

요구사항 변경이 확정가 프로젝트 계획에 미치는 영향

이 상 운[†]

요 약

요구사항이 빈번히 변경되는 동적 비즈니스 시스템 프로젝트를 확정가로 개발할 경우, 개발과정에서 발생하는 요구사항의 변경에 따른 추가 소요 비용으로 인해 확정가 범위 이내로 개발을 완료하는 것은 거의 불가능하다. 확정가 범위 내에서 성공적으로 프로젝트를 완료하기 위해서는 요구사항 변경 규모와 이에 기반하여 개발이 되지 않아도 되는 옵션 요구사항의 비율을 관리하여야 한다. 이에 대해 Bhagwat는 요구사항 변경이 구축단계에서만 발생한다고 한정하였다. 또한, 소프트웨어 개발비용, 구축단계 소요 비용과 요구사항 변경에 소요되는 추가 비용이 동일하다고 가정함으로써 요구사항 변경 비율과 옵션 요구사항의 비율 유도의 모순을 보였다. 본 논문은 요구사항 변경은 정련단계와 구축단계에서 발생한다고 가정하였다. 또한, 소프트웨어 개발비용, 구축단계 소요 비용과 요구사항 변경 비용간에는 차이가 발생한다고 가정하였다. 그 결과 확정가 대비 요구사항 변경에 따른 추가 소요 비용과 개발이 되지 않아도 되는 옵션 요구사항의 비율도 보다 현실성을 반영하는 결과를 얻을 수 있었다.

키워드 : 요구사항 변경, 확정가, 옵션 요구사항, 폭포수 프로세스, 반복 프로세스

The Effect of Requirement Creep on the Fixed-Cost Project Planning

Sang-Un Lee[†]

ABSTRACT

To develop a dynamic system project in which the requirement changes frequently, it's impossible to finish the development within a fixed-cost due to additional budget occurring in need of requirement creep. To manage the successive project within a fixed-cost, it's better to manage the ratio of necessarily changed size of project and necessary optional requirement. According to Bhagwat, it is occurred in the construction phase. Also, he stated that the software development cost, construction phase cost and requirement cost are equal and it was wrong explanation in the ratio of requirement creep and optional requirement. This paper assumes the requirement creep to be happening in the phase of elaboration and construction. In addition, some differences were supposed to happen between software development cost, construction phase cost, and requirement creep cost. As a result, the reality was preferred rather than the ratio of optional requirement and the ratio of requirement creep.

Key Words : Requirement Creep, Fixed-Cost, Optional Requirements, Waterfall Process, Iterative Process

1. 서 론

소프트웨어 개발계획 작성시 개발될 소프트웨어의 규모(Size), 개발에 투입되는 비용(또는 시간)과 개발 기간을 신뢰할 수 있는 수준으로 추정하는 어려움에 직면한다. 소프트웨어 규모 추정에 관해서는 COCOMO[1], COCOMO-II[2], 기능점수(Function Point)[3, 4], COSMIC-FFP[5], 쓰임새점수(Use Case Point) 기법[6] 등이 적용되고 있다. 추정된 소프트웨어 규모에 기반하여 소프트웨어 개발에 소요되는 총 인시수(Manhour)와 개발기간(Duration)을 추정하는 모델들이 다양하게 연구되고 있다. 이러한 정보를 토대로 입찰과 계약에 대한 의사결정이 이루어진다. 다양한 계약방식들 중에서 확정가(Fixed-Cost) 계약 방법은 조기에 이윤

을 추구할 수 있다는 망상을 가지고 확정된 선행투자를 할 수 있게 해주기 때문에 고객이 가장 선호하는 방법이다. 그러나 개발될 소프트웨어의 요구사항과 복잡도에 대해 완전한 이해를 하지 못하는 동적 비즈니스 프로젝트라면 개발비용을 신뢰할 수 있는 수준으로 정확히 예측하는 방법이 없는 것이 확정가 프로젝트의 역설(Fixed-cost Project Paradox)이다[7].

계약이 성립되고 개발이 진행되면서 대부분의 비즈니스 시스템 프로젝트들이 요구사항 변경(Requirements Creep)을 경험하고 있다[8]. 요구사항 변경이란 개발과정에서 요구사항에 대한 중요한 추가(Addition) 또는 수정(Modification)으로 인해 소프트웨어의 기능(Functionality) 또는 범위(Scope)가 확장(Extension)되거나 대체(Alternation)되는 것을 말한다. 요구사항 변경으로 인해 새로운 요구사항을 적시에 구체화시킨 시스템을 납품하기 위해서는 비용, 자원, 품질 또는 능력이 추가로 필요로 하므로 적절히 관리되지 못하면 개발자에

[†] 정 회 원 : 강릉대학교 컴퓨터공학부 조교수
논문접수: 2007년 4월 11일, 심사완료: 2007년 5월 22일

게 특히 골치 아픈 것이 될 수 있다[9]. Capers[10]는 새롭거나 예상되지 못한 요구사항들이 매일 총 요구사항의 1%의 비율로 추가로 발생한다고 제시하였다. 또한, Lieberman[11]는 요구사항의 대부분은 도입단계 (Inception Phase)와 정련단계 (Elaboration Phase)에서 수집되고 구축단계 (Construction Phase)에서 요구사항 변경 또는 추가로 인해 소프트웨어 시스템에 보다 큰 영향을 미치기 때문에 특히 시스템 아키텍처에 대한 위험에 대한 충분한 관리가 필요함을 제기하였다.

주어진 확정이 범위내에서 프로젝트를 성공적으로 완료하려면 요구사항 변경에 따른 추가 소요 비용을 고려하여 개발이 되지 않아도 되는 요구사항들을 분류하여 관리하는 방법을 선택해야만 한다. 이에 대해 Bhagwat[7]는 COCOMO II 모델[2]로부터 반복 프로세스 (Iterative Process)의 구현 단계에서 요구사항 변경 분포를 유도하고 요구사항 변경으로 인한 추가비용이 얼마인지와 개발이 되지 않아도 되는 옵션 요구사항의 비율을 제시하였다. 그러나 Bhagwat[7]가 제시한 이론은 요구사항 변경이 발생하는 단계 적용의 모호함, 요구사항 변경의 비율의 오 적용 등 문제점을 내포하고 있다. 따라서, 본 논문은 이 분야에 대한 보다 이론적으로 타당한 모델을 제시한다.

