

신디케이션을 활용한 비정형 프로세스 기반의 소프트웨어 개발방법론 연구

김민균*, 박용태**

*서울대학교 공과대학 협동과정 기술경영대학원

**서울대학교 공과대학 산업공학과

Abstract

본 연구는 기존 소프트웨어 개발 방법론의 이론적 근거를 파악하고, 이에 대한 분석을 통해 비프로세스 영역과 불규칙 프로세스 영역을 처리하는데 적합한 새로운 소프트웨어 개발 방법론을 정의하고 이를 바탕으로 한 Framework과 Platform을 개발하는 데 그 목적이 있다.

비정형 소프트웨어 개발 방법론은 요구변동성이 높은 소프트웨어 개발, 혹은 선도 기술 개발 프로젝트의 경우에 활용 가능성이 높은 방법론이라 할 수 있다. 이와 같은 특징을 지닌 소프트웨어 개발에 있어서 중요한 점은 요구사항과 사용자(개발자)의 지속적인 Feedback에 대응해야 하는 것이기 때문에 본 소프트웨어 개발 방법론은 과정 중심의 대응 방법을 제시하고 있다.

본 연구를 통하여 프로세스가 갖춰지지 않은 Firm Level에서의 Task Transaction과 Management를 Syndication을 활용하여 구축하고, 기존 소프트웨어 개발 방법론과의 차이점을 검토하고 그 가능성을 살펴보고자 한다.

Keywords:

Syndication, Trackback, Software Development, Process Management, Knowledge Management, Expert System

1. 서론

IT산업이 발전해감에 따라 기술의 속도와 변화가 빨라지고, 고객의 요구사항이 다양화, 복잡화 되고 있다. 따라서 이의 수용여부와 대응이 소프트웨어의 품질과 경쟁력 제고에 무엇보다 중요해 지고 있다.[2]

소프트웨어는 생명주기(Software Life Cycle)를 가지고 있다.[9] 소프트웨어가 거치는 생명주기 단계를 소프트웨어 개발 주기(Software Development Life Cycle)라고 하며 여러 가지 모형이 제안되어 있다. 소프트웨어 개발 모형은 소프트웨어 엔지니어가 여러 기술적 업무를 어떤 순서로 할 것인지에 대한 지침을 제공한다.

소프트웨어 개발 주기에 따른 SDLC모형은 Waterfall Model, Prototype Model, Spiral Model, USPD(Unified Software Development Process), Agile Methodology 등으로 나눌 수 있다.

기존의 소프트웨어 방법론들은 초기설계에 기반한 산출물 중심의 개발 방법론이 대부분이었다.

특히, USPD를 바탕으로 하는 RUP(Rational Unified Process) 중심의 개발 방법론은 소프트웨어 개발의 전체 생명주기를 지원하는 프로세스 프레임워크이다. 즉, 바로 적용하여 사용할 수 있는 방법론이라기보다는 다양한 유형의 소프트웨어 시스템, 도메인, 그리고 조직을 위한 프로세스의 프레임워크를 제공해준다.[14]

이렇듯 RUP는 대규모 프로젝트에 적합한 프로세스를 갖추고 있어 많은 부하가 따르고 설계 변경에 있어 빠른 대응이 힘들었다.

이에 비해 Agile Methodology를 대표하는 XP(eXtreme Programming)는 요구사항 변경으로 인한 비용이 개발 기간에 상관없이 일정하게 유지되도록 하는 것을 주목적으로 한다. XP는 설계를 단순하고 명확하게 유지하려고 하며 가장 우선순위가 높은 것을 먼저 개발한다.[10],[11] 이러한 특징을 가진 XP는 Test First Programming과 개발자 중심의 개발 방법으로서 소규모 프로젝트를 다루는 데 있어 많은 각광을 받고 있다. 하지만 XP의 적용에 있어 가장 힘든 부분은 테스트이라고 할 수 있다. 웹 중심의 분산환경 하에서 다양한 프로젝트를 진행함에 있어 유닛테스트 환경을 갖추기도 힘들 뿐만 아니라, 잦은 변경에 대한 결과 확인이 자주 발생되다 보니 클라이언트 환경에 비해 효용성이 떨어진다.

