스크럼이란

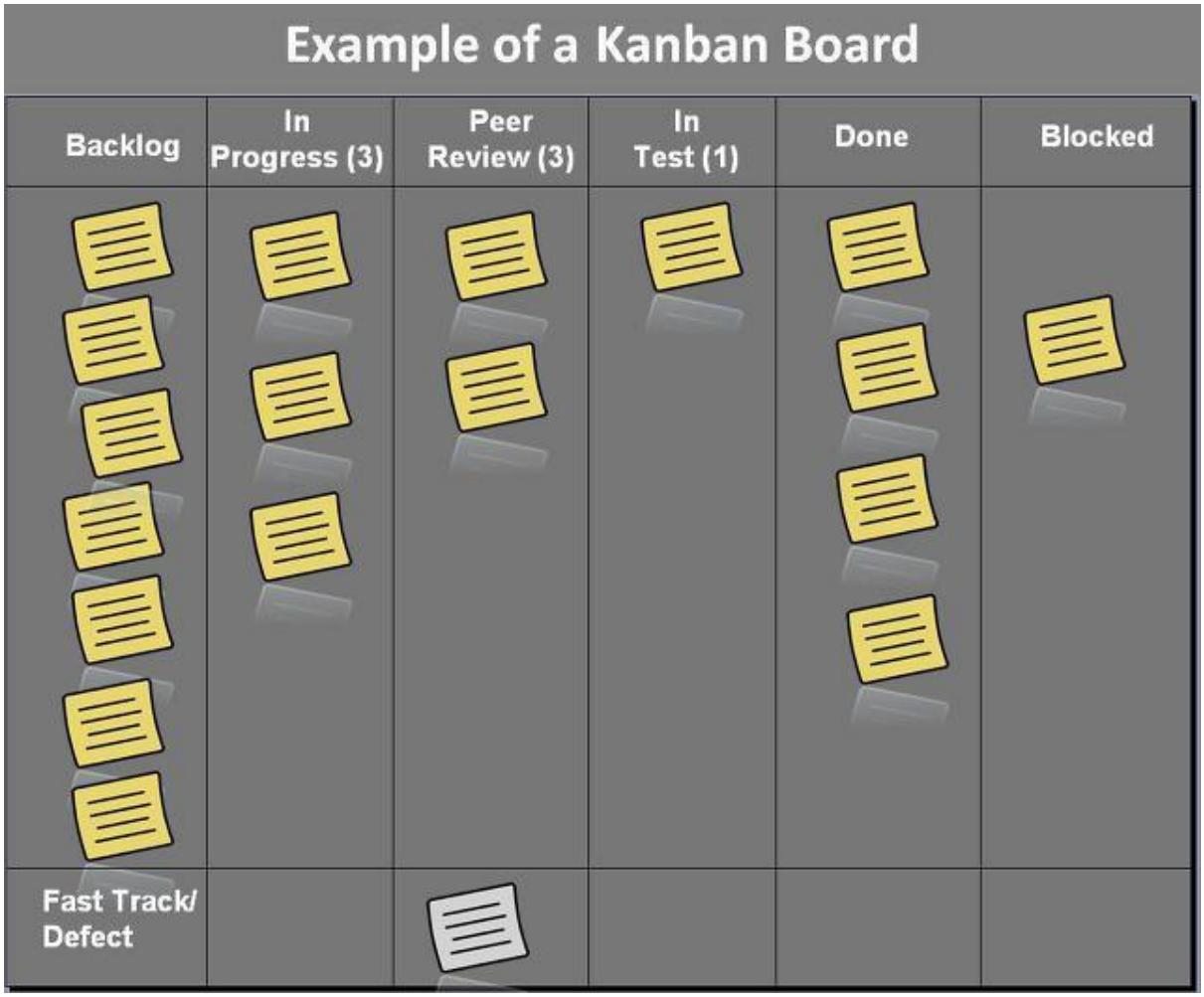
- 각 **이슈는 우선순위가** 존재한다. (Story Point, Backlog, Icebox 등이 있다.)
- **스프린트**를 통해 **실행 주기**를 조절한다. 실행 주기는 1 주 ~ 4 주로 다양하다. (각 팀의 실정에 맞춰 진행한다.)
- 매일 **스크럼 회의를 진행**한다. 서로의 작업을 파악하고, 어떤 작업을 언제 진행할지 파악하기 위함이다.
- 모든 작업의 할당은 **당김 방식**으로 진행한다.
- 스크럼 마스터를 둔다. 하지만 독단적인 역할을 수행하는 것이 아닌 구성원의 의견을 종합하는 역할과 목표 수립을 조절하는 역할을 한다.