2장에서는 확정이 프로젝트를 관리하는 방법을, 3장에서는 Bhagwat[7]가 제시한 요구사항 변경의 비율과 옵션 요구사항에 관한 연구 결과와 문제점을 살펴본다. 4장에서는 Bhagwat[7]가 제시한 이론의 요구사항 변경이 발생하는 단계 적용의 모호함, 요구사항 변경의 비율의 오 적용 등 문제점을 개선한 이론을 제시하고 제안된 이론의 적합성을 검증한다.

2. 확정이 프로젝트 관리

2.1 개발방법론별 요구사항 변경관리 방법

전통적인 순차적 (Sequential) 또는 폭포수 (Waterfall) 프로세스는 개발 초기의 요구사항 단계 (Requirement Phase)에서 모든 요구사항들을 상세히 확정한 후 계약을 체결한다. 이로 인해 이후 개발단계에서 새로운 요구사항이 추가되거나 수정됨으로 인해 요구사항 변경이 발생하면 적절히 대처하지 못하는 전형적인 문제점을 내포하고 있다.

요구사항 변경에 대한 필요성이 발생하여 프로젝트 목표가 변경되기까지 약 80%의 작업이 순조롭게 진행된다. 이후 프로젝트 납품시점이 빠르게 다가오면 막연히 개발을 계속 수행하는 방법 또는 간신히 자금자축할 수 있는 시스템을 납품하는 방법 중에서 하나를 선택한다. 이의 근거로 CHOAS 보고서에 의하면 전형적인 IT 프로젝트의 성공률은 30% 이하이다[12].

이에 비해, 유스케이스 구동 (Use-case Driven) 반복개발 (Iterative Development) 접근법은 프로젝트를 본 궤도로 유지하도록 보다 나은 대안을 제공하고 있다. 이 방법은 프로젝트 시작 시점에서는 요구사항들이 거의 완전히 이해되지 않았으며, 단지 확실한 사실은 개발과정에서 요구사항이 변경되는 것을 다룰 수 있다는 것이다. 기능 요구사항 (Functional Requirement)을 가장 적절히 발췌할 수 있는 방법이 유스케이스 기법이다. 따라서, 반복개발 프로세스에서는 유스케이스 기법을 적용하고 있다. <표 1>은 Larman[13]이 제시한 반복개발 프로세스를 적용하여 요구사항에 투입되는 노력에 대한 사례이다. <표 1>의 사례에서 알 수 있듯이 사실상 모든 유스케이스들의 목록을 도입단계 (Inception Phase)에서 명확히 제시하고 다음에 수행되는 정련단계 (Elaboration Phase)의 반복 (Iteration) 들에서 일정한 양의 요구사항들이 상세히 기술된다. 또한 상세한 기술이 완료되거나 확정이 이루어진 유스케이스들은 다음에 수행되는 구축단계 (Construction Phase)의 반복들에서 실제로 구현되어 실현된다.

개발 초기에 확정이 계약을 할 경우 반복개발 프로세스 접근법은 개발비용을 추정하는 공식에서 정확한 비용산정 (Accurate Costing)이 되지 않는 단점을 갖고 있다. 만약 개발 조직이 이 접근법을 채택한다면 <표 1>의 프로젝트 관리 Discipline과 같이 아키텍처가 고안되고 구체화되면서 계속되는 반복들의 시작 시점에서 개발비용 추정 정확성에 대한 보완작업이 수행되어야만 한다. 그러므로 이 방법은 전체 추정 과정의 정확성은 요구사항의 불명확성보다는 요구사항 변경에 따른 영향에 보다 중요하게 초점을 맞추고 있다. 이와 같은 이유로, 나중에 수행되는 반복들에서 구현되도록 할당된 유스케이스들은 먼저 수행되는 반복에서 보다 많은 변경이 발생할 경향이 있어 반복이 진행됨에 따라 개발비용 추정의 정확성은 낮아진다.

<표 1> UP를 적용한 요구사항 투입노력 사례

Discipline	산출물	Inception (1주)	Elaboration			
			E1 (4주)	E2 (4주)	E3 (3주)	E4 (3주)
요구사항	유스 케이스 모델	유스케이스 목록 식별 10% 상세기술	30% 상세 기술	50% 상세 기술	70% 상세 기술	80~90% 상세 기술
설계	설계 모델	-	설계 시작	계속 수행	계속 수행	계속 수행 고 위험도와 구조적 중요 부분 안정상태
구현	구현 모델	-	구현 시작	5% 구축	10% 구축	15% 구축
프로젝트 관리	개발 계획	총 비용 개략 추정	추정 구체화	추정 구체화	추정 구체화	전체 프로젝트 개발 기간, 노력, 비용의 합리적 추정

2.2 확정이 계약의 프로젝트 관리

확정이 계약시 프로젝트 계획은 순조로운 항해(Smooth Sailing)를 하는 경우와 폭풍이 심하여 결국에는 배를 버리는 경우를 고려할 수 있다[7]. 첫 번째 방법인 순조로운 항해는 일단 확정된 요구사항은 변경이 발생하지 않아 확정가 범위 내에서 프로젝트를 완료할 수 있는 경우로 현실적으로 거의 불가능하다. 이에 비해, 두 번째 방법은 요구사항 변경으로 인해 초기 계획대로 프로젝트 개발을 하지 못해 실패하는 경우이다. 두 번째 방법의 문제점을 해결하기 위해서는 최악의 경우 배를 구하기 위해 귀중한 짐들을 물 속으로 버리는 위기를 극복하기 위해 사전에 개발이 되지 않아도 되는 요구사항들을 옵션으로 분류하여 관리하는 것이 필요하다. 대부분의 프로젝트 계획은 요구사항의 잦은 변경으로 인해 이 방법을 적용하여야만 프로젝트를 성공적으로 완료할 수 있다.

