

2013. 8. 26. [제64호]

# SW개발 방법론 : Scrum과 Kanban의 비교

소프트웨어공학센터 경영지원TF팀

## C o n t e n t s

- ▶ 서론
- ▶ Scrum과 Kanban의 산출물 차이
- ▶ 개발 프로세스의 진척을 추적하기 위해 사용되는 척도의 차이
- ▶ 역할과 미팅에 있어 Scrum과 Kanban의 차이
- ▶ 이행측면에서의 Scrum과 Kanban의 차이

## Key Message

애자일 개발방법론의 대두와 함께 떠오른 Scrum은 많은 개발자들에 점차 친숙해지고 있으며, 최근에는 Kanban이 떠오르고 있어 Scrum을 마스터한 조직들은 점차 Kanban으로 옮겨가고 있는 추세에 있음. 이에 따라 Scrum과 Kanban을 구분 짓는 차이점과, 개발방법론 이행시 차별성, 방법론 간 상호보완 관계 등에 대한 여러 가지 시사점을 제시함.

## ▶ 서론

- Scrum은 개발자들 사이에서 애자일과 동의어로 받아들일 정도로 일반화되었으며, 최근에는 Kanban이 점차 떠오르고 있는 추세임
- Scrum을 이행한 많은 조직들이 Kanban으로 이동하고 있고 많은 ALM 벤더들이 제품에 Kanban의 특징을 추가하고 있으며 수 많은 컨퍼런스에선 Kanban의 사용을 의미하는 ‘린(lean)’생산방식과 ‘흐름생산(continuous flow)’에 관한 세션을 많이 열고 있음
- 그러나 분명 Kanban이 무엇이며 Scrum과 어떻게 다른가? 이 둘이 함께 사용될 수 있는가? 이번 원고를 통하여 프로젝트 관리자 및 팀 리더들은 Scrum과 Kanban이 어떻게 다르고 어떻게 상호보완적이 될 수 있는지 보다 자세히 살펴보게 될 것임

## ▶ Scrum과 Kanban의 산출물 차이

- ‘애자일 개발의 떠오르는 새로운 방식인 Kanban에 대한 탐구<sup>1)</sup>’는 블로그 게시판에서 언급된 바와 같이, 그리고 Henrik Kniberg에 의한 Kanban과 Scrum의 비교에서, Scrum은 아래의 9개의 규칙, 그리고 Kanban은 3개의 규칙을 가지고 있다고 일컬어짐
  - 1. 스크럼 마스터(Scrum Master)<sup>2)</sup>
  - 2. 제품 책임자(Product Owner)<sup>3)</sup>
  - 3. 스크럼 팀(Team)<sup>4)</sup>

1) Exploring Kanban, agile development' s new rising star,  
<http://itknowledgeexchange.techtarget.com/software-quality/exploring-kanban-agile-developments-new-rising-star/>

2) 스크럼 팀의 업무를 방해하는 요소를 제거하기 위해 노력하는 역할자

3) 제품 기능목록에 해당하는 제품 백로그를 만들고 우선순위를 조정하거나 새로운 항목을 추가하는 일을 관리하는 역할

4) 하나의 스프린트 기간 동안 구현해야 할 기능을 사용자 스토리로 도출하고 이를 구현하는 역할

- 4. 스프린트 계획<sup>5)</sup> 미팅(Sprint planning meeting)
- 5. 일일 스크럼(Daily Scrum)<sup>6)</sup>
- 6. 스프린트 리뷰(Sprint review)<sup>7)</sup>
- 7. 제품 백로그(Product backlog)<sup>8)</sup>
- 8. 스프린트 백로그(Sprint backlog)<sup>9)</sup>
- 9. 소멸 차트(Burndown chart)<sup>10)</sup>
- Kanban은 불과 아래 세 가지 사항만이 필요함
  - 1. 워크플로우의 시각화(Visualize the workflow)
  - 2. 워크-인-프로세스(WIP, work-in-process)<sup>11)</sup> 제한
  - 3. 플로우의 측정 및 최적화(Measure and optimize flow)