본 연구는 기존 소프트웨어 개발 방법론의 이러한 취약점을 인지하고, 이에 대한 분석을 통해 비즈니스 로직 기반의 유연성을 확보한 새로운 소프트웨어 개발 방법론을 정의하고 이를 바탕으로 한 프레임워크와 PMS(Project Management System) Tool을 개발하는 데 그 목적이 있다.

2. 이론적 배경

2-1. 요구사항 관리

요구사항 관리란 요구사항을 수집하고 문서, 이메일, 사용자 인터페이스 스토리보드, 실행 가능한 프로토타입, 그 외 여러 형식을 이용해 요구사항을 기록하고 요구사항과 맞지 않는 설계나 코드를 조사하고 프로젝트 전반에 걸쳐 변경 사항을 관리하는 과정을 말한다.[1]

이러한 요구사항 관리에 있어 기존에는 관리

방법의 융통성 및 대응이 취약했고, 구조적으로 요구사항 관리가 잘 정의되어 있어도 복잡도가 높아 이를 활용함에 있어 부적절한 경우가 많았다. 또한 일정 및 예산 초과, 기능 미흡을 초래하는 핵심적인 요소로 사용자 참여 부족, 미흡한 요구사항, 요구사항 변경[6],[12] 불충분한 요구사항 관리[13] 등이 제기되었다.

2-2. 신디케이션

신디케이션(Syndication)의 원 뜻은 ‘분배’ 또는 ‘배급’이란 뜻이다.

신디케이션이 온라인으로 확장되면서 콘텐츠를 필요로 하는 수요자와, 콘텐츠를 생산하는 공급자를 중간에서 연결시켜 주는 서비스의 개념으로 쓰이고 있다. 더불어 이러한 서비스의 주체를 콘텐츠 신디케이터(Content Syndicator)라 한다.

신디케이션의 활용분야는 크게 세 가지로 볼 수 있다. 사이트의 콘텐츠 배급, 수집 그리고 사용자의 콘텐츠 사용 편리성 제공을 들 수 있다.

신디케이션은 필요에 따라 정보를 재포장 및 용도 변경하도록 하는 서비스를 제공하는 한편 분석 혹은 콘텐츠 통합을 통한 부가 가치를 창출한다.

2-3. 트랙백

트랙백(Trackback)의 기본 개념은 “어느 특정 글에 연관된 내용의 글을 내가 작성함에 있어서 그 사실(글이 작성된 것)을 알리는 것”이다.

즉, 간단히 말해서, 트랙백은 웹사이트 간에 서로 뭔가를 알려줄 수 있는 수단을 제공하기 위해 만들어진 것이다.

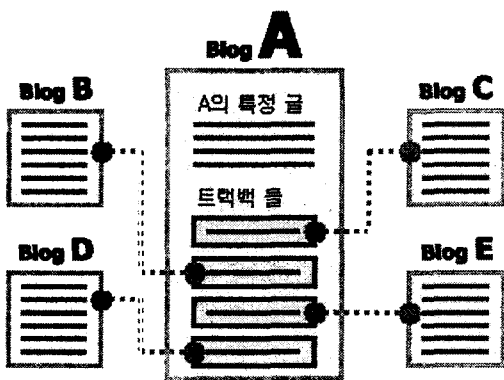


그림 1 - 트랙백 구조

트랙백은 웹 상에서의 커뮤니케이션에 있어서 새로운 방식 중 하나이며 보다 효과적으로 요구사항 관리 및 개발을 하는데 있어 적용 될 수 있다.

3. 트랙백을 통한 비정형 프로세스 기반의 소프트웨어 개발 방법론 연구

3-1. 비정형 프로세스

비정형의 사전적인 의미는 ‘부정형(不定型)의, 불규칙적인’이란 뜻이다.

따라서 본 연구에서 정의하는 비정형 프로세스란 ‘프로젝트 진행 전에 세부 명세를 작성하는 것이 모호한 프로세스’로써 Iteration Process 정의가 힘든 프로세스를 말한다.

다시 말해 ‘프로젝트의 요구변동성이 높은 프로세스’를 일컫는다.