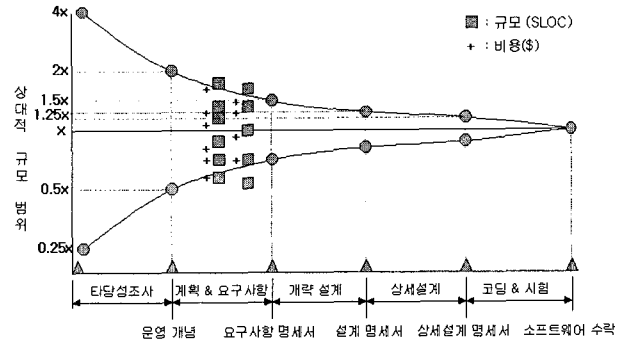
순조로운 항해를 한다고 가정하였을 경우 개발에 소요되는 비용을 확정이 개발비용 x 라 하자. 이후에 개발과정에서 요구사항 변경으로 추가로 소요되는 비용 Δx 가 포함되면 총 개발비용은 $x + \Delta x$ 로 증가한다. 이는 확정이 x 의 범위 내에서는 프로젝트가 고객에게 $x/(x + \Delta x)$ 분량의 요구사항 들만을 납품할 수 있으므로 $1 - \{x/(x + \Delta x)\} = \Delta x / (x + \Delta x)$ 의 요구사항들은 개발되지 않을 수 있다. 이 부분이 개발이 되지 않아도 되는 옵션으로 개발 초기에 결정될 수 있다면 프로젝트는 계약단계에서 추정된 확정이 x 범위 내에서 사실상 완료될 수 있다고 할 수 있다. 동적으로 진화하는 프로젝트는 요구사항이 빈번히 변경되는 특성이 있음에도 불구하고 사전에 이러한 옵션 요구사항들을 확정하지 않고 무작위로 선택된 순서대로 요구사항들을 개발할 경우, 확정가 범위를 초과하는 시점에서 중요한 요구사항들이 아직 개발되지 않는 경우가 발생할 수 있다. 따라서, 배를 구하기 위해 (확정이 범위 내에서 개발을 종료) 귀중한 짐들 (중요한 요구사항)을 물 속으로 버리는 위기시점까지 돌진할 수 있다. 그러므로 제한된 확정가 비용 x 범위 내에서 개발이 이루어 지도록 사전에 요구사항들의 중요도를 결정하여 개발 우선순위를 부여하고 회생될 수 있는 옵션들을 분류할 필요가 있다. 또한 옵션들로 분류할 요구사항이 얼마인지를 알 필요가 있다.

3. 요구사항 변경 비율과 옵션 요구사항 비율

3.1 관련 연구

본 절은 Bhagwat[7]가 제안한 확정이 범위 내에서 프로젝트를 완료하기 위해 요구사항 변경에 소요되는 비용과 개발이 되지 않아도 되는 옵션 요구사항의 비율에 관한 연구 결과를 살펴본다.

(그림 1)은 COCOMO II 모델[2]에서 인용된 것으로 프로젝트의 불확실성이 소프트웨어의 규모와 비용 추정에 미치는 영향을 나타내고 있다. 프로젝트 초기 단계에서 관리자는 개발될 프로젝트의 구체적인 특성을 알지 못해 실제 개발될 소프트웨어 규모와 비용에 대해 4배 정도의 추정 오차



(그림 1) 단계별 소프트웨어 비용과 규모 추정 정확도

를 유발시킬 수 있다.

즉, 새로운 영역에서 전례가 없는 새로운 제품을 얻기 위한 소프트웨어 개발 프로젝트의 경우 타당성조사 단계 (Feasibility Phase)의 시작시점에서 요구사항 변경으로 추가되는 총 비용 Δx 는 $3x(x + \Delta x = 4x)$ 정도로 예측된다. 타당성조사 단계가 종료되고 기본적인 아키텍처가 가시화되면 Δx 는 $x(x + \Delta x = 2x)$ 가 된다. 이는 친숙하거나 성숙된 영역에서 동적으로 진화하는 비즈니스 시스템의 프로젝트 시작 시점에서 생각할 수 있다. 프로젝트가 진행되면서 Δx 는 점점 느린 속도로 감소하는 경향이 있으며, 기본적인 아키텍처가 결정되고 주요 구현이 시작된 이후에는 거의 선형적으로 감소한다. 이는 구조 완성, 상세 설계와 진정한 구현이 반복적이고 점진적으로 발생하는 단계이며, 또한 프로젝트의 구조에 큰 영향을 미치지 않는 범위 내에서 요구사항 변경이 최고로 발생하는 기간이다.

구현단계에 수행되는 반복들의 총 수를 n , 요구사항을 실현하는데 소요되는 기본적인 비용 (구현에 할당된 비용)을 x , i 번째 반복에서 기본적으로 소요되는 비용 (i 번째 반복에 할당된 비용)을 $x_i (i=1, 2, \dots, n)$ 라 하자. 또한, i 번째 반복 I_i 에서 요구사항 변경으로 인해 추가로 투입이 예상되는 비용을 Δx_i , 요구사항 변경으로 인해 임의의 반복에서 추가로 소요가 예상되는 비용의 총합을 $\Delta x (= \sum \Delta x_i)$ 라 하자. 요구사항 변경이 발생하는 동적으로 진화하는 비즈니스 시스템과 관련된 전형적인 프로젝트의 구현에 대해 다음과 같이 가정한다.

- (1) 구현단계의 각 반복에서 투입되는 비용은 거의 동일한 수준이다.
- (2) 구현 단계의 첫 번째 반복에서 Δx 는 x 이다. 즉, 구현단계에서 요구사항 변경으로 인해 소프트웨어 개발 총 비용만큼 추가로 비용이 소요된다.
- (3) 반복 수 n 에 대한 Δx 는 선형적으로 감소한다.
- (4) i 번째 반복이 시작되는 시점에서 이 반복에서 구현되는 요구 사항들은 확정되어 있다고 가정한다. $\Delta x_i = 0$.