## ▶ 개발 프로세스의 진척을 확인하기 위해 사용되는 척도의 차이

- Scrum에서는 타임 박스(time-boxed)된 혹은 특정 시(時)구간 내에서 반복 수행되는 팀 작업을 스프린트(sprint)라 부르는데, 보통 2주에서 4주간 수행됨
  - 각 반복의 시작 시점에 스프린트 계획 미팅이 있는데 이 미팅을 통해 제품 백로그로부터 스토리를 선택하고 그 다음 반복에서 개발을 위한 스토리를 선택하게 됨
- 각 반복에서 작업이 얼마나 많이 수행되어질 수 있는가를 결정하기 위하여, 번다운 차트를 사용하면서 팀 퍼포먼스를 추적하게 됨
  - 차트의 x축은 수행되어질 작업량의 예측치를 나타내고 y축은 반복 시간이 됨
  - 번다운 차트는 팀의 속도를 측정하는데 측정치를 통하여 각 반복 내에서 얼마나 작업이 생산적인지를 나타낼 수 있음
  - 종종 “스토리 포인트”에서 수행되는 각 스토리 예측작업은 한 번의 반복과정 내

- 
- 5) 각 스프린트에 대한 목표를 세우고 제품 백로그로부터 스프린트에서 진행할 항목을 선택하고 각 항목에 대한 담당자를 배정하고 태스크 단위로 계획을 수립
  - 6) 매일 진행하는 15분간의 프로젝트 진행상황을 공유하고 회의하면서 각자 한 일, 할 일, 문제점 등을 논의하는 것
  - 7) 스프린트 목표를 달성했는지 진행 및 결과물을 확인하는 회의, 스크럼 팀은 스프린트 동안 작업한 결과를 데모하고 피드백을 받음. 스크럼 마스터는 스프린트 동안 잘된 점, 아쉬운 점, 개선할 사항 등을 찾기 위한 회고를 진행
  - 8) 제품 기능의 우선순위를 정리한 목록
  - 9) 하나의 스프린트 동안 개발할 목록(사용자 스토리와 이를 완료하기 위한 작업을 태스크로 정의)
  - 10) 개발을 완료하기까지 남은 작업량을 보여주는 그래프
  - 11) WIP(재공품): 저장 또는 판매 가능한 상태에 있는 반제품과는 다르게 재공품은 앞으로 더 가공하여 야만 제품이나 부분품이 될 수 있는 것

에서 완성될 수 있는 적절한 크기로 과업을 나누기 위해 필요함

- Kanban에서는 프로젝트가 제대로 진행되고 있는지 아닌지 결정하기 위해 다른 척도가 사용됨
  - 워크플로우를 표현하기 위해 대기행렬이 사용되는데, 예를 들어 백로그 내에 모든 아이템들 즉 개발 대기, 테스트 대기, 배포 릴리즈 대기 등의 대기행렬이 있을 수 있음
  - 종종 Kanban보드가 사용되고 스토리 카드가 진척이 될 때마다 보드사이를 움직임
- Kanban은 스토리를 타임 박스화하기 보다는 스토리가 한 번의 반복과정 내에서 완성되어지도록 WIP을 제한함
  - 팀 구성원은 각 프로젝트 활동에 대해 WIP 제한치를 결정해야 하나 한번 WIP 제한에 봉착하게 되면 팀 구성원들은 방해가 되는 어떠한 영역이라도 제거하기 위해 함께 작업해야 함
- 총 주기 시간(total cycle time)은 스토리를 완성하기 위해 취해지는 시간인데, 이를 측정함으로써 팀은 실제로 스토리를 완성하기 위해 얼마의 시간이 소요되는지를 인지하게 될 것임
  - Kanban 지지자들은 주기를 이용하여 프로젝트 팀이 정확한 예측을 할 수 있다고 확신함

