

애자일 기법을 접목한 대형 프로젝트 기반 유스케이스 모델링

남동균, 이종왕, 한정헌
SK씨앤씨 IT공학센터 품질혁신팀

Use case modeling with agile practices for Enterprise

Nam Dongkyun, Lee Jongwang, Han Jeongheon

SK C&C

E-mail : ndk0706@skcc.com, bellking@skcc.com, scant@skcc.com

요 약

많은 SI프로젝트에서 유스케이스 모델링 기법을 사용하여 요구 분석을 수행하여 왔지만 모델링 기준과 가이드의 상이함으로 인해 프로젝트 현장에서 많은 문제점을 야기하고 있다. 본 논문에서는 애자일 방법론의 장점을 접목하여 애자일한 유스케이스 시나리오 작성법 및 사용자스토리를 개념을 차용한 두 단계 우선순위 부여를 통하여 요구 분석의 효율성과 생산성 제고방안을 제안한다.

I. 서론

SI 프로젝트에서 유스케이스 모델링을 사용하여 요구사항 분석하여 왔지만, 프로젝트나 회사별로 작성 방법과 수준이 상이하고 일부는 지나치게 상세하게 모델링(Big Requirements Up Front)함으로 인해 일정 지연과 재작업을 초래한다.[1]

이에 본 논문에서는 애자일한 유스케이스 시나리오 작성법을 통하여 유스케이스 시나리오 작성의 효율성을 제안하고 유스케이스에서 사용자스토리를 식별하는 방안과 두단계 우선순위를 통한 진척 관리방안을 제안한다.

II. 본론

1. 현황 분석

1.1. 현행 유스케이스 모델링 분석

유스케이스란 사용자 관점에서 시스템의 기능을 표현한 것으로써 시스템 개발 범위를 명확하게 하기 위해 프로젝트 초기에 사용자의 기능요구사항에서 도출한다. 또한, 유스케이스는 프로젝트의 이해관계자간의 주요 의사전달의 도구로 사용되므로, 모든 이해관계자가 시스템에 대한 이해를 같이할 수 있는 수준으로 도출 및 명세를 하여야 한다.

하지만, 유스케이스 크기(Granularity)에 대한 혼란과 이견으로 프로젝트 초기부터 재작업(Rework) 가능성과 향후 요구사항의 변경 위험을 내포하고 진행되는 경우가 많다. 일반적으로 <표1>에서 보는 바와 같이 유스케이스 크기가 작으면 유스케이스 개수가 많아져 프로젝트 관리 효율성과 모델의

변경 용이성이 저하되지만 각 유스케이스별 내부 시나리오 개수가 적어지고 유스케이스의 기능을 명확하게 기술되는 장점이 있다. 반대로 유스케이스 크기가 크면 각 유스케이스의 명칭만으로 그 내부 기능을 표현함에 한계가 있어 이해관계자가 오해할 가능성이 커지고, 유스케이스별 내부 시나리오 개수가 많아진다. 또한, 유스케이스의 개수가 적어 프로젝트 관리 효율성과 모델의 변경 용이성이 제고되는 장점이 있다.

구분	유스케이스 크기	
	小	大
유스케이스 수	多	少
유스케이스 내 시나리오 수	少	多
유스케이스의 기능 명확성	高	低

<표1> 유스케이스 크기와 명확성간의 상관관계

1.2. 애자일 기법 특장점 분석

유저스토리와 유스케이스의 가장 명확한 차이점은 범위(Scope)이다. 둘 다 비즈니스 가치를 제공하지만 사용자 스토리는 스케줄링에 사용하기 위해 주어진 기한 내에 개발이 완료되어야 한다는 제약사항으로 인해 범위측면에서 더 작다.[3,4]

유저스토리와 유스케이스의 또 다른 차이점은 완성도에 있다. Grenning은 사용자 스토리는 유스케이스의 주요 성공 시나리오(Main success scenario)에 해당하고, 사용자 스토리의 테스트케이스는 유스케이스의 확장(Extensions)에 해당한다고 한다.[4] 그리고, 유스케이스는 고객과 사업팀과의 합의를 문서화 할 목적으로 작성되지만, 스토리는 릴리즈와 이터레이션 계획수립을 용이하게 하고 사용자의 세부 니즈에 대한 커뮤니케이션 도구로 이용된다. [4]

이와 같이 특장점 분석을 기반으로 다음 절부터 애자일 기법을 접목한 유스케이스 모델링 방안들을 제안한다.

