# 애자일 기반의 유비쿼터스 다중 프로젝트 관리 및 협업

이지현* 허성진*
*한국전자통신연구원
e-mail : {jihyun, sjheo}@etri.re.kr

# A Study on Project Management and Collaboration in Ubiquitous Multi-Project Environment based on Agile Methodology

Jihyun Lee* Sung Jin Hur*
*Electronics and Telecommunications Research Institute

요 약

유비쿼터스 멀티 프로젝트 환경에서 프로젝트 팀은 프로젝트 결과를 얻기 위해 필요한 업무를 도출하여 실행한다. 이를 위해 실질적인 팀 관리 및 요구 사항 정제가 이뤄져야 성공적인 제품 출시로 연결될 수 있다. 프로젝트 관리에서 중요한 부분은 프로젝트 진행 사항에 대한 가시적인 조회와 평가가 중요하다. 그러기 위해서는 협업과 공동 책임 업무에 대한 지원이 필요하다. 이 논문에서 애자일 개발 방법론에 기반한 프로젝트 관리 및 협업 지원 방법 및 시스템을 제안한다.

## 1. 서론

물리적 시간과 공간의 제약 없이 소프트웨어 개발 프로젝트를 관리하는 활동은 점차 발전하였다. 특히 유비쿼터스 다중 프로젝트 환경에서 프로젝트 팀들은 다른 프로젝트에서 파생되는 상대적인 상황 정보를 프로젝트에 반영하기 위해 협력하고 자신의 실제 업무를 이에 맞춰 조정하는 필요가 증가해 왔다 [1]. 이를 위해서 실용적으로 팀을 조직하고, 프로젝트 관리 활동을 이행해야 한다. 이를 위해 프로젝트 팀은 웹을 통해 중앙 집중화된 어플리케이션의 형태로 유비쿼터스 멀티 협업 및 프로젝트 관리를 지원하는 솔루션을 제공할 수 있다. 웹 기반 프로젝트 관리 솔루션은 프로젝트 관리에 대한 정보를 언제 어디서나 관리할 수 있고, 팀 구성원들과 효율적으로 협업할 수 있다. 또한, 프로젝트 상태에 대한 가시성을 제공하여 프로젝트 진행 정보에 대한 팀 공유가 가능하다.

소프트웨어 개발 프로세스 중 하나인 애자일 개발 프로세스는 빈번한 개조(adaption)에 대한 신속한 납기를 보장하고 팀 협업 및 변화하는 요구 사항에 대한 적응을 강화한다. 따라서 개발을 반복하는 동안 프로젝트 관리자는 비용 및 일정에 관한 초기 데이터를 기반으로 파생된 요구 사항 및 제약 조건을 개발 중인 소프트웨어에 반영함으로써 개발 작업에 소요될 시간과 필요한 리소스에 대한 예측이 가능하다.

본 논문은 유비쿼터스 환경에서 다중 프로젝트를 관리하고 팀원들 간의 협업을 돕기 위한 애자일 프로세스 기반의 프로젝트 관리 솔루션의 개발에 대해 기술한다.

본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 일환으로 수행하였음. [2009-S-033-01, 중소기업 SW 서비스를 위한 SaaS 플랫폼 개발]

## 2. 애자일 소프트웨어 개발 원칙

민첩한 프로세스에 따라 더 나은 소프트에어를 만들기 위한 이념으로부터 애자일 소프트웨어 연맹에서 12 가지 애자일 소프트웨어 개발 원칙을 결정하였는데 세부 내용을 살펴본다 [2][3].

