

유즈케이스 모델링을 위한 유저 스토리 분석 방법에 관한 연구

채홍일^o 박영수 장덕철

광운대학교 컴퓨터과학과

{chaehongil^o, yspark, dcjang}@kw.ac.kr

Research on User story Analyzing Method for Use case Modeling

Hong-il Chae^o, Young-soo Park, Duk-chul Jang

Graduate School of Computer Science, Kwangwoon University

요 약

소프트웨어 생명주기 동안 변화하는 요구사항을 관리하기 위하여 다양한 기법들이 등장하고 있다. XP(eXtreme Programming) 기법에서 사용되는 유저 스토리는 고객이 시스템 관련 표현들에 대한 부담을 가지지 않고 자신의 요구사항을 표현할 수 있다는 장점을 가지고 있지만 그런 반면 체계를 가진 정형화된 문서가 아니고 일상적인 용어로 기술된다는 점 때문에 XP 기법에 대한 이해가 없는 타 부서나 개발 관계자들과의 요구사항에 관한 문서 교환 시 문제가 발생한다. 본 고에서는 이 문제를 해결하기 위한 방안으로 유저 스토리로부터 유즈케이스를 모델링하는 방법을 제안하였으며, 제안된 방법을 검증하기 위하여 XP 기법을 사용하여 진행된 DVD 타이틀 대여 웹 사이트 개발 프로젝트에 적용한 사례를 분석하였다.

1. 서론

소프트웨어 개발에 있어 요구 사항 단계에서 발생한 오류는 소프트웨어 프로젝트에서 발견된 모든 문제점들 중 40~60%의 원인이 되고 있으며, 후반부에 발생하는 요구 사항의 오류를 수정하는 비용은 개발 초기 단계인 요구공학 단계에서 오류를 검출하여 수정하는 비용의 200배가 소요되며, 소프트웨어 프로젝트 예산의 40~80%가 초기 결함을 수정하는 데 쓰인다고 한다[1].

일반적으로 프로젝트는 고객의 의뢰로 시작되며, 고객과 개발팀이 함께 무엇을 개발할 것인가를 정의하고, 개발팀이 그 무엇에 해당하는 것을 개발해나가는 것이 소프트웨어 개발인데 많은 조직들은 이 필수적인 활동에 대해서 비효과적인 방법을 택하고 있다. 유럽에서 행해진 조사 결과에 따르면 인식된 소프트웨어 문제점 중 50% 정도는 요구사항 명세에, 그리고 나머지의 50%정도는 요구사항 관리에 있었다[2].

근래에 주목 받고 있는 여러 개발 방법론들은 변경이나 유지, 보수에 보다 유연하도록 발전하고 있는데 이는 사용자 요구사항의 변경이 어느 정도 당연한 것이라는 인식이 개발팀들에게 받아들여지고 있기 때문이다. 애자일 동맹 선언[3]을 따르는 SCRUM[4], Crystal[5], FDD(Feature Driven Development)[6], ADP(Adaptive Software Development)[7], XP(eXtreme Programming)[8] 등의 수많은 개발 기법들은 기본적으로 요구사항이 불완전하여 개발기간

동안에도 자주 변경되는 것에 보다 유연하게 대처하는 것을 그 목적으로 하고 있다.

최근 중소규모 프로젝트에서 각광받는 XP 기법에서는 사용자의 요구사항을 기술하는데 유저 스토리(user story)를 사용한다. 유저 스토리는 고객이 유즈케이스, UML, IEEE 830-1998 SRS 등의 기능 명세 기법에 대한 지식을 가지고 있지 않을 것이라는 가정에 기반하여 일상에서 사용하는 자연어로 어떤 기능이나 모듈에 관하여 기술하는 기법이다[9]. 그러나 일반적으로 널리 사용되고 있는 유즈케이스에 비해 체계적인 문서화에 취약하다는 단점이 있다. 유저 스토리는 너무나도 느슨한 형태로 기록되기 때문에 유저 스토리 카드나 유저 스토리를 관리하는 위키(Wiki) 사이트를 매개로 하여 업무상의 전문적인 의사소통을 하기에는 문제가 있다. 본 고에서는 유저 스토리로부터 UML로 표기되는 유즈케이스 다이어그램을 도출하는 방법을 제안하여 이러한 문제를 해결하고자 한다.

2. 관련연구

수많은 요구사항 명세 기법이 있지만 일반적으로 쓰이는 것들 중 몇 가지만 간략히 살펴보면 IEEE 830-1998 SRS, 레쇼널 통합 공정에 사용되는 유즈케이스 그리고, XP 기법에서 사용되는 유저 스토리가 있다.