2. 애자일 유스케이스 시나리오 방안

유스케이스 시나리오 템플릿은 일반적으로 유사하나 작성 내용은 사람마다 달라 요구사항 분석에 효과적으로 대응하지 못하고 있다. 다음 <표3>에

서 유스케이스 시나리오 내용을 비교하였다.[4]

Good Use Case	Bad Use Cases
화면(GUI) 중심의 명세가 아님	화면(GUI) 중심의 명세
상세한 데이터 속성을 작성하지 않음	상세한 데이터 속성과약하여 작성
3~9스텝의 성공 시나리오	여러 페이지에 걸친 시나리오
읽기 쉬움	읽기 어려움
사용자가 친숙한 단어들로 작성	개발자가 이해하는 구현중심의 작성

<표2> 시나리오 비교

특히 국내 SI 프로젝트에서 심각하게 벌어지는 오류는 아래 <표3>과 같이 정의 할 수 있다.

구분	개선 대상	개선 포인트
1	화면 묘사 및 화면 흐름 중심의 명세서 작성	사용자가 요구한 기능 도출에 중점을 둔 명세
2	상세한 데이터 속성을 분석하여 기입	대략적인 속성만을 작성(분석, 설계 단계에 상세 속성 도출)
3	기본흐름, 대안흐름, 예외흐름 간의 모호한 정의로 작성시 혼란 가중 및 일관성 결여	3~9단계의 기본흐름과 그 외 경우는 서브흐름으로 전체를 표현
4	개발자의 시각으로 작성되어 사용자가 이해하기 어려움	해당 업무영역에 통용되는 단어들 사용하여 명세 작성
5	반복(Iteration) 여부와 상관없이 전체 유스케이스를 명세	해당 반복(Iteration) 때에 유스케이스를 작성함
6	중요도의 유명 무실화	중요도대신 우선순위(Priority)를 표시

<표3> 시나리오 개선 포인트 도출

구분1과 2에서 언급한 개선포인트는 유스케이스를 통한 요구사항 분석 시점은 프로젝트 초반이기 때문에 정확한 화면 레이아웃이나 입출력 혹은 연계 데이터 속성을 파악하는 것은 불가능하다. 또한, 요구사항도 아직 불완전하여 변경 가능성이 높은 만큼 상세한 정의는 재작업으로 이어질 공산이 크다. 프로젝트 초기의 요구 정의 단계는 무엇보다도 너비 우선 접근법(Breadth first approach)를 통하여 변경에 대한 유연성을 확보하는 것이 중요하다.[5] 구분3과 4에서의 개선 포인트는 성공 시나리오(기본흐름)를 사용자 스토리 작성처럼 사용자의 용어를 사용하여 간결하게 3~9 스텝 정도로 작성하고 각 스텝에서 발생하는 조건들을 대안과 예외에 상

관없이 서브흐름에 표현한다. 이를 통하여 사용자가 읽기 쉬운 구조로 표현되며 전체적으로 시나리오 작성에 일관성을 유지할 수 있다.