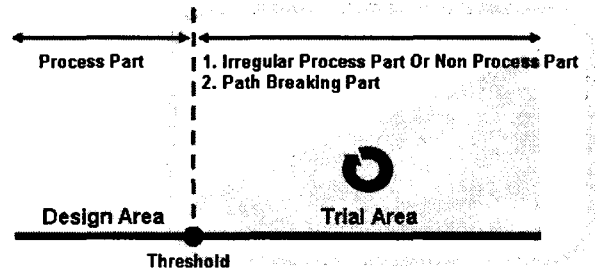


그림 2 - 비정형 프로세스 정의

3-2. 비정형 프로세스의 특징

비정형 프로세스는 B2C를 기반으로 하는 요구변동성이 높은 소프트웨어 개발, 혹은 선도 기술 개발 프로젝트의 경우 많이 나타난다. 비정형 프로세스는 요구사항과 사용자(개발자) 간의 지속적인 피드백에 빠르게 대응해야 성공 가능성이 높기 때문에 프로젝트 중반까지 추가 기획이 반영될 수 있다. 그리고 과정 중심의 신속한 의사결정이 필요하므로 이에 대한 지표설정이 무엇보다 중요하다.

표 1 - 트랙백을 활용한 비정형 프로세스 접근 시 장·단점 분석

장 점	<ol style="list-style-type: none"> 1. 선택적 구독 - 사용자가 원하는 topic과 정확히 일치하는 channel 선택. 2. 빠른 구독 - 동시에 다양한 channel 소스 접근. 3. History 관리 - 다양한 channel의 과거 기록들 보관이 가능. 4. 커뮤니케이션 방식의 변화 - 1:1에서 1:N으로 발전.
	<ol style="list-style-type: none"> 1. 선각자의 오류 - 초기 방향 설정 오류 시 문제의 왜곡.

단	2. Multi Issue 도출 - 문제의 본질이 흐려질 가능성이 존재.
점	3. 문제의 소멸 - 중요한 문제임에도 불구하고 전문가의 부재나 관심영역에서 벗어 날 때 Priority가 밀리게 될 경우.

그림 4-Task의 Status Flow

그림 3은 비정형프로세스의 전체적인 Sequence를 보여주고 있다. 그림 4는 Task의 처리에 대한 Status와 그 흐름도를 나타내고 있다.

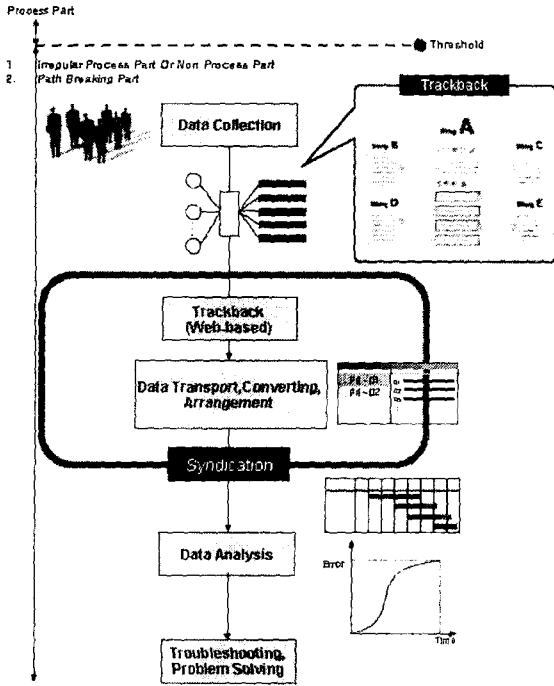


그림 3- 비정형 프로세스의 Logical Flow

3-3. 비정형 프로세스의 Flow 및 프레임워크

비정형 프로세스는 하나의 Task를 단위로 형성되며, 각 Task는 분류(Category)와 우선순위(priority), 상태(Status) 파라미터를 통해 계층이 분류된다. 그리고 처리되는 Task의 상태에 따라 Task를 수행할 수 있는 권한이 바뀔 수 있고, 이는 실시간 레포팅을 통해 그 결과를 파악할 수 있게 된다.