위 가정들에 기반하면, I_1 에서 I_n 까지 Δx 의 선형적 감소 관계에 따라 $\Delta x = x$ 가 $n-1$ 번 감소하므로 각 반복에서는

Δx_i 가 $x/(n-1)$ 씩 감소한다. 즉 i 번째 반복 I_i 의 시작시점에서 j 번째 반복까지 요구사항 변경으로 추가로 소요되는 총 비용 Δx 는 $x - \{(i-1)x/(n-1)\} = x\{1 - (i-1)/(n-1)\} = x\{(i-1)/(n-1)\}$ 이 된다.

나중에 수행되는 반복들에서 구현되도록 할당된 유스케이스들은 먼저 수행되는 반복에서 보다 많은 변경이 발생할 경향이 있으므로 Δx_i 값들은 i 가 증가함에 따라 선형적으로 증가한다고 가정한다. i 번째 반복 I_i 를 기준으로 했을 때, 이 반복시점부터 시작하여 이후에 수행되는 반복을 j 라 하면 j 번째 반복에서 요구사항 변경으로 추가로 요구되는 비용은 Δx_j 가 된다. 즉, 첫 번째 반복 I_1 을 기준으로 하면 $i=1$ 이 되며 $j=1, 2, \dots, 6$ 으로 남아있는 반복 수를 의미한다. $j=1$ 반복에서는 요구사항들의 변경이 발생하지 않으므로 $\Delta x_j = 0$ 가 된다. 다음 반복부터는 요구사항 변경으로 추가로 비용이 발생하기 때문에 $\Delta x_{j+1} = \Delta x_2 = y$ 라 하면, $\Delta x_{j+2} = 2y, \dots, \Delta x_n = \Delta x_{j+(n-j)} = y(n-j)$ 가 된다. 즉, $\Delta x_i = y(i-j) = \Delta x_{j+1}(i-j) \cdot I_j$ 반복 시작 시점에서 $\Delta x = \sum_{i=1}^n \Delta x_i = \Delta x_{j+1} \sum_{i=j+1}^n (i-j) = x\{(n-j)/(n-1)\}$, $\Delta x_{j+1} = x\{(n-j)/(n-1)\} / \sum_{i=j+1}^n (i-j)$ 가 된다.

예로 확정가가 x 이고, 6개의 반복($n=6$)으로 구현한다면 <표 2>와 같이 각 반복에서 요구사항 변경으로 인한 총 추가비용과 옵션으로 분류할 요구사항의 비율을 결정할 수 있다. 반복 2의 시작 시점에서 볼 때, 반복 3에서 요구사항 변경으로 추가로 소요되는 비용은 전체 비용 추정치의 8% ($=2x/25$)가 된다. 이후 반복들에서는 각각 16%, 24%, 32%의 추가 비용이 발생하므로 확정가의 80% 비용이 추가로 소요된다. 만약 반복 2 까지 성공적으로 수행되고 반복 3의 시작시점에 도달하였다면, 이후 반복들에서 요구사항 변경으로 10%, 20%, 30%의 추가비용이 소요되어 확정가의 60%가 추가로 소요됨을 알 수 있다.

만약 $\Delta x/(x + \Delta x)$ 만큼이 옵션 요구사항들로 취급될 수 있다면 확정가 비용 x 로 프로젝트의 성공을 보장한다. 첫

번째 반복 시작 시점에서 볼 때, 옵션 요구사항의 비율은 $\Delta x/(x + \Delta x) = 100/(100 + 100) = 50\%$ 가 되며, 2번째 반복이 시작되는 시점에서는 $80/(100 + 80) = 44.4\%$ 로 감소한다. 이후의 각 반복 시작시점에서 37.5%, 28.6%, 16.6%와 0%로 각각 감소한다.

3.2 문제점

Bhagwat[7]가 제시한 이론의 문제점을 살펴보자. UP의 단계와 폭포수 프로세스의 단계는 정확히 일치하지 않는 것이 일반적인 사실이다.[14] 그럼에도 불구하고 기존 제안 이론이 폭포수 프로세스를 적용한 COCOMO II 모델[2]의 데이터를 근거로 UP에 적용하였으므로 두 가지 개발 프로세스의 단계들을 개략적으로 일치하도록 상호 매핑시켜 보면 다음과 같은 문제점이 도출될 수 있다.

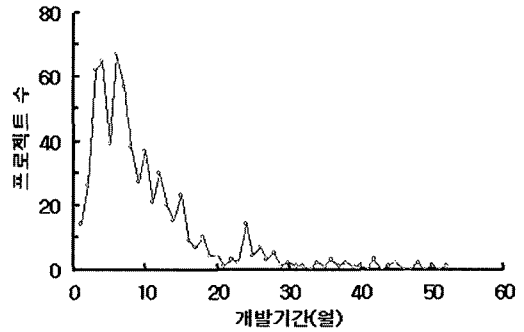
- (1) 타당성조사 단계가 종료되고 기본적인 아키텍처가 가시화되면 Δx 는 $x(x + \Delta x = 2x)$ 가 되며, 친숙하거나 성숙된 영역에서 동적으로 진화하는 비즈니스 시스템의 프로젝트 시작 시점으로 생각할 수 있다. 이 이론은 순차적인 폭포수 프로세스에 관련된 단계를 의미하며, 반복 프로세스에서는 “도입단계” 완료시점으로 생각할 수 있다.
- (2) 기본적인 아키텍처가 결정되고 주요 구현이 시작된 이후에는 구조 완성, 상세 설계와 진정한 구현이 반복적이고 점진적으로 발생하는 단계이며, 또한 프로젝트의 구조에 큰 영향을 미치지 않는 범위 내에서 요구사항 변경이 최고로 발생하는 기간으로 Δx 가 거의 선형적으로 감소한다. 여기서 의미하는 구현이 시작된 이후는 반복 프로세스의 “구축단계”로 생각할 수 있으며, 구조 완성, 상세설계와 구현이 일어나는 단계의 의미로 볼 때 폭포수 프로세스는 “상세설계단계”부터 시작됨을 알 수 있다.
- (3) 구현 단계의 첫 번째 반복에서 요구사항 변경으로 추가되는 비용 Δx 는 x 이다. 즉, 구현단계에서 요구사항 변경으로 총 비용과 동일한 비용이 소요되므로 총 개