## ▶ 역할과 업무협약에 있어 Scrum과 Kanban의 차이

- Scrum은 매우 특정한 역할과 업무협약(meeting)에 대한 정의를 하고 있는데 스크럼 마스터, 즉 스토리를 코드화하고 테스트하는 제품 책임자와 개발 팀이 존재하여야 함
  - 스크럼 마스터는 미팅을 주관하게 하고 프로젝트를 진척시키는데 있어 팀을 방해하는 모든 장애물을 제거하도록 도움
  - 제품 책임자는 비즈니스와 사용자를 대표하는 것은 물론, 스토리를 우선순위화하고 이를 명확히 하도록 도움
- Scrum은 어떤 스토리들이 스프린트로 가고 있는지를 결정하기 위한 스프린트 계획 미팅이 존재해야함
  - 때때로 일일 스탠드업(daily stand-up)이라 불리는 일일 스크럼 미팅이 있는데 이는 각 구성원들이 그 전날 수행했던 것을 팀이 공유할 수 있도록 짧게 진행되는 상태보고 미팅임

- 스프린트 말미에 소프트웨어가 작동되는지를 데모하고 무엇이 잘 진행되었는지 무엇이 개선될 수 있었는지 시험하기 위한 스프린트 리뷰 미팅이 있음
- Kanban은 Scrum에서 정의된 역할과 미팅을 요구하지 않으나 Scrum 방법론으로부터 과도기를 겪고 있는 많은 팀들은 스크럼 팀에 사용된 미팅과 역할을 유지함
  - Kanban에서는 플로우가 측정되고 최적화되어야 하는데 팀은 이 프로젝트 측정치를 검토하고 지속적인 개선을 수행함

### ▶ 이행측면에서의 Scrum과 Kanban의 차이

- Kanban보다 덜 자율적인 스크럼은 일반적으로 소프트웨어 개발 팀들에 대해 더욱 어려운 이행단계인 과도기를 겪게 될 수 있음
  - 준수해야 하는 특정 프레임워크가 존재하는데 팀이 현재 전통적인 소프트웨어 개발 방법론을 사용하고 있다면 새로운 방식과 새로운 역할로 작업에 큰 변화를 겪을 수 있음
- Kanban은 개발자들이 사용하는 기존 프로세스와 함께 시작하면서 지속적인 개선해 나가도록 설계됨
  - 팀이 자신들의 현재 워크플로우 프로세스를 계획하고 병목현상을 결정하도록 권고하는데 프로세스가 팀을 위한 유일하고 최적화된 워크플로우로 발전한다는 것을 의미함

### ▶ Scrum과 Kanban은 상호보완적인가?

- 아는 바와 같이 Kanban은 방법론 절차를 반드시 수행할 것을 강요하지 않는 대신 플로우를 최적화함으로써 프로세스를 개선하는 것을 의미함
- Kanban은 Scrum과 스토리 혹은 과업이 할당되고 측정되는 세부 방식에 차이가 있더라도 Scrum과 함께 사용될 수 있음
- 이러한 Scrum과 Kanban 모두 테스트 주도(Test-drive) 개발 혹은 지속적 통합(CI)과 같은 애자일이 고려된 기법을 사용할 수 있음

### 참고 자료

1. <http://blog.david-jensen.com/development/scrum-to-kanban-10-lessons-migrate/>
  2. <http://scrumcrazy.wordpress.com/2013/02/04/kanban-vs-scrum-kanban-is-not-for-software-development-but-scrum-is/>
  3. <http://www.eylean.com/blog/2013/05/scrum-vs-kanban-vs-scrumban-planning-estimation-and-performance-metrics/>
  4. <http://itknowledgeexchange.techtarget.com/software-quality/exploring-kanban-agile-developments-new-rising-star/>
-