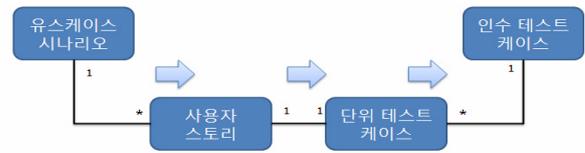
구분 5에서는 린 사고(Lean thinking)의 결정미루기를 도입하여 반복(Iteration)이 있는 프로젝트의 경우 해당 반복 시에 유스케이스 시나리오작업을 수행하여 재작업을 최소화한다.[5]

마지막 구분6에서는 유스케이스 명세서 양식 중 중요도 항목이 존재하고 있지만 사실상 유명무실하고 이 시점에 상(Critical), 중(Important), 하(Useful)를 파악하는 활동보다는 고객으로 하여금 개발 우선순위를 선택하게 하는 것이 더 효율적이다. 이를 통하여 고객이 선택한 기능을 우선적으로 고객에게 제공함으로 인해 보다 가치있는 활동이 된다.[6]

3. 유스케이스 시나리오와 사용자 스토리

3.1. 연동방안

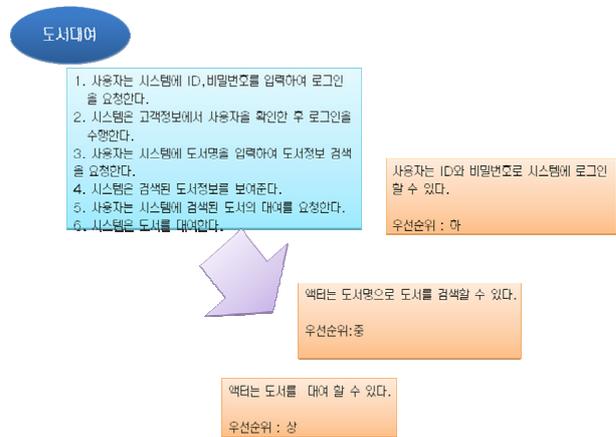
유스케이스 주도 개발프로세스에서 유스케이스만을 기준으로 우선순위를 부여하고 이를 구현의 진척단위를 할 경우 효율적이지 않다. 유스케이스는 의미있는 하나의 업무 흐름 단위로 식별되기 때문에 다수의 단위 기능을 포함하고 있는 경우가 많다. 이때 유스케이스 내의 중요한 다수 기능을 구현했다더라도 전체 100%가 완료되지 않았다면 진척율에 신뢰성 문제가 발생한다. 또한 반복적 개발 프로세스 활용 시, 우선순위에 의한 핵심기능(보통 유스케이스보다 크기가 작음)을 초기에 구현하는 것이 필수적이기 때문에 이에 따른 효율적인 관리가 필요하다. 본 제안에서는 이를 위해 유스케이스 시나리오를 기반으로 세부 기능위주의 사용자 스토리를 식별하고 유스케이스 및 사용자 스토리별 단계적 우선순위를 부여하는 방안을 제안한다. 또한 이러한 방안은 유스케이스 시나리오인 인수 테스트 케이스로, 각 사용자 스토리는 단위 테스트 케이스로 발전하게 함으로써 단계적 테스트의 통합을 가능케 한다.



<그림1> 제안 개념도

3.2 사용자 스토리 식별방안

유스케이스 시나리오는 액터가 시스템과 상호 작용하는 흐름을 시간의 흐름에 따라 묘사한다. 따라서 액터의 요청에 의한 시스템의 반응을 한 쌍으로 식별하여 사용자 스토리로 구분한다. 그 식별의 예는 <그림2>와 같다.



<그림2> 사용자 스토리 식별 예

‘도서 대여’라는 유스케이스 시나리오에서 3개의 사용자 스토리가 식별된 것을 볼 수 있다.