스크럼의 단점

비현실적인 예측과 끊임없이 빈번하게 발생하는 틀린 추정들은 팀의 스프린트의 계획을 흐트릴 것이다.

회고는 우리가 한 실수들에 대한 비평으로 변할 수 있다.

틀린 추정, 우선순위와 스펙의 변경. 사업팀은 기술팀이 목표를 놓치고 있다고 느낄 것이고, 기술팀은 사업팀이 계속해서 목표를 바꾸고 있다고 느낄 것이다. 모두의 사기가 줄어든다.



칸반이란

- 소프트웨어에서의 칸반은 도요타의 TOC(Theory Of Constraints, 제약 이론)과 당김 방식(Pull Process)에서 착안하여 데이비드 앤더슨(David J.Anderson)이 주장한 방법론이다.
- 각 파이프 라인(Column, 열)은 **WIP(Work In Process)**를 가지고 있다. 어떤 파이프 라인의 최대 수용량은 WIP 로 결정된다.
- 작업의 중요성을 추정하는 대신 **동시에 처리할 수 있는 이슈의 수(WIP)를 제한**함으로써 제어할 수 있다.
- 모든 작업의 할당은 **당김 방식**으로 진행한다.
- 칸반의 핵심 특성 중 하나는 **조직적 피드백의 시행**(원문 : Implement Organizational Feedback)이다.

- IT 개발 조직에서 굳이 역할을 나누자면, TODO 파이프 라인을 기획자 혹은 PM 이 관리하고, 그 이후의 파이프 라인은 개발자가 관리한다.
- 데드라인이 존재하지 않는다.

왜 칸반을 쓰는가?

프로젝트에서 개발팀이 주어진 이슈를 들어오는 족족 처리하여 보내준다면 좋겠습니다만, 대부분 이슈가 과제목록에 쌓이게 되기 일쑤입니다.
칸반에서는 이러한 이슈를 언제까지 해야한다! 라는 정해진 기간은 없습니다만, 우선순위를 정하고 급한 순으로 progress 에 넣습니다.
즉, 과제 목록이 하나의 큐로 순서대로 들어가서 처리되는 것이죠.
칸반의 핵심은 대신 이 progress 에 넣을 수 있는 작업의 수, WIP(Work In Progress)에 제한을 두는것이 핵심입니다.
팀의 특성을 고려하여 WIP 의 수를 정하고, 그 이상의 작업을 동시에 하는 것을 제한하는 겁니다.

칸반의 장점

1. 개발자에게 데드라인이 정해져 있지 않고, 압박이 적어 부담을 덜 수 있게 하고, 너무나도 많은 이슈가 동시에 들어와 혼란을 주는 경우를 차단할 수 있습니다.
2. 칸반 보드를 보면 어떠한 작업이 누구에 의해, 어느정도 진행해졌는지 시각적으로 볼 수 있습니다.
3. 다음 작업을 위해 무엇이 필요하며, 현재 막히는 구간이 어디이기에 프로젝트가 더뎌지는 것인지 파악할 수 있습니다.
4. 여태까지 해온 작업과 해야 할 작업이 뚜렷하게 나뉘어있어, 개선 방안을 모색하기 편리합니다.
5. Hotfix(예상치 못한 문제)를 다루기 수월합니다. Hotfix 도 하나의 이슈카드로 처리되어 우선순위를 고려하여 프로젝트가 끝나기 위해 처리해야할 것으로 남겨놓을 수 있기 때문이죠.
6. 번아웃을 현저하게 감소 + 생산성 증가(optional)
7.모두가 칸반 보드만 보면 무엇이 누구에 의해, 얼마정도 진행되고 있는지, 다음 작업을 위해서 무엇이 필요한지, 병목이 어디인지 알 수 있기 때문에, 프로젝트 관리 오버헤드를 감소시켰다.