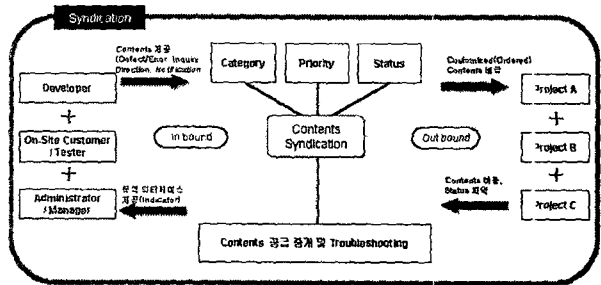


그림 5- 비정형 프로세스 Framework

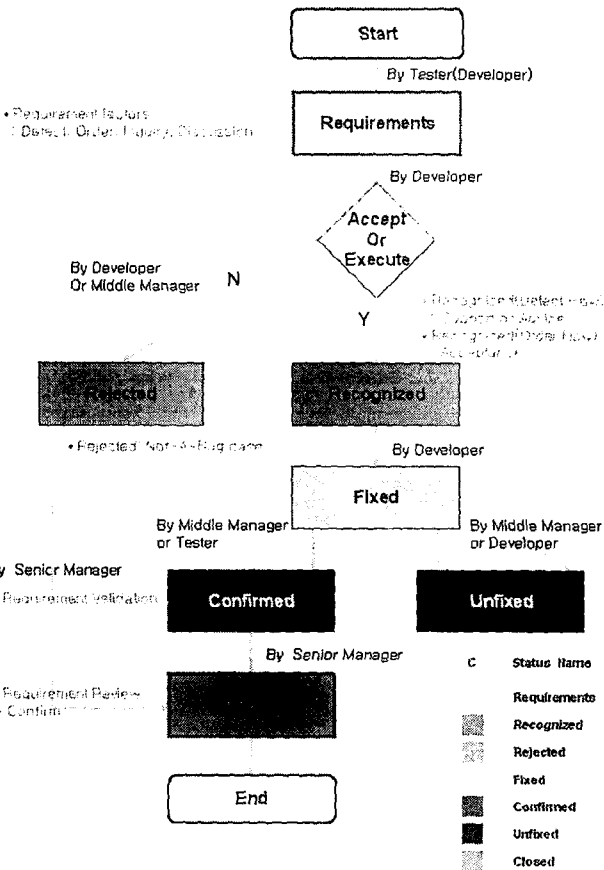
비정형 프로세스는 그림 5에서와 같이 개발자나 이해관계자의 요구사항을 확인하고 이에 대한 다양한 의견이나 제안, 솔루션을 취합하고 검토하여 확인하는 과정을 거치게 된다. 이러한 이해관계자간의 상호작용은 개발 프로세스를 진행함에 있어 필수적으로 고려되어야 하는 요소이다.[5]

이러한 과정 속에서 관리자는 각 Parameter에 대한 Status를 관찰하여 Task의 진행 절차 및 상황을 이해하고, 정상적이지 않은 Task는 원인 분석 및 Troubleshooting을 통한 해결안을 도출한다. 또한 완료된 Task는 확보된 History 절차에 따라 다른 Task를 수행함에 있어 프로세스의 확인(Validation)과 평가(Evaluation)에 대한 기준 및 참고자료로 활용될 수 있다.[4]

4. 연구 결과

본 연구에 제시된 신디케이션 프레임워크를 기반으로 다음과 같은 소프트웨어 개발을 위한 웹 기반의 PMS 시스템을 구축하였다.

이는 프로젝트를 수행함에 있어 다양한 Task를 처리하기 위한 요구사항을 수집하고, 수집된 요구사항의 분류를 통해 개발자 Pool 내에서의 커뮤니케이션을 효율적으로 진행시키도록 추진하고, 관리자에게 각각의 Task가 어떤 식의 흐름을 통해



진행되는지를 보여주게 된다.

그림 6은 요구사항에 대한 Task의 분류 및 처리과정을 그림 7은 각 Task의 전체적인 상황을 Monitoring하는 화면을 보여주고 있다.

