



소프트웨어 개발

스크럼과 JIRA 기반의 소프트웨어 개발 프로세스

#목표

“애자일 스크럼과 협업 도구인 JIRA를 이용한 소프트웨어 개발 프로세스를 이해한다.”

스크럼 개발 방법론 요약

개발 방법론의 변화



그래서...

저명한 박사님들께서.. 방법론을 만드시니..
RUP,CBD,CMMI

개발 방법론의 변화

But...

그러나 현실은

- 방법론은 방법론...
- 우리는 조금 더 현실적인 방법론이 필요합니다.

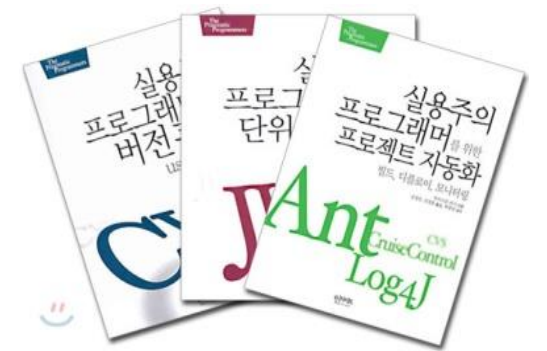
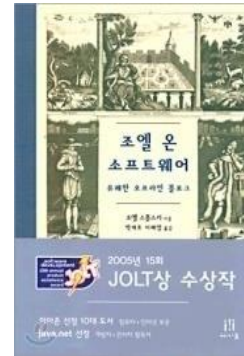


개발 방법론의 변화



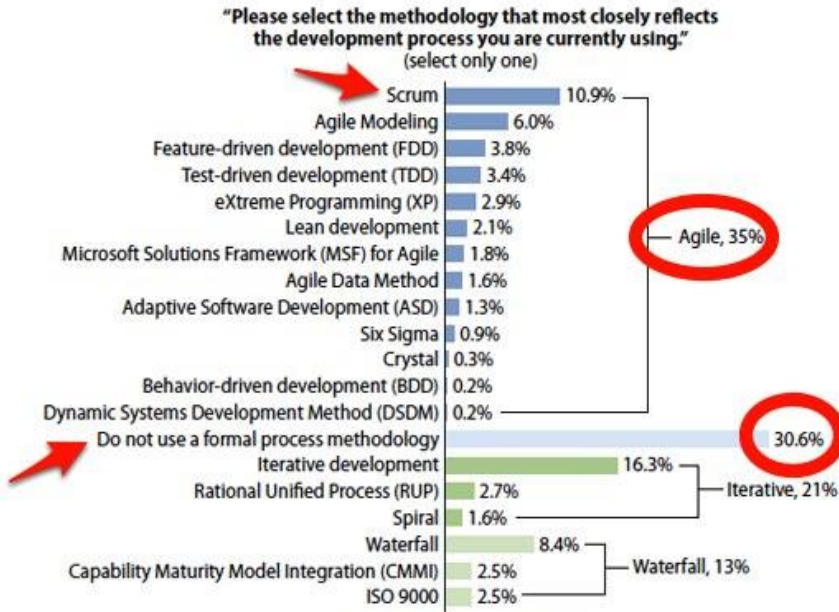
실용주의 방법론
구세주 등장!!

- 실용주의 방법론
 - Erich Gamma, Joel Spolsky, Kent Beck, Andrew Hunt
 - Iterative & Incremental
 - 애자일 기반
- 기존 방법론과 차이
 - 요구 사항이 변화할 것을 가정
 - 에러가 있을 것을 가정하여, 자주 테스트
 - 협업과 커뮤니케이션