실제로 어떻게 사용되고 있는가?

각 팀을 위해 사무실 구석구석에 칸반 보드를 만들어서 운영하고 있으며,

각 칸반 운영방식은 팀마다 독자적으로 운영됩니다.

따라서, 같은 칸반 방식이라고 할지라도 각 행을 구성하는 수영 레인(프로세스)에 대한 정의가 각각 다르며, 이슈를 구성하는 카드의 구성도 모두 다릅니다.

어떤 팀은 앞서 예를 들어 to-do, doing, done 식의 형식으로 구성한 팀이 있는 반면, 어떤 팀은 사전조사, 계약 준비, 계약 성사, 계약 체결 등의 업무 베이스로 구성하는 팀도 있습니다.

카드의 색상에 대한 정의도 각각 다릅니다. 어떤 팀에서는 카드 색상으로 담당자를 표현하기도 하며, 다른 팀에서는 긴급도에 따른 이슈로 구분하기 위해 사용하기도 합니다.

활용 할 수 있는 것

WIP 를 통한 데드라인에서 벗어나 자유로운 개발환경 구성 및 직관적인 업무 프로세스 확인 가능

지속적인 개선

1) 우리는 프로세스가 개선될 것이라는 가설을 세우고,

2) 간접제어기(WIP 리밋같은)것을 조절하고,

3) 처리용량, 리드타임, 품질, 예측가능성등을 관찰한다. 그 결과를 바탕으로 결론을 도출하고,

4) 다른 것들을 변경하며 지속적으로 프로세스를 개선한다.

핵심요소는 아래와 같은 피드백루프이다.

무엇인가 변경한다 → 어떻게 되는지 관찰한다 → 학습한다. →다시 변경한다.

스크럼에서 스프린트가 기본 피드백루프이다.

칸반에서는 원하는 기간마다 피드백 가능

이것은 올바른 것을 만들고 있는가?에 대한 피드백 사이클이다.

<칸반의 실시간 지표>

- 평균리드타임 : 아이템이 완료에 도달할때마다 갱신한다.

- 병목지점 : X+1 칼럼이 비어있는데도 X 칼럼에 아이템이 잔뜩 들어있는 현상보드에서 공기방울 을 찾아보라

(공기방울 : 물속의 공기방울처럼 보드의 비어있는 칼럼을 뜻함.)

피드백루프의 너무길면, 프로세스 개선 속도가 늦어지고,

너무짧으면, 프로세스 안정화 기간이 부족하여 프로세스가 망가질 수 있다.

비교항목	스크럼	칸반
기간	기간이 고정된 이터레이션을 규정한다.	기간이 고정된 이터레이션은 선택사항.
약속	팀이 이터레이션에서 할 일의 양을 결정, 약속한다.	약속은 선택사항이다.
지표	계획하기와 공정 개선에 속도(velocity)를 기본 지표로 사용한다	리드 타임을 기본 지표로 사용한다.
팀	교차 기능 팀을 규정한다	교차 기능 팀은 선택사항, 전문가 팀도 허용한다.
아이템 크기	아이템들을 한 스프린트 안에 완료될 수 있을 정도의 크기로 잘게 쪼개야만 한다.	아이템 크기를 특별히 규정하지 않는다.
차트	번다운 차트를 규정한다.	차트 사용 규정 없다.
WIP	WIP 리밋을 간접적으로 한다.(스프린트마다)	WIP 리밋을 직접적으로 한다.(작업 흐름 단계마다)
추정	추정을 하도록 규정한다.	추정은 선택
아이템 추가	이터레이션이 진행 중일때는 아이템 추가 불가.	수용 능력이 허용한다면 새로운 아이템을 추가할 수 있다.
보드 소유	스프린트 백로그는 특정 팀이 소유한다.	칸반 보드는 다수의 팀이나 개인들 간에 공유하기도 한다.
역할	세가지 역할을 규정. (제품 책임자, 팀, 스크럼 마스터)	어떤 역할도 규정하지 않는다.
보드 초기화	이터레이션마다 스크럼 보드 초기화	칸반보드는 계속 유지
우선순위	제품 백로그에 우선순위를 정의할 것을 규정한다.	우선순위 정의하기는 선택 사항이다.