3.2 두 단계 우선순위 부여 방안

두 단계 우선순위 부여 방안은 유스케이스 및 유스케이스에서 식별된 사용자 스토리에 각각의 우선순위를 단계적으로 부여하는 방안이다. 앞서 언급한 바와 같이 유스케이스만의 우선순위 부여 방식은 그 효율성이 낮다. 따라서, 본 제안에서는 먼저 <표4>에서와 같이 우선순위의 기준을 명시하고,

분류	의미	점수
Must-do	필수	3
Should-do	해야 됨	2
Could-do	가능 함	1

<표4> 우선 순위 부여기준[7]

<표5>에서는 유스케이스별, 사용자 스토리별 우선 순위 부여의 예를 예시하고 있으며 유스케이스 대 사용자 스토리의 가중치 부여를 통하여 결과를 도출하였다.

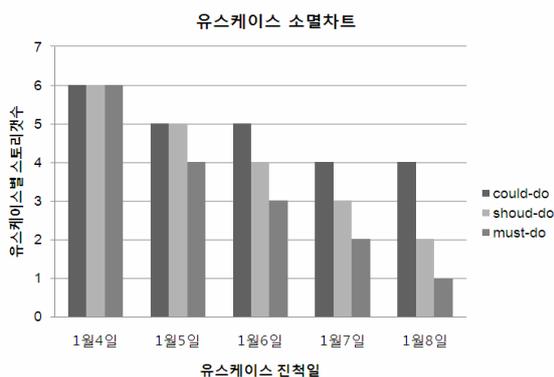
이름	사용자 스토리명	우선순위(가중치)		결과
		유스케이스 (0.6)	사용자 스토리 (0.4)	
A유스케이스	Aa사용자스토리	3(1.8)	3(1.2)	3
	Ab사용자스토리	3(1.8)	2(0.8)	2.6
	Ac사용자스토리	3(1.8)	1(0.4)	2.2
B유스케이스	Ba사용자스토리	2(1.2)	3(1.2)	2.4
	Bb사용자스토리	2(1.2)	2(0.8)	2.0

<표5> 우선순위 부여 예시

이러한 두 단계 우선순위 부여는 유스케이스 간의 우선순위를 파악은 물론 To-be 시스템의 핵심기능을 명확히 식별하게 한다. 또한, 유스케이스에 식별된 사용자 스토리는 단위 테스트 케이스의 작성의 기초가 되고, 유스케이스 시나리오는 인수테스트 케이스 작성의 명확한 기반이 된다.

3.3 진척도 측정 방안

두 단계 우선순위 부여는 진척도를 체크할 때 활용함으로 가치가 있다. 다음은 유스케이스별 사용자 스토리의 진척을 소멸차트(Burn-Down Chart)로 표현한 예이다.



<그림3> 우선순위별 소멸차트

<그림3>에서 보듯이 우선순위가 낮은 스토리인 경우 그 진척도가 낮지만 우선순위가 높은 스토리인 경우 그 진척도가 높음을 확인할 수 있다. 이

러한 차트는 우선순위에 입각하여 핵심 업무의 진척도를 차별적으로 판단 가능가능하며 유스케이스 단위 진척관리의 단점을 보완해 준다.

III. 결론

애자일 기법을 접목한 유스케이스 모델링을 통하여 유스케이스 시나리오 작성의 일관성과 생산성을 향상할 수 있으며 유스케이스를 사용자 스토리로 분해함으로써 명시적인 프로젝트 진척관리와 유스케이스 크기 식별에 도움을 줄 수 있다.

[참고문헌]

- [1] Scott Ambler, <http://www.agilemodeling.com/essays/examiningBRUF>
- [2] Alistair Cockburn(2004), Agileusecases1dy.ppt, <http://alistair.cockburn.us>
- [3] 김창준 외(역)(2006), 익스트림 프로그래밍, 인사이드
- [4] Mike Cohn(2004), Advantages Of User Stories for Requirements, InformIT Network
- [5] 김정민 외(역)(2007), 린 소프트웨어 개발, pp 101~110, 인사이드
- [6] 한주영 외(역)(2006), 사용자스토리, 인사이드
- [7] 에드워드 요든, 죽음의 행진-문제프로젝트에서 살아남는 법, pp. 207, 소동