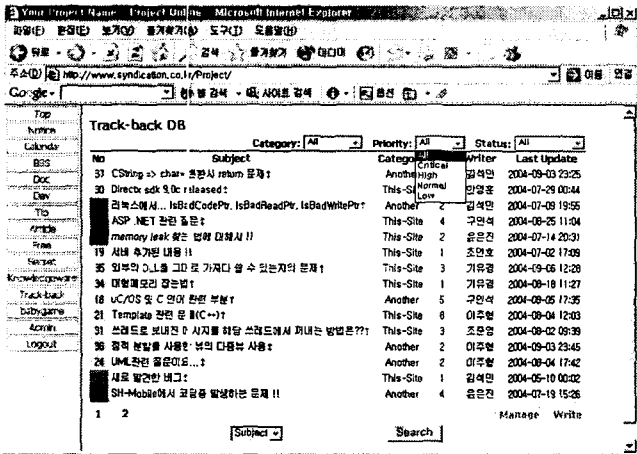


그림 6-트랙백을 활용한 PMS Screenshot

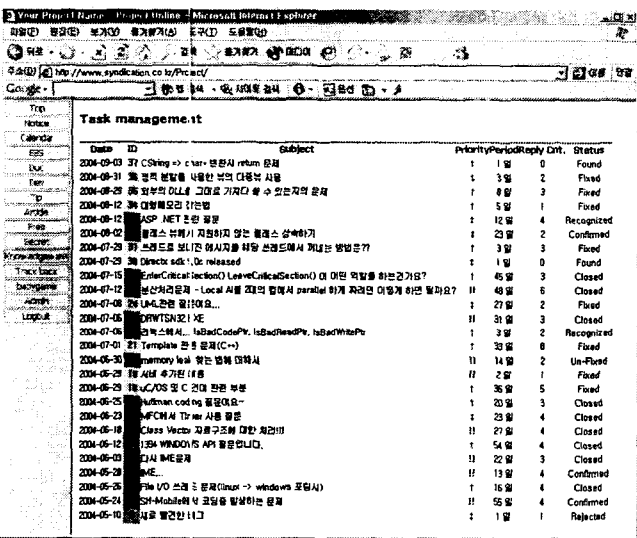


그림 7-Task management

5. 결론

본 연구는 기획-개발 연동의 불확실성과 선도기술 중심의 프로젝트에서의 비 프로세스 영역과 불규칙 프로세스 영역을 다루는 소프트웨어 개발 방법론에 대해 기술하였다. 그리고 불확실성을 줄이기 위한 위험관리 요소를 정의하고 실시간 환경에서의 Task에 대한 프로세스가 어떻게 처리되는지를 살펴보았다. 신디케이션을 통한 비정형 프로세스 상에서 사용자는 내외부 출처의 다양한 콘텐츠의 접근으로 정보의 처리 및 의사결정의 효율성을 높이고, 정보 분석 및 광범위한 콘텐츠 조합을 통한 부가 가치 창출이 가능하다.

향후 과제는 소프트웨어 개발에 있어 타 방법론과의 융합을 통한 복합적인 솔루션 개발과 더불어 RSS(Really Simple Syndication)를 이용하여 수많은 상이한 기술, 다양한 개발언어와의 인터페이스 및 이 기존의 기술에 대한 전문성을 확보할 수 있는 소프트웨어 개발 방법론을 개발하는 것이다.[7]

참고문헌

- [1] Kotonya, G., and Sommerville, I. (1998) Requirements Engineering: process and techniques, John Wiley & Sons Ltd.
- [2] Kwak, Y. H., and Stoddard, J. (2003) Project risk management: lessons learned from software development environment, Technovation, In Press, Corrected Proof, Available online.
- [3] Arora, V., and Cooke, J.E. (1991) Developing and Managing Expert System Programs, Proceedings of the IEEE/ACM International Conference on, pp 339 - 345.
- [4] O'Leary, T.J. et al. (1990) Validating expert systems, Expert, IEEE [see also IEEE Intelligent Systems] , Volume: 5 , Issue: 3, pp 51 - 58.
- [5] Yau, C. and Sattar, A. (1994) Developing expert system with soft systems concept, Expert Systems for Development, Proceedings of International Conference on , pp 79 - 84.
- [6] Software Engineering Institute, Carnegie Mellon University. (1994) The Capability Maturity Model: Guidelines for improving the software process, Addison Wesley.
- [7] Lindahl, C. and Blount, E. (2003) Weblogs: simplifying web publishing, Computer, Volume. 36, Issue 11, pp 114 - 116.
- [8] Humphrey, W. (2001) Winning with Software, Addison Wesley.
- [9] McConnell, S. (1996) Rapid Development, Microsoft Press.
- [10] Beck, K. (1999) Extreme Programming Explained : Embracing Change, Addison Wesley.
- [11] Ann, A. (2000) Extreme Programming Installed, Addison Wesley.
- [12] Dennis, M. (1994) Charting the Seas of Information Technology, The Standish Group.
- [13] Kitson, D.H. and Masters, S.M. (1993) An analysis of SEI software process assessment results : 1987-1991, Software Engineering 15th International Conference, pp 17-21.
- [14] Bergstrvm, S. and Reberg, L. (2003) Adopting the Rational Unified Process: Success with the Rup, Addison Wesley.