

## 유스케이스 기반의 진척 관리 기법

이광찬<sup>0</sup> 백종현  
대우정보시스템 주식회사  
{ebiz, baegih}@disc.co.kr

### Use case - Driven Project Management Technique

KwangChan Lee<sup>0</sup>  
Daewoo Information Systems Co., Ltd.

#### 요 약

최근의 정보시스템 개발은 컴포넌트 기반 소프트웨어 개발로서 반복적인 소프트웨어 개발 공정을 따른다. 반복(Iteration)이란 정보시스템 개발 생명주기(SDLC)의 요구사항, 분석, 설계, 개발, 테스트 과정을 기존의 Waterfall방식으로 개발하지 않고, 각 단계들을 반복적으로 수행하여 시스템을 개발하는 방식이다. 따라서 기존의 단계, 액티비티 그리고 하위의 TASK에 해당하는 산출물 완성 유무를 가지고 프로세스 측면의 공정 진척을 적용하게 되면 프로젝트 현 시점의 공정 상태를 명확하게 파악하는 것이 불가능하다.

본 연구에서는 최근의 컴포넌트 기반 소프트웨어 개발에서 요구되어지는 유스케이스 기반의 프로덕트 중심의 공정 진척 관리에 필요한 기법들을 연구하였다. 본 논문에서는 유스케이스가 분석, 설계, 개발, 테스트, 배포되는 과정에 있어서 산출물 기반 진척관리에 대한 기법을 제안함으로써 반복적인 소프트웨어 개발에서의 진척 관리 체계화를 증대할 수 있는 기법을 제시한다.

#### 1. 서론

기존의 정보시스템 개발은 Waterfall방식으로써 고객에 대한 요구사항을 파악하고 이를 바탕으로 하여 요구사항, 분석, 설계, 개발, 테스트의 과정을 포함하는 생명주기를 이루었다. 따라서 프로젝트에서의 오류에 영향을 미칠 수 있는 위험요소가 각 단계가 끝나고 나서 발생된다면 프로젝트에 대하여 막대한 비용이나 일정에 대한 연기를 초래하였다.

정보시스템 구축 프로젝트의 수행에 있어서 위험요소는 지금까지 모든 프로젝트에 내재되어 있었다. 그럼에도 불구하고, 위험요소가 지닌 불확실성(발생가능성, 영향도, 대안의 적합성 등)으로 인하여, 아직까지 구체적인 틀로써 위험요소들을 추적하고 관리하는 것은 매우 어려운 과제로 남아있다. 더욱이 정보시스템의 개발이 단순한 소프트웨어 개발에서, 이른바 통합이라 일컬어지는 사용자 중심의 통합 솔루션 제공을 위한 개념으로 확대, 발전함으로써, 프로젝트 수행의 규모와 복잡성이 매우 커졌다. 다시 말해 대규모의 개발조직이 장기간에 걸쳐 다양한 구성요소들을 개발, 통합하게 되면서 각각의 위험요소들이 기하급수적으로 증가하게 되었고, 이에 따라 프로젝트 수행 시 효과적인 위험관리가 프로젝트 성공의 주요 핵심요인으로 자리하게 되었다.

따라서 폭포수 모델에 대한 대안으로 점진적 개발과 반복 개발 프로세스를 도입함으로써 프로젝트 초기에 프로젝트 수행 동안 나타날 수 있는 폭포수 모델의 여러 위험들

을 개선할 수 있다.

DSDM/가능쇠 CBD 개발방법론은 점진적이고 반복적인 개발 프로세스를 적용한 컴포넌트 기반 개발 방법론으로써 사용자의 요구사항을 소프트웨어로 만들어 가는 활동들을 정의하고 있다. DSDM/가능쇠 CBD는 단계(phase), 액티비티, TASK의 절차와 기법, 사례, 산출물, 프로젝트의 산출물을 관측하고 측정하는 다양한 기준을 제공하고 있고, 또한 프로젝트의 관리 프로세스 영역에서 가장 중요한 일정에 대한 진척관리를 제공한다.

본 논문에서는 컴포넌트 기반 소프트웨어 개발 프로젝트 수행을 지원하는 프로젝트의 진척 관리에 필요한 기법을 연구하였고 실제 이 연구를 통하여 반복적인 소프트웨어 개발에서의 계획 수립 및 진척 관리의 체계화를 증대할 수 있는 기법을 제시한다.