2.1 IEEE 830-1998 SRS (Software Requirements Specification)

IEEE 830-1998 SRS는 소프트웨어 요구사항 명세에 관한 표준인데 이 표준에 의해 파생된 템플릿은 크게 소개와 전체 설명, 시스템 특징, 외부 인터페이스 요구사항, 기능 이외의 다른 요구사항, 다른 요구사항, 부록A: 용어집, 부록B: 분석 모델, 부록C: 문제 목록으로 구성되며 각각은 다시 여러 세부 항목들로 구성된다.

2.2 유즈케이스 (Use Case)

유즈케이스는 시스템이 행하는 일련의 작업들과 그 결과에 대해서 기술하며, 현재 고객의 요구사항을 정의하는데 널리 쓰이고 있다. 래쇼널 통합 공정 (Rational Unified Process: RUP)에서는 요구사항을 유즈케이스를 통해 수집한다. 유즈케이스는 유저 스토리보다 훨씬 체계적인데 행위자(Actor), 설명, 사전 조건, 사후 조건, 사용 빈도, 작동 순서, 예외, 포함 등을 그 요소로 가진다

2.3 유저 스토리 (User Story)

유저 스토리는 사용 실례 같은 것과 비슷하며, 고객에 의해서 비전문적인 용어로 쓰여진다. 보통의 유저 스토리는 3~4 문장 정도의 길이로 이루어진다. 개발자는 유저 스토리를 보고 해당되는 개발에 필요한 시간을 추정한다. 좀더 자세한 예측을 위해 개발자들이 스파이킹(spiking)을 하는 경우도 있다.

2.4 IEEE 830 SRS vs. 유즈케이스 vs. 유저 스토리

먼저 IEEE 830 표준을 따르는 템플릿에 의해 작성된 SRS는 읽고 작성하는데 시간이 많이 소요하게 되고 딱딱한 형식으로 기술되어 읽기에 다소 지루하여 읽는 이가 일부를 건너뛸 수도 있다.

유즈케이스는 표준을 따르는 SRS 문서에 비해서는 덜 상세하지만 유저 스토리보다는 더 구체적이고 상세하다. 유즈케이스의 최대 단점은 고객이 개발자들의 언어로 그들의 요구사항을 표현해야 한다는 점이다. 그러나 UML이라는 모델링 언어로 모델링하여 시각적인 다이어그램으로 표현이 가능하다는 장점이 있다.

유저 스토리는 위의 두 방법에 비해 기술적으로 덜 상세하며 무엇보다 고객이 고객 자신의 언어로 요구사항을 설명하도록 해준다. 고객이 개발자가 원하는 만큼의 수준으로 자신의 요구사항을 설명하기에는 무리가 따른다. 비전문가가 UML이라는 모델링 언어를 사용하여 시스템의 관점에서 각종 연산을 전문가에게 설명한다는 것은 거의 불가능한 일이다. [표 1]은 유저 스토리의 예인데 일반 사람들의 표현이라는 것을 한 눈에 알 수 있다.

...
 * DVD 타이틀 검색: 고객이 자신이 원하는 DVD 타이틀이 관리 데이터베이스의 보유 목록에 있는지, 그리고 있다면 대어 가능한지를 검색한다. 한글 또는 영문 제목, 장르, 국가, 출시일, 감독, 출연배우를 기준으로 검색할 수 있다.
 ...

표 1 XP 기법에서 사용되는 유저 스토리의 예

물론 위와 같은 스토리는 복잡도를 낮추거나 계획 짜기 게임 단계에서의 추정을 위해 분할되거나 통합되며, 스파이킹을 위해 보다 구체적으로 기술되면서 분할되기도 한다.

3. 유즈케이스 모델링을 위한 유저 스토리 분석 방법

3.1 유저 스토리의 분석

유저 스토리의 요소 중 하나인 이름에는 일반적으로 기능이나 클래스 또는 특정 동작을 위한 모듈의 특징을 보여 준다. 그러므로 그 자체가 하나의 유즈케이스가 될 수 있다.

N^1 : P^1 이 A^1 을 통한 O^1 의 달성을 요구한다. [O^1 에 대한 조건은 $C^1 \dots C^n$ 이다. [P^n 은 P^{n-1} 의 A^{n-1} 을 위해 A^n 을 행한다.]]

유저 스토리는 일반적으로 위와 같은 형식을 가지는데 여기에서 []안의 내용은 존재하기도 하고 존재하지 않을 수도 있다.

이름 N^1 은 유즈케이스의 후보이며, 고객 또는 시스템인 P^1 은 행위자(Actor)의 후보이다. 각각의 A^n 은 유저의 인터페이스 또는 시스템이 수행할 연산의 후보이다. C^n 은 유즈케이스 문서에서의 사전 조건 또는 사후 조건의 후보이다. 그리고 유저 스토리에 기술된 순서는 유즈케이스 문서의 작동 순서가 된다.