개발 방법론의 변화

Figure 1 Agile Is Organizations' Primary Development Approach

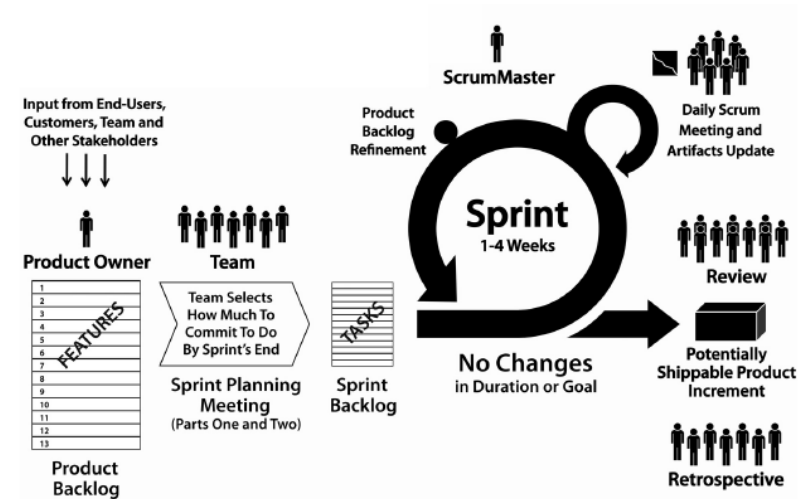


Base: 1,298 IT professionals

Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

56100

Source: Forrester Research, Inc.

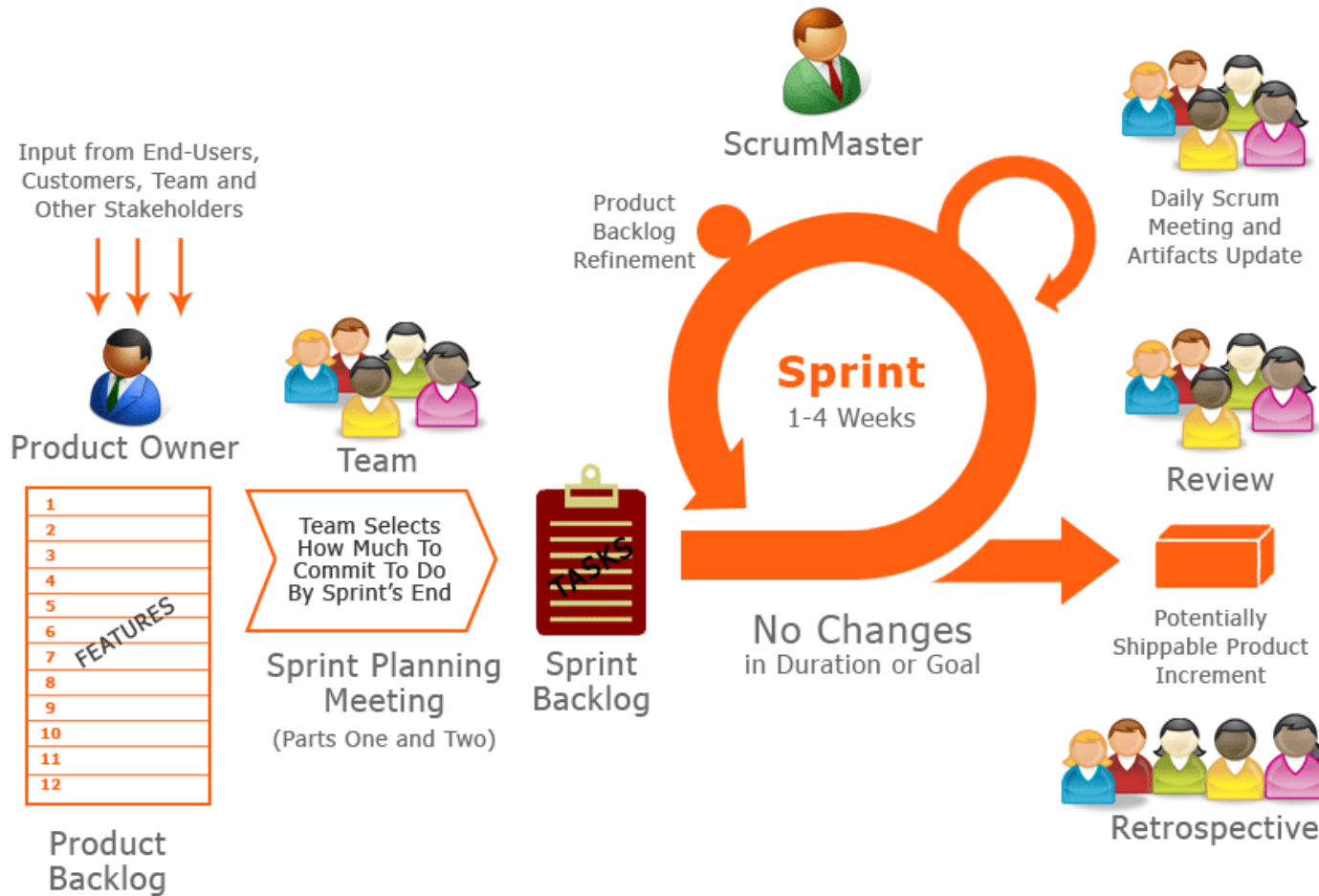


스크럼이 대세!!