#### 2. 기능 점수(FP) 크기 산정을 위한 프로세스

이장에서는 기능 점수의 크기를 견적하는데 사용되는 전체 프로세스를 다음과 같이 8단계로 요약하였다.

첫째, 프로젝트(신규개발, 기능개선, 재구축, 플랫폼교체, 패키지적용, 유지보수)에 따라 기능 점수 값의 유형을 결정한다.

둘째, 현재의 기능을 포함한 계산 범위와 경계를 확인한다.

셋째, 이벤트에 의한 유스케이스 리스트를 도출한다.

넷째, 각 유스케이스별 모든 자료 기능(내부 논리 파일과

외부 인터페이스 파일)과 복잡도를 확인한다.  
 다섯째, 각 유스케이스별 트랜잭션 기능(외부 입력, 외부 출력, 외부 조회)과 복잡도를 확인한다.  
 여섯째, 미조정 기능 점수 값을 결정한다.  
 일곱째, 14개의 일반 시스템 특성(GSC)에 근거한 값 조정 인자를 결정한다.  
 여덟째, 조정 기능 점수 값을 계산한다.

3. 공정 진척 관리 절차

3.1 DSDM/가능최 CBD개발방법론

본 논문에서 제안하는 CBD 프로젝트의 진척 관리 기법은 DIS의 CBD개발방법론을 적용한 프로젝트를 대상으로 한다. CBD개발방법론은 절차로서 단계(Phase), 액티비티(Activity), 타스크(Task)와 산출물(Output)로 해당 액티비티와 타스크에 각각 대응되는 프로덕트와 컴포넌트를 체계적으로 정리한 방법론이다.

기본적으로 정보시스템 개발 생명주기(SDLC)의 분석, 설계, 개발, 테스트 과정을 기존의 Waterfall방식으로 개발하지 않고, 아래의 그림과 같이 각 단계들을 반복적으로 수행하여 시스템을 개발하는 방식을 따른다. 따라서 소프트웨어의 개발은 여러 번의 반복(Iteration)을 거치며 각각의 반복은 분석, 설계, 개발, 테스트 과정을 포함하고 있어 자체로서도 하나의 개발주기를 이룬다. 이러한 반복적인 개발 방법에서는 반복마다 실행 가능한 릴리즈가 산출되고 이는 반복이 거듭될수록 향상되어 결국 최종 시스템으로 발전된다.

프로젝트 개발 중 반복은 여러 번 수행될 수 있고 각각의 반복마다 주요점은 조금씩 달라진다. 초기의 반복에서는 분석에 중점을 두며, 다음 번 반복에서는 점차 분석&설계, 구현쪽에 중점을 두게 된다. 그리고 말기의 반복에서는 테스트와 배포에 중점을 두게 된다.

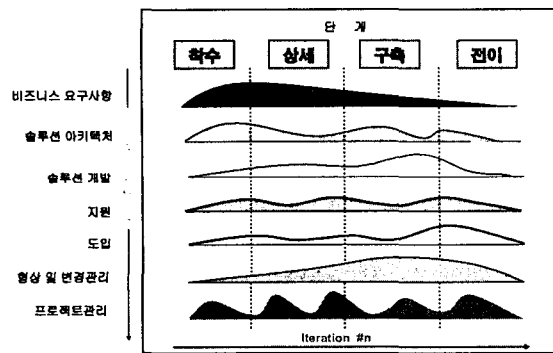


그림 1. DSDM/가능최 CBD 방법론 구조

각 단계의 목적은 아래와 같이 요약된다.  
 프로젝트 완료기준을 적용하여 완료기준(분석, 설계, 개발, 테스트, 배포)별 산출물을 완성하게 되면 유스케이스의 계획 기능점수를 실적 기능점수로 한다. 따라서 본 예제에서의 유스케이스의 실적 기능 점수는 각각 24, 25, 40점으로 하여 총합은 89점이고, 전체 공정 진척율은 89%(실적/계획)를 나타낸다.

첫째, 착수 단계는 개발생명주기 상에서 모든 이해관계자 사이의 의견 일치를 달성하는 것으로서 프로젝트의 범위, 비용 그리고 일정에 대한 동의를 획득한다.

둘째, 상세 단계는 설계 및 개발에 대한 안정성을 제공하기 위하여 시스템의 아키텍처 구조를 베이스라인 하는 것이다.

셋째, 구축 단계는 베이스라인 아키텍처에 기반한 시스템의 개발을 완성하고 남아있는 요구사항을 명백히 하는 것이다.