3.2 유즈케이스 생성

3.1에 나온 유저 스토리의 형식에 3.1의 '방법을 적용하여 유즈케이스를 작성한다면 [표 2]과 같이 재구성될 수 있다.

표 2 도출된 유즈케이스

유즈케이스 ID: UC1	
유즈케이스 이름: N ¹	
작성자: John Doe	최종 갱신자: Jill Doe
작성일: 200X. X. XX	최종 갱신일: 200X. X. XX
행위자: P ¹	
설명: P ¹ 이 A ¹ 을 위해 A ¹ 을 행한다.	
사전 조건:C ¹ ... C ⁿ	
사후 조건:C ^m ... C ^o	
작동 순서:P ¹ 이 A ¹ 을 행한다.	
P ⁿ 이 A ⁿ 을 행한다.	
...	

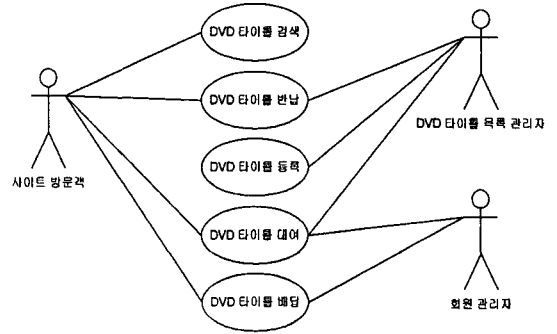


그림 1 모델링된 유즈케이스 다이어그램

나머지 항목 들은 스파이킹과 태스크 작성후 추가 된다. 이렇게 작성된 유즈케이스를 바탕으로 유즈케이스 다이어그램을 모델링한다.

4. 적용 사례 - DVD 타이틀 대여 웹사이트 개발 프로젝트

본 논문에서 제안된 방법을 DVD 타이틀 대여 웹사이트 개발 프로젝트의 기획단계에서 적용하였는데, 관련 부서들 간의 요구사항 의사소통에 사용되었다. [그림 1]은 본 고에서 제안된 방법을 적용하여 [표 3]의 유저 스토리를 간략히 모델링한 유즈케이스 다이어그램이다.

표 3 프로젝트에 사용된 유저 스토리의 일부

- * DVD 타이틀 검색: 고객이 자신이 원하는 DVD 타이틀이 관리 데이터베이스의 보유 목록에 있는지 그리고 있다면 대여 가능한지를 검색한다. 한글 또는 영문 제목, 장르, 국가, 출시일, 감독, 출연배우를 기준으로 검색할 수 있다.
- * DVD 타이틀 대여: 고객이 원하는 DVD 타이틀이 보유 목록에서 대여 가능으로 표시되어 있는 경우, 해당 타이틀을 대여하면, 회원관리자가 확인인지 판별하여 회원이면 대여를 허용하고, 목록 관리자는 데이터베이스에 대여된 타이틀로 표시하고 배달 준비를 한다.
- * DVD 타이틀 배달: 회원 관리자로부터 배달할 주소를 받은 배달 직원이 대여 타이틀을 배달하고, 목록관리자는 배달한 타이틀로 표시한다.
- * DVD 타이틀 반납: 고객이 타이틀을 직접 반납하거나 또는 직원을 통해 반납하면 목록관리자는 대여가능 타이틀로 표시한다.
- * DVD 타이틀 등록: 구입한 신작 타이틀을 목록 관리자가 데이터베이스에 등록한다.

5. 결론 및 향후 연구 과제

본 논문에서는 XP 기법을 적용한 프로젝트 진행 시 요구사항 명세를 위해 유저 스토리를 사용함으로써 나타나는 부서간 또는 관계자들간의 요구사항 명세서 교환 시 발생하는 문제를 유즈케이스 모델링을 통하여 해결하고자 하였다. 향후 과제로는 보다 다양한 사례에 적용해봄으로써 이와 같은 분석 방법을 적용하는 것이 불가능한 유저 스토리가 등장하는 경우 이를 해결할 수 있는 방안을 모색하고, 고객으로 하여금 자연어로 기술되는 유저 스토리를 보다 정련할 수 있는 유도 기법이나 지침에 대한 연구가 필요하다.

6. 참고문헌

- [1] Boehm, B. W. "Software Engineering Economics", Prentice Hall, 1981
- [2] European Software Institute, "European User Survey Analysis", Report USV_EUR 2.1, ESPITI project, January 1996
- [3] <http://agilealliance.org>
- [4] <http://www.controlchaos.com>
- [5] <http://crystallmethodologies.org>
- [6] <http://www.featuredrivendevelopment.com>
- [7] Highsmith, J. A. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. New York, NY: Dorset House, 2000
- [8] Beck, Kent. Extreme Programming Explained: Embracing Change. Reading, MA: Addison Wesley, 1999
- [9] Cohn Mike, User Stories Applied: For Agile Software Development. XP/Agile Universe 2003, August 2003