관리자 입장에서는 예측 불가
조직에 맞게 바꿔서 쓰세요

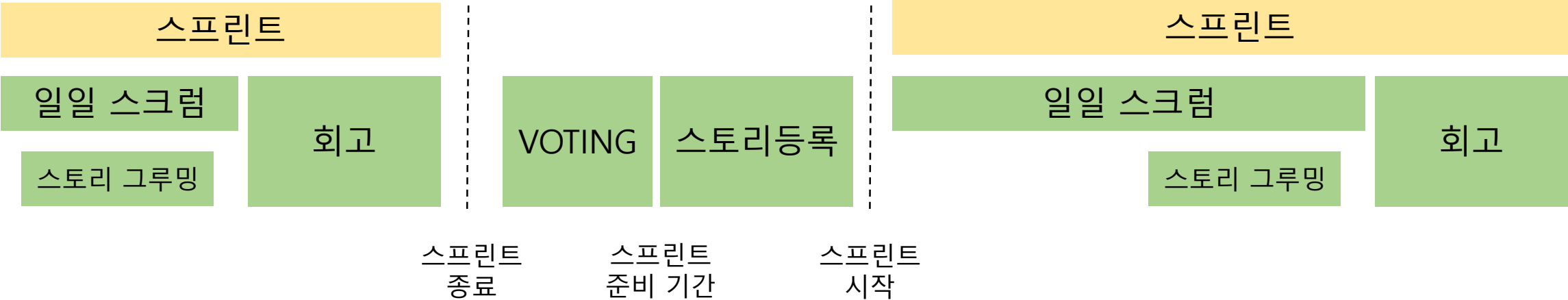
<http://agilescout.com/learn-more-agile-software-development-methods-this-year/>

스크럼 프로세스 개요



- Product BackLog
- Sprint Meeting
- Sprint BackLog
- Sprint
- Review
- Restrospective

스크럼 프로세스



조직 구조

- 2 피자 팀 (5~7명)
- Cross functional team (기획 + 개발 + 디자인 + QA + 운영)
- Self Organized team (스스로 결정하고 진행)
- 스크럼 마스터, 프로젝트 오너 (PO)
- 애자일 코치가 있으면 효과가 높음 (EX. OO슬라이드)

스크럼 프로세스

1. 그루밍

- "다음 스프린트에 들어가기전에, PO가 다음 스프린트에 개발할 기능에 대해서 대략적으로 리뷰를 하는 행위". 스프린트 진행중 일어남
- WHY
 - 다음 스프린트에 대한 가시성 확보 (미리 생각할 시간을 준다)
 - 개발 개시전 리뷰를 통해서 개발 가능성, 기획상 구멍등을 찾아서 수정할 시간을 갖는다.
- 어떻게 ?
 - 1시간 정도로, 사용자 스토리나 UX 프로토타입을 리뷰
 - 가급적 실제로 돌아가는 UX 목업이 좋음 (애니메이션 복잡도등을 미리 예측 가능)

2. 사용자 스토리 작성

• 사용자 스토리

: 주요 기능을 사용자 관점에서 기술
 "As a {user}, I want do {something}".
 "나는 {사용자로서}, {무엇}을 한다".