<표 2> 요구사항 변경에 따른 반복별 개발비용 증가와 옵션 요구사항

구분		시작시점					
		반복 1	반복 2	반복 3	반복 4	반복 5	반복 6
남아있는 반복수		6	5	4	3	2	1
Δx (반복별 총 추가 비용)		x (100%)	$0.8x$ (80%)	$0.6x$ (60%)	$0.4x$ (40%)	$0.2x$ (20%)	0 (0%)
옵션 요구사항 $\Delta x/(x + \Delta x)$		50.0%	44.4%	37.5%	28.6%	16.6%	0%
요구 사항 변경에 따른 각 반복 시점의 추가 비용	반복1	Δx_1	0	-	-	-	-
	반복2	Δx_2	$x/15$ (6.7%)	0	-	-	-
	반복3	Δx_3	$2x/15$ (13.3%)	$2x/25$ (8%)	0	-	-
	반복4	Δx_4	$3x/15$ (20%)	$4x/25$ (16%)	$x/10$ (10%)	0	-
	반복5	Δx_5	$4x/15$ (26.7%)	$6x/25$ (24%)	$2x/10$ (20%)	$2x/15$ (13%)	0
	반복6	Δx_6	$5x/15$ (33.3%)	$8x/25$ (32%)	$3x/10$ (30%)	$4x/15$ (27%)	$x/5$ (20%)

발될 요구사항은 처음 고객이 제시한 요구사항의 2배가 됨을 의미한다. 프로젝트가 진행되면서 처음에 제기한 요구사항 만큼의 변경 (추가 또는 수정)이 발생할 수 있는가? Capers[10]는 새롭거나 예상되지 못한 요구사항들이 매월 총 요구사항의 1%의 비율로 추가로 발생한다고 제안하였다. 즉, 2년 동안 개발을 수행하는 프로젝트라면 요구사항 변경은 총 요구사항의 24%만이 발생한다. 즉, 100%의 요구사항 변경이 발생하려면 8년 이상의 개발기간을 가진 프로젝트라야 가능해지며 이러한 장기간 개발을 하는 프로젝트들이 거의 없는 실정이다. 이를 뒷받침하는 근거로 ISBSG Release 6[15]의 1989년부터 2000년까지 개발된 789개 프로젝트들 중 개발기간이 명시된 640개 프로젝트를 대상으로 개발기간을 분석한 결과는 (그림 2)와 같다. 그림에서 대부분의 프로젝트들이 2년 이내로 개발기간을 설정하고 있다. 즉, 대부분의 프로젝트들의 요구사항 변경은 처음에 제시한 요구사항의 24% 수준이다. 또한, (그림 1)에서 구현 (코딩과 시험)단계에서의 Δx 는 $0.15x$ ($x + \Delta x = 1.15$)이며, 이를 상세설계 단계부터 적용하면 Δx 는 $0.25x$ ($x + \Delta x = 1.25x$)로 2년 이내의 개발기간 설정시 요구사항 변경 비율과 대략 일치한다.

- (4) 요구사항을 구현하는데 소요되는 기본적인 비용을 x 라 한다. x 는 프로젝트를 개발하는 도입, 정련, 구축과 전이단계 모두에 소요되는 확정가 비용이다. RUP의 경우 도입-정련-구축-전이단계에 소요되는 비용은 5%-20% -65%-10%로 추천하고 있다.[16] 이에 따르면 구축단계에는 확정가 x 의 65%인 $0.65x$ 가 된다.
- (5) 기존 이론은 요구사항 변경은 구현 (구축)단계에서만 발생한다고 가정하고 있다. 그러나 반복개발 프로세스에서는 도입단계에서 요구사항인 유스케이스 목록이 작성되고, 정련단계에서 이들 유스케이스들이 상세히 기술되고 일부분이 구현되며, 구축단계에서 대부분



개발기간	프로젝트 수	비율
1년	438개	68.4%
2년	114개	17.8%
3년	30개	4.7%
4년	13개	2.0%
5년	2개	0.3%

(그림 2) 프로젝트의 개발기간 분포

의 유스케이스들이 구현된다. 따라서 정련단계에서도 기존에 상세히 기술된 유스케이스들이 변경되거나 새로운 유스케이스들의 추가가 발생할 수 있다. 결론적으로 요구사항 변경은 구축단계에서만 발생한다고 한정하지 않고 정련단계 까지 확장할 수 있다. 결국 구현단계란 구축단계만으로 한정시킬 수 있으므로 이후부터는 구현단계라는 용어를 사용하지 않기로 한다.

4. 확정가 프로젝트의 추가 소요비용과 옵션 요구사항

반복 프로세스에서 얼마만큼의 반복이 수행되는지 살펴보자. Kruchten[16, 17]는 RUP가 제시하는 반복 수에 대한 구체적인 지침으로 <표 3>을 제시하였다. () 안은 전형적인 프로젝트인 기준 반복 수 (1-2-2-1 = 6 반복)의 ±를 의미한

<표 3> 프로젝트 개발 특성과 복잡도에 따른 반복 수

총 반복수	단계별 반복수 할당				프로젝트 개발 특성
	I	E	C	T	
1	0	0	0	1	폭포수 프로세스
2	0	0	1	1	베타 버전후 정식버전 납품 축소수 프로세스 또는 단순 유지보수
3	0	1	1	1	반복 개발 경험이 전혀 없을 경우
4	1	1	1	1	단순 프로젝트
5	0	1	2	2	아키텍처가 이미 존재하고 요구사항이 안정되어 잘 이해될 경우
	0	1	1	3	제품이 안정, 견고한 아키텍처를 가지고 있고 점진적인 단계별 납품
6	1	2	2	1	전형적인 프로젝트
	0(-1)	2	2	2(+1)	타당성 조사가 불필요, 프로젝트 범위를 쉽게 결정할 수 있을 경우
	0(-1)	1(-1)	1(-1)	3(+2)	
7	1	2	3(+1)	1	
8	1	3(+1)	3(+1)	1	
	1	2	3(+1)	2(+1)	전형적인 RUP 생명주기
9	1	3(+1)	3(+1)	2(+1)	큰 규모 프로젝트
10	2(+1)	3(+1)	3(+1)	2(+1)	대형 프로젝트

다. 따라서 정련단계와 구축단계에서 수행되는 반복 수를 더하면 0, 1, 2, 3, 4, 5, 6의 경우가 발생하므로 최대 6개의 반복을 수행한다.

