에 특성화된 전문 연구기관 중에서 국방 부에서 지정한 전문 연구기관의 자격, 평가기준 등을 사전에 제시하여 요구공학을 전문적으로 수행할 대학을 2~3개 정도로 선정할 수 있다. 국방부에서는 국방 정보체계나 무기체계와 같은 국방 소프트웨어 관련 사업을 추진할 때에 제안요구서(RFP)에 지정된 민간 요구공학 센터와 컨소시엄을 형성하여 참여할 수 있도록 조치를 취할 수 있으며 지정된 요구공학 센터는 2~3년 주기로 평가를 통해 재지정 할 수 있다[12].

이러한 형태로 운영되는 민간 요구공학 센터가 존재한다면 국방 소프트웨어 관련 사업을 수주한 업체는 국방 요구공학 센터의 지원을 받아 군이 요구하는 기능 및 성능을 보다 잘 만족시키는 시스템을 개발할 수 있고, 군 주도의 국방 소프트웨어 개발을 보다 효과적으로 할 수 있다. 또한 국방 분야에 특화된 요구사항 자료가 축적되면서 국방 소프트웨어 요구공학 연구가 활성화되고 나아가 국방 소프트웨어의 생산성과 품질 향상에 기여할 것으로 기대된다.

## 5. 결 론

잘 정의된 요구사항은 좋은 시스템을 만들고 사용자들을 만족시킬 뿐만 아니라 좋은 공급자를 낳는다. 즉 국방 획득 상에서의 요구사항은 좋은 시스템을 만들수 있을 뿐만 아니라, 각 군의 사용자들을 만족시키며, 전투력 향상에 기여하고 결국 개발자 및 공급자에게도 이익을 가져다준다.

무기체계 소프트웨어 및 정보체계 기술의 발달, 획득환경의 변화에 따라 요구공학의 중요성은 아무리 강조해도 지나치지 않는 것이다. 제한된 재원으로 전투력을 극대화하고 효율적인 군 운영을 위해서는 요구공학 절차 적용을 통한 효율적 국방 소프트웨어 획득관리 및 비용절감을 꾀하지 않으면 안 된다. 특히, 가치창출을 위한 요구공학 방법론은 이러한 국방 분야 무기체계 및 자동화 정보체계의 개발에 있어서 반드시 적용되어야 할 방법론이다. 이를 통하여 우리 군은 효과적인 무기체계 및 자동화 정보체계의 획득으로 전투력 향상과 경제적 군 운영을 도모할 수 있을 것이다.

국방시스템의 개발에 적용될 수 있는 다양한 방법론

과 이론적인 연구는 많이 성숙되어 있으며 국방 분야의 수요만을 기다리고 있다. 실질적인 국방 분야의 적용을 위해서는 요구공학에 대한 체계적인 연구 및 제도화가 필요하다. 군·민 전문가들로 구성된 요구공학 지원체계 확립은 세계최고의 IT 강국을 자부하고 있는 우리의 민간기술을 국방분야에 적용하여 효율적인 군 운영과 전투력 향상에 기여할 수 있는 지름길일 것이다.

## 후 기

본 글의 교신저자는 고려대학교 컴퓨터학과 인호 교수(hoh\_in@korea.ac.kr)와 서강대학교 박수용 교수(sypark@sogang.ac.kr)입니다.

## 참고문헌

- [1] Boem, B. W., et al. "Some Experience with Automated Aids to the Design of Large-Scale Reliable Software." IEEE Transactions on Software Engineering 1, 1(March 1975): 125-33.
- [2] Standish Group, "CHAOS," Standish, 1995. <http://www.standishgroup.com/chaos.htm>.
- [3] Alan M. Davis, Software Requirements: analysis & Specification, Prentice-hall, 1990.
- [4] Berry W. Boehm, "Value Based Software Engineering," ACM SIGSOFT Software Engineering Notes, vol.28, no.2 March, 2003.
- [5] J. Bullock, "Calculating the Value of Testing," Software Testing and Quality Engineering, May/June, pp.56-62, 2000.
- [6] J. Thorp and DMR, The Information Paradox, McGraw Hill, 1998.
- [7] PMI, PMBOK: Project Management Body of Knowledge, Project Management Institute, 2000.
- [8] B. Boehm, "Software Risk Management: Principles and Practices", IEEE Software, Vol.8, No.1, pp.32-41, January, 1991.
- [9] 김상수, 임상원, 박용식, 인호 "블루오션 전략을 적용한 요구공학 프로세스", 한국시스템엔지니어링학술지, 제2권 1호, pp.11-17, 2006.
- [10] W. Chan Kim, Renee' Mauborgne, "Blue Ocean Strategy," Harvard Business School Press, 2005.



- [11] Basem S. El-Haik, "Axiomatic Quality,"  
Wiley- Interscience, 2005.
- [12] 김화수, 군주도 국방 정보체계 건설 및 운영유지  
관리 방향(안) 연구, 국방부 용역연구 보고서,  
2004. 1.

### 김 상 수



2004 국방대학교 무기체계 M&S(석사)  
1993~1997 공군군수사령부 번역장교  
1997~2002 전투비행단 중대장 및 참모  
2004~2005 공군전투발진단 무기체계처  
2005~현재 고려대학교 컴퓨터학(박사과정)  
1993~현재 공군소령(항공무기장비장교)  
관심분야: 요구공학, 임베디드 소프트웨어  
공학, 모델링 시뮬레이션(M&S)  
E-mail : sookim@korea.ac.kr

### 인 호



1990 고려대학교 전산학과(학사)  
1992 고려대학교 전산학과(석사)  
1998 USC Computer Science(박사)  
1993~2003 Texas A&M University  
조교수  
2003~현재 고려대학교 컴퓨터학과 조교수  
관심분야: 요구공학, 임베디드 소프트웨어  
공학, 소프트웨어 보안, 정황인  
지 미들웨어  
E-mail : hoh\_in@korea.ac.kr

### 최 순 황



2001 서강대학교 컴퓨터학(학사)  
2003 서강대학교 컴퓨터학(석사)  
2003~현재 서강대학교 컴퓨터학(박사과정)  
관심분야: 요구공학, 프로덕트 라인, 소프  
트웨어 프로세스, 소프트웨어  
아키텍처  
E-mail : soonhwang@sogang.ac.kr

### 박 수 용



1986 서강대학교 전자계산학과(공학사)  
1988 플로리다 주립대 전산학(석사)  
1995 George Mason University  
정보기술학(박사)  
1995 George Mason University 연구  
조교수  
1996~1998 TRW Senior Software  
Engineer  
1998~2002 서강대학교 컴퓨터학과  
조교수  
2002~현재 서강대학교 컴퓨터학과 부교수  
2005~현재 국방소프트웨어산학연협회 총무이사  
관심분야: 요구공학, Adaptable Components, Web Services  
E-mail : sympark@sogang.ac.kr