• 특징

- 직관적으로 이해가 되어야 하며
- 테스트가 가능해야 함

• 분류, 스토리, 설명 형태로 기술하는 것이 좋음

스토리 진행 순서

Feature (Lv1)	Feature (Lv 2)	Detail
서비스 가입 및 로그인	페이스북 사용자는 SSA6 에 가입할 수 있다.	- 페이스북에 가입한 사용자는 페이스북을 통하여 SSA6 그룹에 가입할 수 있다.
	페이스북 사용자는 Yurry 서비스에 로그인 할 수 있다.	- 모든 페이스북 사용자는 페이스북 계정을 이용하여 Yurry 서비스에 로그인할 수 있다. - 글을 쓰는 권한은 SSA6 멤버만이 할 수 있다. - 서비스에서 SSA6의 페이스북 페이지로 링크를 제공하여, 일반 페이스북 회원이 SSA6에 가입할 수 있는 경로를 제공한다.
글 읽기	로그인한 사용자는, 포스팅 글과 댓글을 읽을 수 있다.	- 로그인한 사용자는 누구나 포스팅을 읽을 수 있다. - 포스팅한 글에는 글을 올린 사람의 사진,이름,포스팅 일정, 좋아요 카운트를 포스팅 글과 함께 출력한다. - 댓글에는 글을 올린 사람의 사진,이름,포스팅 일정, 좋아요 카운트를 포스팅 글과 함께 출력한다.
	로그인한 사용자는, 포스팅 글에 첨부된 그림을 볼 수 있다.	- 포스팅한 글은 최신 순서에 따라서 정렬 된다 - 포스팅 글에 댓글이 있을 경우, 댓글이 함께 포함 되어 출력된다. - 포스팅한 글에 첨부 이미지가 있을 경우, 썸네일 형태로 해당 이미지를 출력하여 준다. - 사용자가 썸네일을 출력하였을 경우, 원래 크기의 이미지로 확대해서 보여준다.
글 쓰기	로그인한 사용자중 SSA6 멤버만이, 포스	- 포스팅시에 이미지를 첨부할 수 있다.
	로그인한 사용자중 SSA6 멤버만이, 포스	- 댓글을 입력할 수 있는 창이 포스팅글 아래, 댓글들의 맨 아래 나타나고, 이를 통해서 사용자는 댓글을 남길 수 있다.
좋아요	로그인한 사용자중 SSA6 멤버는, 포스팅	
	로그인한 사용자중 SSA6 멤버는, 댓글에	
검색	로그인한 사용자는, 검색어를 사용하여,	
	포스팅이나 댓글을 포함하여 검색을 할 수	
사용자 랭킹	로그인한 사용자는, SSA6의 활동이 활발한 사용자 들을 출력해준다.	- 사용자들의 활동을 유도하기 위해서, 사용자들의 활동 내용을 점수로 계산하여 상위 랭커를 페이지에 출력한다. - 포스팅글은 5점, 댓글은 1점으로 계산하여, 당일 대비 1개월간의 점수를 총 합하여 사용자들의 랭킹을 메기고, 상위 랭커 5명의 이름과 사진, 점수를 화면에 할당 출력해준다
페이스북 연동	모든 포스팅과 댓글을 페이스북과 동기화 된다.	- Yurry 서비스에서 작성한 포스팅,댓글,좋아요는 페이스북 SSA6 페이지에 실시간으로 반영된다. - 반대로 페이스북 SSA6 페이지에 업데이트 내용은 Yurry 서비스에 실시간으로 반영된다.

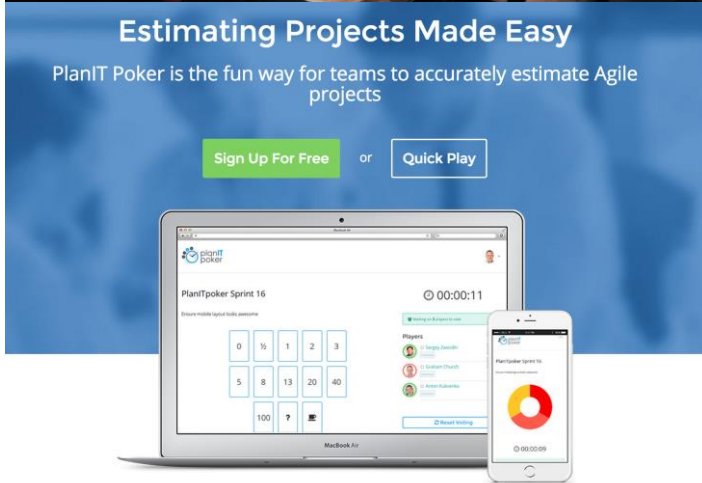
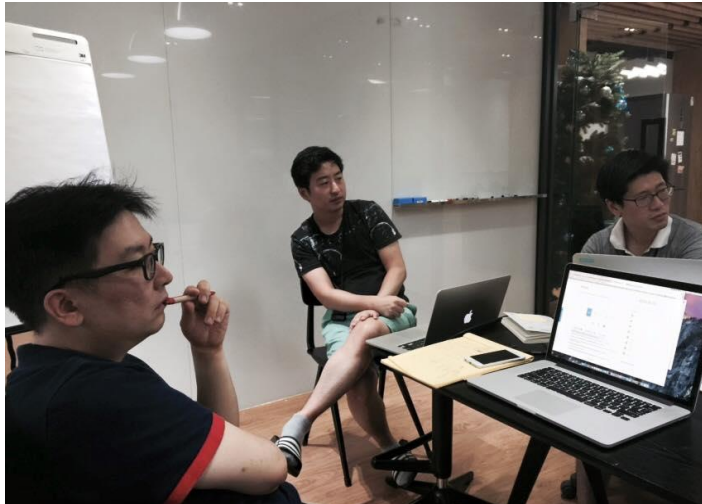
기능 분류