기존에 제안된 모델의 문제점을 해결하기 위해 요구사항 변경이 발생하는 동적으로 진화하는 비즈니스 시스템과 관련된 전형적인 프로젝트의 구현에 대해 다음 2가지 경우에 대해 고찰해본다.

- (1) 확정가 x 를 기준으로 할 때 요구사항 변경으로 25%의 비용이 추가로 발생하는 경우
- (2) 정련과 구축단계 비용인 $0.85x(=0.20x+0.65x)$ 를 기준으로 하였을 때 요구사항 변경으로 25%의 비용이 추가로 발생하는 경우

4.1 확정가 x 기준 요구사항 변경으로 25%의 비용이 추가로 발생하는 경우

프로젝트 개발에 소요되는 총 비용인 확정가 x 를 기준으로 하였을 경우 다음과 같은 가정에 기반하여 모델을 제시한다.

- (1) 각 반복에서 투입되는 비용은 거의 동일한 수준이다. 실제로 정련단계와 구축단계의 각 반복에서 투입되는 비용은 동일한 일양분포를 하지 않으나 모델의 단순성을 위해 동일하다고 가정한다.
- (2) 반복 수 n 에 대한 Δx 는 선형적으로 감소한다. 엄밀히 말하면 요구사항 변경으로 인해 발생하는 추가 소요 비용은 지수적으로 감소하지만 모델을 단순화시키기 위해 선형적으로 감소한다고 가정한다.
- (3) i 번째 반복이 시작되는 시점에서 이 반복에서 구체화되거나 실현되는 요구사항들은 확정되어 있다. ($\Delta x_i = 0$). 또한, 실현된 요구사항은 이후 시점에서 수정이 발생하지 않는다.
- (4) 요구사항 변경으로 발생하는 추가 비용 Δx 는 확정가 x 를 기준으로 산출한다. 즉, 첫 번째 반복에서 요구사항 변경으로 인해 추가로 소요되는 총 비용 Δx 는 $0.25x$ 이다. 요구사항 변경에 따라 추가되는 비용은 기존에 정련단계에서 상세화된 요구사항의 수정에 따른 소요 비용과 프로젝트에 대한 상세한 지식이 부족

하여 새로운 요구사항이 추가됨에 따른 비용을 더한 값으로 본다.

I_1 에서 I_n 까지 $\Delta x = \sum_{i=1}^n \Delta x_i$ 의 선형적 감소에 따라 $\Delta x = 0.25x$ 가 $n-1$ 번 감소하므로 각 반복에서 Δx 는 $0.25x/(n-1)$ 씩 감소한다. 즉 I_j 반복의 시작시점에서 Δx 는 $0.25x\{(j-1)/(n-1)\}$ 이 된다. Δx_i 의 선형 증가에 따라 I_j 반복 시작 시점에서 $\Delta x = 0.25x\{(n-j)/(n-1)\}$, $\Delta_{x_{j+1}} = 0.25x\{(n-j)/(n-1)\} / \sum_{i=j+1}^n (i-j)$ 가 된다.

정련단계와 구축단계에서 수행되는 최대 반복 수는 6이므로, 확정가가 x 이고, 6개의 반복($n=6$)으로 수행된다면 <표 4>와 같이 각 반복에서 요구사항 변경으로 인한 총 추가비용과 옵션으로 분류할 요구사항의 비율을 결정할 수 있다.

반복 1의 시작 시점에서 볼 때 반복 2에서 요구사항 변경으로 추가로 소요되는 비용은 확정가의 1.7%, 반복 2에서는 3.3%, 이후 반복들에서는 각각 5.0%, 6.7%, 8.3%로 증가한다. 따라서 확정가의 25% 비용이 요구사항 변경으로 추가로 소요된다. 첫 번째 반복이 시작되는 시점에서 옵션 요구사항의 비율은 $\Delta x / (x + \Delta x) = 25 / (100 + 25) = 20\%$ 가 되며, 2번째 반복이 시작되는 시점에서는 $20 / (100 + 20) = 16.7\%$ 로 감소한다. 이후의 각 반복 시작시점에서 13%, 9.1%, 4.8%와 0%로 각각 감소한다.

4.2 정련과 구축단계 비용 $0.85x$ 기준시 요구사항 변경으로 25%의 비용이 추가로 발생하는 경우

정련과 구축단계에 소요되는 비용을 기준으로 하였을 경우 다음과 같은 가정에 기반하여 모델을 제시한다.

- (1) 각 반복에서 투입되는 비용은 거의 동일한 수준이다.
- (2) n 에 대한 Δx 는 선형적으로 감소한다.
- (3) i 번째 반복이 시작되는 시점에서 이 반복에서 구현되는 요구사항들은 확정되어 있다고 가정한다. ($\Delta x_i = 0$). 요구사항이 실현되는 시점 이후에는 변경이 발생하지 않는다고 가정한다.
- (4) 요구사항 변경으로 발생하는 추가 비용 Δx 는 정련단계와 구축단계에서 소요되는 비용 $0.85x$ 를 기준으로