구분	Kanban	SCRUM
Time box 내의 iteration	선택적 적용	사전 정의 필수
작업량의 팀 내 승인	팀 승인 선택적	Iteration 내의 작업에 팀 승인 필수
계획과 프로세스 지표	리드 타임(lead time)사용	Velocity 사용
팀 구성	특정 분야의 전문가로 구성 가능	Cross functional team
작업 규모 사전 분할	규모에 대한 정의가 되지 않음	Sprint에 수행 가능한 수준
진척 확인 척도	대기행렬 및 칸반보드 (개발 대기, 테스트 대기, 배포 대기)	Bum down chart (x:작업량, y:반복시간)
WIP 제한	Workflow 상태 별 직접 제한	스프린트 단위로 간접적 제한
진행 중 작업 추가	여력이 있을 경우 진행 중 추가가능	진행중인 Iteration에 추가불가능
작업 항목공유	여러 팀이나 여러 개인에 공유	Sprint backlog 별 특정 팀에 소유
역할 지정	사전에 정의된 역할 없음	PO, SM, Team 등 사전에 지정
규칙	3개 - Workflow 시각화 (Visualize the workflow) - WIP(work-in-process) 제한 - flow 의 측정 및 최적화 (Measure and optimize flow)	9개 -스크럼 마스터(Scrum Master) -제품 책임자(Product Owner) -스크럼 팀(Team) -스프린트 계획미팅(Sprint planning meeting) -일일 스크럼(Daily Scrum) -스프린트 리뷰(Sprint review) -제품 백로그(Product backlog) -스프린트 백로그(Sprint backlog) -소멸 차트(Burndown chart)

구분	Kanban	SCRUM
진척관리	- 하나의 스토리가 한번의 반복 과정에서 완성(WIP) - total cycle time을 활용하여 성과 측정	- time box 구간 내에서 반복 수행(sprint) - bum down chart사용 팀 성과 측정
역할과 업무 협의	- 별도의 정의된 역할과 미팅이 없음 - 조직 내에서 익숙한 방법을 사용 - 기존 SCRUM사용 조직은 그대로 사용	- 특정 역할과 업무 협의(meeting)정의(스크럼마스터, 제품 책임자) - 스프린트 계획 미팅(스토리 내 스프린트 결정)
기존 방법론에서 이행	- 기존 프로세스와 함께 시작하면서 지속적 개선	- SCRUM 자체의 엄격한 체계와 기사용 중인 방법론에서 전환 어려움

보완대상	보완 상세 내용	기대효과
잦은 미팅의 오버헤드	비판적이고 잦은 Daily Scrum, 회고(Retrospective) 미팅으로 불필요한 시간소요를 최소화	회의 소요시간 감축 및 긍정적 에너지 집중
Story Point 추정의 불확실성	예상보다 큰 작업은 Sprint 에 오버헤드를 초래하므로, WIP 조정 및 프로세스 최적화로 불확실성 대응	프로세스 시각화, 업무예측 용이
Sprint 변경의 경직성	Sprint 주기 내에 변경을 허용불가를 WIP 를 조정으로 유연하게 변경을 수용	업무 프로세스 유연성 확보

결론 :

가능한한 조직을 작게 만든다. 3~4명이 팀이 되게 그리고 스크럼 마스터를 한명 지정한다.

그사람이 그 팀의 매니저역할을 한다. (Scrum)

유저스토리를 기반으로 만들어야 구현해야할 기능을 Feature로 구분한다. (XP -> 빼도 됨)

팀단위로 구현할 Feature을 칸반 보드에서 가져간다. (회사 전체 업무를 정리한 칸반 보드 1개, 각 팀마다 칸반 보드 1개씩 존재) (kanban)

팀마다 상세하게 Backlog를 작성하고 우선순위를 부여한다. (Scrum)

팀 마다 각각 칸반보드를 활용하여 sprint를 진행한다. (kanban)

2~3일마다 스크럼 회의를 각 팀마다 자율적으로 하고 일주일에 한번은 팀매니저끼리 모여서 feature에 대한 진행도 회의를 한다. (Scrum)