사용자 기능

상세 설명

3. 스토리 보팅

- 플래닝포커 (PLANNING POKER)



plantit.com

- 개발팀 전체가 모여서, 각각의 사용자 스토리에 대해서 개발 기간을 투표
 - 포인트의 의미는 알아서 정함. (EX 1포인트 = 1일)
 - 포인트는 0.5,1,2,3,5 단위로 띄워서 정의
 - 전체 팀원이 서로 설득 당할때 까지 계속 진행
 - 사용자 스토리에 대한 디테일은 PO에게 그 자리에서 질의
 - 점수가 가장 높은 사람과, 가장 낮은 사람의 의견을 듣고 다시 보팅.
- 가장 낮은 사람 (빠른 개발 방식을 알고 있을 수 있음), 가장 높은 사람 (놓친 무언가를 알 수 있음)
- 정확하지 않으나, 정확함

4. 개발 범위(일정 지정)

- 산출된 포인트로 개발 일정 지정 (2~4주)
- 버퍼 (회의, 문서, 배포, 오류 수정 등) 배정 필요 (1.5~2배)
- 4주가 넘는 일정은 우선순위에 따라 자름

5. 스프린트 시작

- JIRA에 등록
- 요건 UX 확인은 JIRA 로 핑퐁, UX 시안도 JIRA로 전달
- JIRA ISSUE 구조 정의

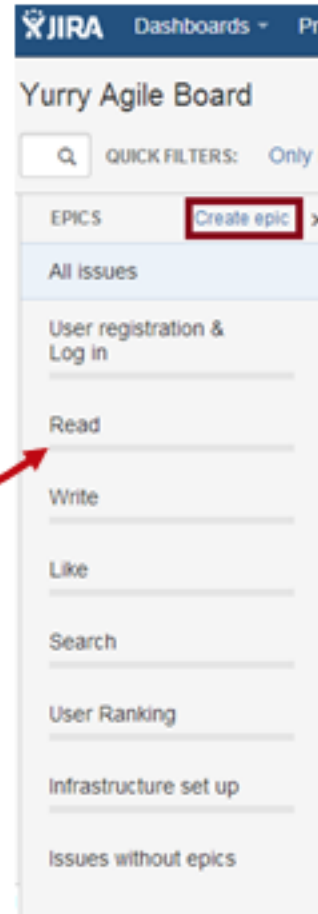
5. 스프린트 시작

- JIRA의 이슈 타입
 - EPIC : 스토리의 집합
 - STORY : 사용자 중심의 기능 목록
 - CHORE : 사용자 기능 중심이 아닌, 기술적인 태스크
 - ISSUE : 메니저가 관리하는 이슈
 - BUG : 버그
 - SUB-TASK : 태스크들의 서브 태스크