<표 4> 확정가 기준 요구사항 변경에 따른 반복별 개발비용 증가와 옵션 요구사항

구분	시작시점					
	반복 1	반복 2	반복 3	반복 4	반복 5	반복 6
남아있는 반복수	6	5	4	3	2	1
Δx (반복별 총 추가 비용)	$0.25x$ (25%)	$0.20x$ (20%)	$0.15x$ (15%)	$0.10x$ (10%)	$0.05x$ (5%)	0 (0%)
옵션 요구사항 $\Delta x / (x + \Delta x)$	20.0%	16.7%	13.0%	9.1%	4.8%	0%
요구 사항 변경에 따른 각 반복 시점의 추가 비용	반복1 Δx_1	0	-	-	-	-
	반복2 Δx_2	$0.25x/15$ (1.7%)	0	-	-	-
	반복3 Δx_3	$0.5x/15$ (3.3%)	$0.5x/25$ (2.0%)	0	-	-
	반복4 Δx_4	$0.75x/15$ (5.0%)	$1.0x/25$ (4.0%)	$0.25x/10$ (2.5%)	0	-
	반복5 Δx_5	$1.0x/15$ (6.7%)	$0.5x/25$ (6.0%)	$0.5x/10$ (5.0%)	$0.5x/15$ (3.3%)	0
	반복6 Δx_6	$1.25x/15$ (8.3%)	$2.0x/25$ (8.0%)	$0.75x/10$ (7.5%)	$1.0x/15$ (6.7%)	$0.25x/5$ (5.0%)

산출한다. 따라서 첫 번째 반복에서 요구사항 변경으로 인해 추가로 소요되는 총 비용 Δx 는 약 $0.20x$ ($0.85x$ 의 25%)이다.

확정가가 x 이고, 6개의 반복으로 구현한다면 <표 5>와 같이 각 반복에서 요구사항 변경으로 인한 총 추가비용과 옵션으로 분류할 요구사항의 비율을 결정할 수 있다.

반복 1의 시작 시점에서 볼 때, 반복 2에서 요구사항 변경으로 추가로 소요되는 비용은 1.3%, 반복 3에서는 2.7%, 이후 반복에서는 4.0%, 5.3%, 6.7%로 증가한다. 따라서 확정가의 20%가 요구사항 변경으로 추가로 소요된다.

첫 번째 반복이 시작되는 시점에서 옵션 요구사항의 비율은 $\Delta x / (0.85x + \Delta x) = 20 / (85 + 20) = 19.0\%$, 2번째 반복이 시작되는 시점에서는 15.8%, 이후의 각 반복 시작시점에서 12.4%, 8.6%, 4.5%와 0%로 각각 감소한다.

4.3 모델 평가

기존 모델과 제안된 모델의 반복별 추가비용과 옵션 요구사항의 비율을 요약 정리하면 <표 6>과 같다.

기존 제안된 모델은 확정가 x 범위 내에서는 총 요구사

항의 50%만을 개발할 수 있으며 요구사항 변경을 고려하여 개발을 완료하면 확정가의 2배의 비용으로 계약을 체결해야 함을 의미한다. 과연 고객이 제안한 요구사항의 50%만을 개발한다면 계약이 체결될 수 있는지 또는 성공한 프로젝트라고 할 수 있는지 의문시될 수 있다. 이에 반해 제안된 모델 중 확정가 x 를 기준으로 했을 경우, 총 요구사항의 80%를 개발할 수 있고, 확정가의 1.25배 비용으로 계약을 체결하면 중요하지 않다고 생각되는 요구사항들도 모두 개발할 수 있음을 의미한다. 또한, 정련과 구축단계 비용만을 기준으로 했을 경우에는 확정가 x 범위 내에서 총 요구사항의 81%를 개발 가능하고 확정가의 1.20배 비용으로 계약시 모든 요구사항들을 개발 가능하다.

제안된 모델의 특징을 기존 모델과 비교하면 <표 7>과 같다.

5. 결론 및 추후연구

확정가로 계약하여 소프트웨어 프로젝트를 개발할 경우, 개발과정에서 빈번히 발생하는 요구사항의 변경으로 인해 추가로 소요되는 비용으로 인해 확정가 범위 이내로 개발을

<표 5> 정련과 구축단계 비용 기준 요구사항 변경에 따른 반복별 개발비용 증가와 옵션 요구사항

구분	시작시점					
	반복 1	반복 2	반복 3	반복 4	반복 5	반복 6
남아있는 반복수	6	5	4	3	2	1
Δx (반복별 총 추가 비용)	0.20x (20%)	0.16x (16%)	0.12x (12%)	0.08x (8%)	0.04x (4%)	0 (0%)
옵션 요구사항 $\Delta x / (x + \Delta x)$	19.0%	15.8%	12.4%	8.6%	4.5%	0%
요구 사항 변경에 따른 각 반복 시점의 추가 비용	반복1 Δx_1	0	-	-	-	-
	반복2 Δx_2	0.2x/15 (1.3%)	0	-	-	-
	반복3 Δx_3	0.4x/15 (2.7%)	0.8x/25 (1.6%)	0	-	-
	반복4 Δx_4	0.6x/15 (4.0%)	0.4x/25 (3.2%)	0.2x/10 (2.0%)	0	-
	반복5 Δx_5	0.8x/15 (5.3%)	1.2x/25 (4.8%)	0.4x/10 (4.0%)	0.4x/15 (2.7%)	0
	반복6 Δx_6	1.0x/15 (6.7%)	1.6x/25 (6.4%)	0.6x/10 (6.0%)	0.8x/15 (5.3%)	0.2x/5 (4.0%)

<표 6> 반복별 추가비용과 옵션 요구사항 비교

반복 1 시작시점 기준	옵션 요구사항 비율	확정가 대비 요구사항 변경에 따른 추가 비용						
		계	반복 1	반복 2	반복 3	반복 4	반복 5	반복 6
Bhagwat 모델[7]	50.0%	100%	0%	6.7%	13.3%	20.0%	26.7%	33.3%
개발 확정가 기준	20.0%	25%	0%	1.7%	3.3%	5.0%	6.7%	8.3%
정련+구축단계 비용 기준	19.0%	20%	0%	1.3%	2.7%	4.0%	5.3%	6.7%

<표 7> 모델 비교

구분	Bhagwat 모델[7]	제안 모델	
		총 개발비용 (확정가)기준	정련+구축단계 비용 기준
확정가 대비 요구사항 변경에 따른 추가 소요 비용	1배	0.25배	0.20배
확정가로 개발 가능 요구사항 비율 : 개발 불가능 옵션비율	50%:50%	80%:20%	81%:19%
적용단계	불명확	명확화	
요구사항 변경 발생단계 취급	구축단계	정련단계+구축단계	