넷째, 전이 단계는 소프트웨어 및 최종 시스템이 최종사용자에게 유용하도록 보증하는 것이다.

3.2 진척관리 개념 모델

착수 단계에서 프로젝트 범위에 해당하는 모든 유스케이스를 도출하고 이에 대한 기능점수를 파악한다.

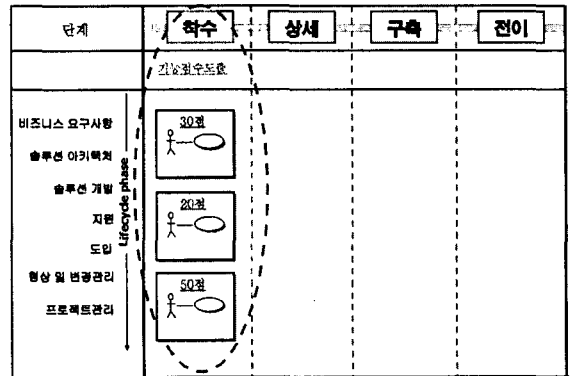


그림 2. 진척관리 모델1

본 진척관리 모델1에서는 착수 단계에서 초기의 유스케이스 3개를 도출하였고, 이에 대한 계획 기능점수는 각각 30, 20, 50점으로 하여 총합은 100점이다.

아래의 그림 3에서는 착수 단계의 유스케이스들이 DSDM/가능최 CBD 방법론의 절차를 따라 개발되면서 프로젝트의 완료기준을 적용한 공정진척율을 보여주고 있다.

프로젝트 완료기준	단계 : 착수단계					공정 진척율
	분석(30%)	설계(30%)	개발(40%)	테스트(10%)	배포(10%)	
UseCase 30점 [User Icon]			24점			80%
20점 [User Icon]				20점		100%
50점 [User Icon]				45점		90%
착수 단계의 프로젝트 공정 진척율 = 89/100						89%

그림 3. 진척관리 모델2

DIS의 프로젝트 완료 기준별 산출물 표는 아래와 같다.

프로젝트 완료기준	산출물 명
분석	유스케이스모델
	클래스모델
	데이터모델
설계	컴포넌트명세서
	UI설계서
	컴포넌트설계서
	데이터베이스명세서
개발	프로그램
	데이터베이스명세
테스트	통합테스트보고서
배포	인수테스트보고서
	매뉴얼

표 1. DSDM/가능쇠 프로젝트 완료기준

#### 4. 결론

이상에서 DSDM/가능쇠 CBD 방법론에서 적용되고 있는 진척 관리 방법을 살펴보았다. 실제 이 연구는 점진적이고 반복적인 개발 기법을 적용하고 있는 CBD 프로젝트의 개발 관리의 어려움을 해결하고자 하였다. 본 논문에서는 프로젝트 관리 프로세스 영역에서 가장 중요시 되고 있는 일정을 관리하기 위하여 유스케이스가 분석, 설계, 개발, 테스트, 배포 되는 과정에서 기능점수 단위로 진척 관리를 하자는 것이다.

이를 위해 유스케이스 단위로의 진척 관리 모델을 제안하였으며, 단계별 산출물에 대한 프로젝트 완료 기준을 설계하였다. 현재 이 기법을 통해 대형 SI 프로젝트의 진척 관리가 이루어지고 있으며, 향후 중, 소규모의 프로젝트에 따른 진척도 관리에 반영할 수 있도록 모델을 좀더 체계적이고 정형화 시킬 필요가 있다.

#### 5. 참고문헌

- [1] Ivar Jacobson, et al, The Unified Software Development, Addison-Wesley, 1999
- [2] Atkinson, C., Component-Based Product Line Engineering with UML, Addison Wesley, 2002.
- [3] Walker Royce, Software Project Management, Addison-Wesley,
- [4] Philippe Kruchten, "The Rational Unified Process An Introduction", 2nd Edition, Addison-Wesley, 2000
- [5] David Garmus and David Herron, "Measuring Software Process: A Practical Guide to Functional Measurements", Englewood Cliffs, NJ: Prentice-Hall, 1995

NJ: Prentice-Hall, 1995

- [6] Duncan, William R, "A Guide to the Project Management Body of Knowledge", PMI Standards Committee, 1996
- [7] Richard Murch, Project Management: Best Practices for IT Professional, Prentice Hall PTR, 2000