5. 스프린트 시작

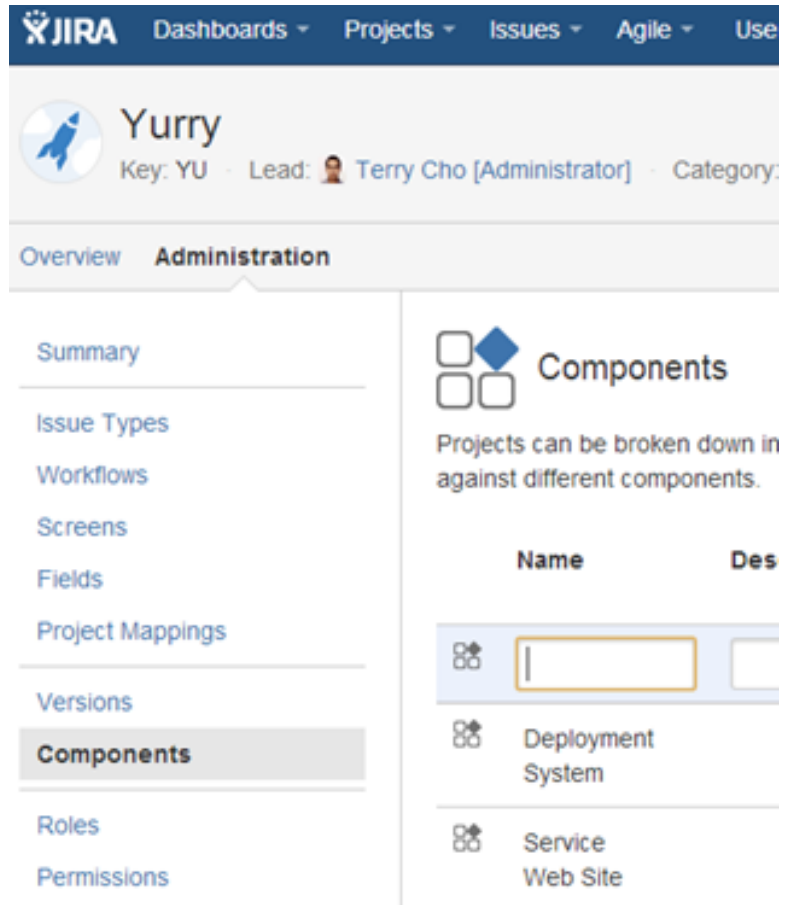
- EPIC 등록

Feature (Lv1)	Feature (Lv 2)	
서비스 가입 및 로그인	페이스북 사용자는 SS46 에 가입할 수 있다. 페이스북 사용자는 Yurry 서비스에 로그인할 수 있다.	스프린트
글 읽기	로그인한 사용자는, 포스팅 글과 댓글을 읽을 수 있다. 로그인한 사용자는, 포스팅 글에 첨부된 그림을 볼 수 있다.	스프린트
글 쓰기	로그인한 사용자중에 SS46 멤버만이, 포스팅글에 댓글을 달 수 있다.	스프린트
좋아요	로그인한 사용자중에 SS46 멤버는, 포스팅글에 "좋아요"를 선택할 수 있다.	스프린트



5. 스프린트 시작

- 이슈 등록 / 컴포넌트 정의



- 컴포넌트는 제품 단위 또는 팀 단위로 분리하는 것이 좋음
- EX) IOS, 안드로이드, 웹
- EX) 관리자 포탈

5. 스프린트 시작

- 이슈 등록 / Priority (우선순위)
 - Blocker의 경우 해결되지 않으면 프로젝트가 진행될 수 없는 경우
 - Critical은 프로젝트 진행은 가능하나 해결하지 않으면 정상적인 서비스 개발이 어려운 경우
 - Major 꼭 개발해야 하는 경우
 - Minor 개발은 해야 하나 없어도 상관 없는 경우
 - Trivial 있으나 없으나 크게 상관 없는 경우

5. 스프린트 시작

- 이슈 등록

Project **Yurry**

Issue Type **Story**

Summary*

Priority **Major**

Due Date

Component/s

Start typing to get a list of possible matches or press down to select.

Affects Version/s

Start typing to get a list of possible matches or press down to select.

Fix Version/s

Start typing to get a list of possible matches or press down to select.

Assignee **Terry Cho [Administrator]** [Assign to me](#)

Reporter* **Terry Cho [Administrator]**

Start typing to get a list of possible matches.

Environment

For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).