완료하는 것은 거의 불가능하다. 따라서, 확정가 범위 이내에서 성공적으로 프로젝트를 완료하기 위해서는 요구사항 변경이 얼마나 발생할 것인지를 알아야 하며, 이를 바탕으로 사전에 개발이 되지 않아도 되는 요구사항을 옵션으로 분류하여 관리하여야 한다. 이에 대해 Bhagwat[7]가 연구한 이론이 있다. 그러나 이 이론은 요구사항 변경이 발생하는 단계 적용의 모호함과 더불어 요구사항 변경의 비율의 오 적용 등 문제점을 내포하고 있다. 또한, 요구사항 변경이 구축단계에서만 발생한다고 한정하였으나 소프트웨어 개발비용, 요구사항 구현비용과 요구사항 변경에 소요되는 추가 비용을 모두 x 라고 가정하는 모순을 보였다.

본 논문은 소프트웨어 개발비용을 확정가 x , 요구사항 구현 비용은 구축단계에 소요되는 $0.65x$ (개발비용의 65%), 요구사항 변경에 소요되는 추가 비용을 $0.25x$ 로 가정하였다. 이에 기반하여 확정가를 기준으로 할 경우와 정련과 구축단계에 소요되는 비용을 기준으로 할 경우로 구분하여 요구사항 변경에 따른 추가 발생 비용과 개발이 되지 않아도 되는 옵션 요구사항의 비율을 계산하였다. 본 제안 이론은 요구사항 변경이 정련단계부터 발생한다고 확장하여 가정함으로써 보다 현실성을 반영하였으며, 확정가 대비 요구사항 변경에 따른 추가 소요 비용과 개발이 되지 않아도 되는 옵션 요구사항의 비율도 보다 현실성을 반영하는 결과를 얻을 수 있었다.

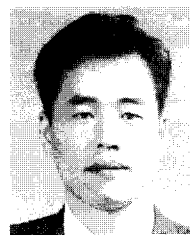
제안된 모델은 각 반복에서 투입되는 비용은 거의 동일하다고 가정하였으나 실제적으로는 정련단계와 구축단계에 투입되는 인력의 분포가 차이가 발생한다. 또한, 임의의 반복들에서 요구사항 변경으로 소요되는 추가 비용 Δx 는 선형적으로 감소한다고 가정하였으나 (그림 1)에서 보는 바와 같이 지수적으로 감소한다. 따라서 추후 이 분야에 대한 연구를 수행하여 보다 현실적으로 타당한 모델을 개발할 예정이다.

참 고 문 헌

[1] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol.10, No.1, pp.7-19, 1984.
 [2] B. Boehm, B. Clark, E. Horowitz, R. Modachy, R. Shelby, and C. Westland, "The COCOMO 2.0 Software Cost Estimation Model," USC Center for Software Engineering, 1995.
 [3] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Line of Code and Development Effort Prediction: A Software Science Validation," IEEE Trans. on Software Eng., Vol. SE-9, No.6, pp.639-648, 1983.
 [4] M. Bradley, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group (IFPUG), 1999.
 [5] C. Symons, "COSMIC-FFP Measurement Manual, Version 2.2 (The COSMIC Implementation Guide for ISO/IEC 19761: 2003)," Common Software Measurement International Consortium, 2003.
 [6] K. Ribu, "Estimating Object-oriented Software Projects with Use Cases," University of Oslo Department of Informatics, Master of Science Thesis, 2001.
 [7] A. Bhagwat, "Estimating Use-Case Driven Iterative Development for Fixed-Cost Project," The Rational Edge e-zine for the

Rational Community, http://www.therationaledge.com/content/oct_03/f_estimate_b.jsp. 2003. 10.

[8] J. Robertson and S. Robertson, "Requirements: Made to Measure," American Programmer, Vol.X, No.8, 1997.
 [9] R. A. Carter, A. I. Antón, A. Dagnino, and L. Williams, "Evolving Beyond Requirements Creep: A Risk-Based Evolutionary Prototyping Model," Fifth IEEE International Symposium on Requirements Engineering(RE '01), Toronto, Canada, 2001.
 [10] J. Capers, "Assessment and Control of Software Risks," Englewood Cliffs, NJ: Prentice Hall, Yourdon Pres, 1994.
 [11] B. Lieberman, "Project Scope Management: Effective Negotiating Change," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/nov_01/m_proScope Management_bl.html, 2001. 11.
 [12] Standish Group, "CHAOS: A Recipe for Success," 2001.
 [13] C. Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process," Prentice Hall, 2002.
 [14] K. Bittner and I. Spence, "Managing Iterative Software Development with Use Cases," The Rational Edge e-zine for the Rational Community http://www.therationaledge.com/content/mar_03/f_iterativeUse_kb.jsp, 2003. 3.
 [15] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
 [16] P. Kruchten, "Planning an Iterative Project," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/oct_02/f_iterative Planning_pk.jsp, 2002. 10.
 [17] P. Kruchten, "Going Over the Waterfall with the RUP," The Rational Edge e-zine for the Rational Community, http://www.therationaledge.com/content/sep_01/t_waterfall_pk.jsp. 2001. 9.



이 상 운

e-mail : sulee@gangnung.ac.kr

1983년 ~ 1987년 한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 경상대학교 컴퓨터과학과 (석사)

1998년 ~ 2001년 경상대학교 컴퓨터과학과 (박사)

1992년 ~ 2002년 국방품질관리소 항공전자장비 소프트웨어 품질보증 담당

2003년 강원도립대학 컴퓨터응용과 전임강사

2004년 ~ 2007년 2월 국립 원주대학 여성교양과 조교수

2007년 3월 ~ 현재 강릉대학교 컴퓨터공학부 멀티미디어정보공학 전공 조교수

관심분야 : 소프트웨어 프로젝트 관리, 소프트웨어 개발 방법론, 소프트웨어 척도 (소프트웨어 규모, 개발노력, 개발기간, 팀 규모), 분석과 설계 방법론, 소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성, 신경망, 뉴로-퍼지, 알고리즘