(1) 고객을 만족시키는 것을 최우선으로 한다.
(2) 변경을 환영한다.
(3) 동작하는 소프트웨어를 자주 인도한다.
(4) 사업부 사람들과 개발자들은 매일 함께 일해야 한다.
(5) 의욕에 찬 사람들로 프로젝트 팀을 구성한다.
(6) 얼굴을 맞대고 의사 소통하는 것이 가장 좋은 방법이다.
(7) 동작하는 소프트웨어가 프로젝트의 진척을 가늠한다.
(8) 지속 가능한 개발을 촉진시킨다.
(9) 우수한 기술과 좋은 설계에 대한 지속적인 관심은 민첩성을 강화시킨다.
(10) 하지 않아도 될 일을 최대화 시키는 단순화 기술은 필수적이다.
(11) 최고의 아키텍처, 요구 사항, 설계는 자율적으로 구성된 팀에서 나온다.
(12) 정기적으로 팀과 팀의 프로세스를 살펴보고 개선 사항을 찾아 본다.

어떤 원칙들은 상당한 논쟁 가능성이 있지만, 전반적으로 이 원칙은 전통적인 소프트에어 개발 방법론과 차별성을 갖는 애자일 방식으로 소프트웨어를 개발하고자 하는 개발 팀과 개발자들에게 경량화된 개발 프로세스를 제공한다. 또한 코드 중심의 개발로 프로토타입을 반복적으로 제공하여 궁극적으로 프로세스 중심의 활동에서 사람 중심의 개발 활동이 이뤄

지는데 초점을 둔다.

## 3. 애자일 접근을 통한 프로젝트 관리

애자일 프로젝트 관리에서 가장 중요한 요소는 요구 사항의 효과적인 관리이다. 애자일 프로세스에서 요구 사항은 스토리 카드[4]의 단위로 추출하여 관리한다. 스토리 카드를 사용한 요구사항 관리는 개발 팀원들 간의 의사 소통에 중점을 둔다. 하지만 큰 규모의 개발 팀에서 스토리 카드의 수가 급격히 증가하면 스토리간의 관계를 이해하는 것이 점차적으로 어려워지게 되므로 스토리 카드 기반의 요구 사항 관리는 중소기업 규모의 프로젝트와 개발 조직에 권장할 만하다. 스토리 카드는 반복 계획에 할당되어 스토리 카드 간의 추적성과 연관 관계가 관리되므로 매 반복 주기 동안 개발자들은 계획된 스토리 카드를 완료하여 완료 상태를 관리 솔루션에 반영하여야 한다. 스토리 카드 기반의 프로젝트 계획은 프로젝트 팀원들 간의 의사 소통을 브레인스토밍, 계획 게임(planning game), 일일 회의 등의 기법[5][6]을 통해 향상시키지만 상세 요구 사항과 제약 사항에 대한 도출은 계획된 반복 계획이 종료되는 첫 번째 배포(release) 계획 단계 이후에 정의할 수 있으므로 상당히 우선 순위가 높은 요구 사항에 대한 처리가 먼저 이뤄지고 나머지들은 다음 처리로 미뤄지도록 함께 계획한다. 개발된 프로그램을 빈번히 전달(delivery)할 수 있도록 한다.

## 4. 애자일 기반의 다중 프로젝트 관리 도구

애자일 프로젝트 관리 방법을 이용한 유비쿼터스 다중 프로젝트 관리 시스템인 APMAL(Agile Project Management tool)에서 지원하는 일일 회의, 배포 계획, 일정 모니터링을 위한 세부 프로세스를 설명한다.

**<표 1> APMAL 지원 프로세스**

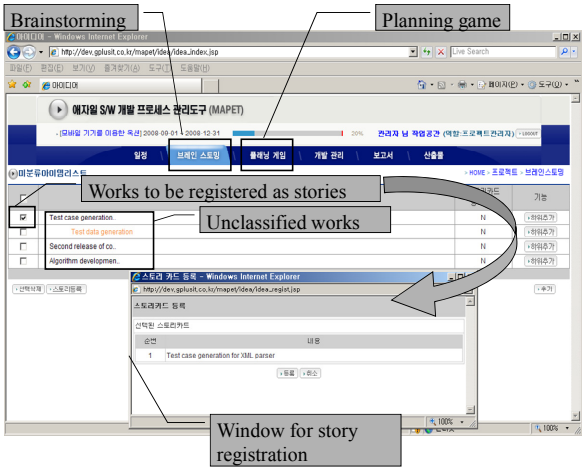
| 기능 | 설명 |
|---|---|
| 일일 미팅 | 스토리 카드 검토/할당/완료 점검 |
| | 요구사항 검토, 이슈, 버그 검토 |
| 배포 계획 및 모니터링 | 스토리 카드 배포 계획, 계획의 공통 소유권, 프로젝트 일정 추적 |
| 반복 계획 및 모니터링 | 사용자 스토리에 대한 반복, 계획의 공통 소유권, 계획 추적 |

첫째, 일일 회의에서 결정되는 스토리 카드의 할당, 작업 우선순위의 일임, 잔여 작업의 분리, 완료 작업 확인 정보가 APMAL 에 반영된다. 또한 일일 회의에서 도출된 새로운 요구 사항, 프로젝트 상태, 해결되어야 할 문제에 대해서도 반영한다. 그림 1 은 스토리 카드를 이용한 프로젝트 계획에 대한 화면이다. 브레인스토밍, 계획 게임을 위한 탭 단위의 UI 를 제공하였고 도출된 스토리 카드는 등록 창을 통해 등록된다.

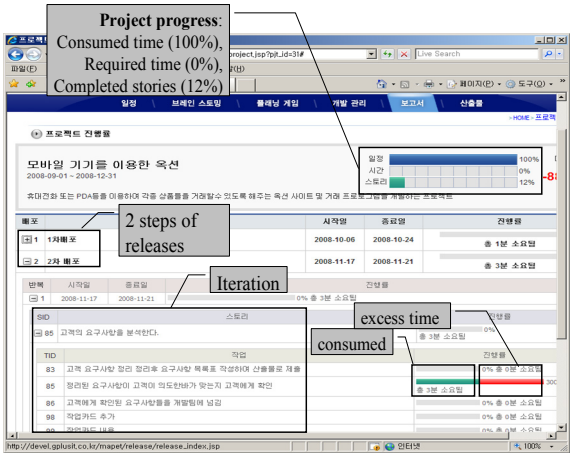
둘째, 배포 일정을 계획하고 모니터링 한다. 배포 일정이 짧더라도 다음 배포를 위해 배포 일정이 완료된 후 계획 대비 수행 실적을 수집하여 반영한다.

셋째, 반복 일정을 계획하고 추적한다. 스토리 카드의 수행 요소인 작업(task)이 두 개발자가 함께 수행해야 하는 할 때 작업에 대한 책임 분리하여 할당할 수 있다. 스토리 카드 완료 조건에 미흡한 경우 전체 반복 일정 하에 반복하여 완료될 수 있도록 한다.

그림 2 는 배포 계획에 대한 프로젝트 진행률을 소요 시간, 계획된 시간, 완료된 스토리 카드 개수로 보여준다. 두 단계의 배포 계획은 여러 반복으로 구성되어 있고 소용된 시간과 초과 수행 시간에 대해 막대 그래프로 시각화한다.




(그림 1) 스토리 카드를 이용한 프로젝트 계획




(그림 2) 배포 계획에 대한 프로젝트 진행률

## 참고문헌

[1] Roger S. Pressman. “Software Engineering, A Marcus Goncalves, “Implementing the Virtual Project Management Office: Proven Strategies for Success,” McGraw-Hill, Aug. 2006.

[2] Kent Beck, Robert C. Martin, Alistair Cockburn, Jim Highsmith, "Manifesto for Agile Software Development", www.agilemanifesto.org, Feb 2001.

[3] Alistair Cockburn, “Agile Software Development: The Cooperative Game”, Addison-Wesley Professional, 2006.

[4] James W. Newkirk, “Extreme Programming,” Addison-Wesley Professional, 1st edition, June 2001.

[5] James W. Newkirk, “Extreme Programming,” Addison-Wesley Professional, 1st edition, June 2001.

[6] Jeff Sutherland, “Agile Development: Lessons Learned from the First SCRUM,” Oct. 2004.