Description

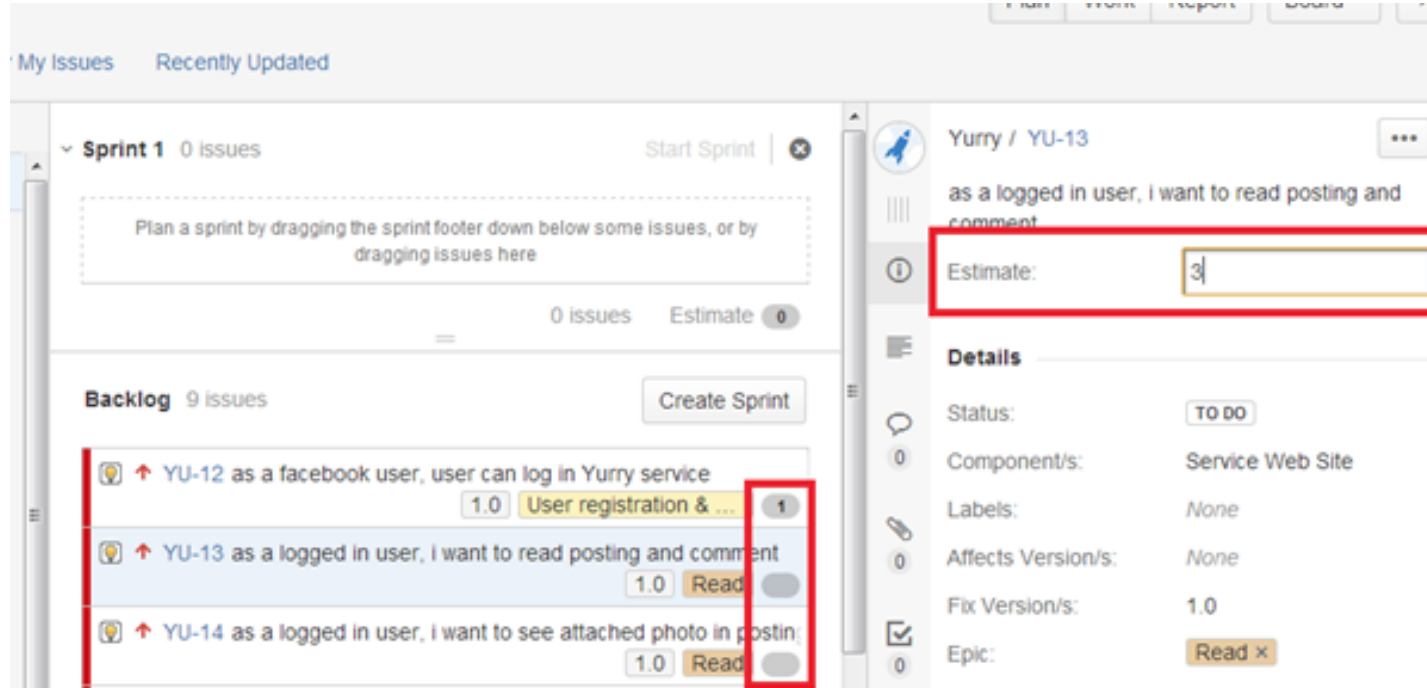
- all of face book user can log in Yurry service with their own facebook id
- Only SSAG group member can have permission to write posting & comment.
- Facebook SSAG group link is provided to facebook user can join the SSAG

- Summary : 제목 입력
- Priority : 우선 순위 생성
- Issue Type : 보통은 Story로
- Assignee : 개발 담당 (또는 스크럼 마스터로 정해 놓고, 나중에 ASSIGN하거나, 각 개발자가 알아서 당겨가는 PULLING 방식)
- Description : Issue에 대한 상세 내용

* 엑셀로 벌크 업로드 가능

5. 스프린트 시작

- 이슈 등록 / 스토리 포인트 부여



- 스토리 보팅 단계에서 산출한 스토리 포인트를 스토리별로 부여

5. 스프린트 시작

- 이슈 등록 / 스프린트 맵핑

The screenshot shows the Jira interface with a Sprint and Backlog view. The Sprint is titled 'Sprint 1 (Dec.16~Dec.27)' and contains 4 issues. The Backlog contains 5 issues. A 'Create Sprint' button is highlighted with a red box.

Sprint 1 (Dec.16~Dec.27) 4 issues Start Sprint

Issue ID	Description	Estimate	Type	Count
YU-12	as a facebook user, user can log in Yurry service	1.0	User registration & ...	1
YU-13	as a logged in user, i want to read posting and comment	1.0	Read	3
YU-15	as a SSAG user, i want to write posting	1.0	Write	2
YU-16	as a SSAG user, i want to write comment on posting	1.0	Write	1

4 issues Estimate 7

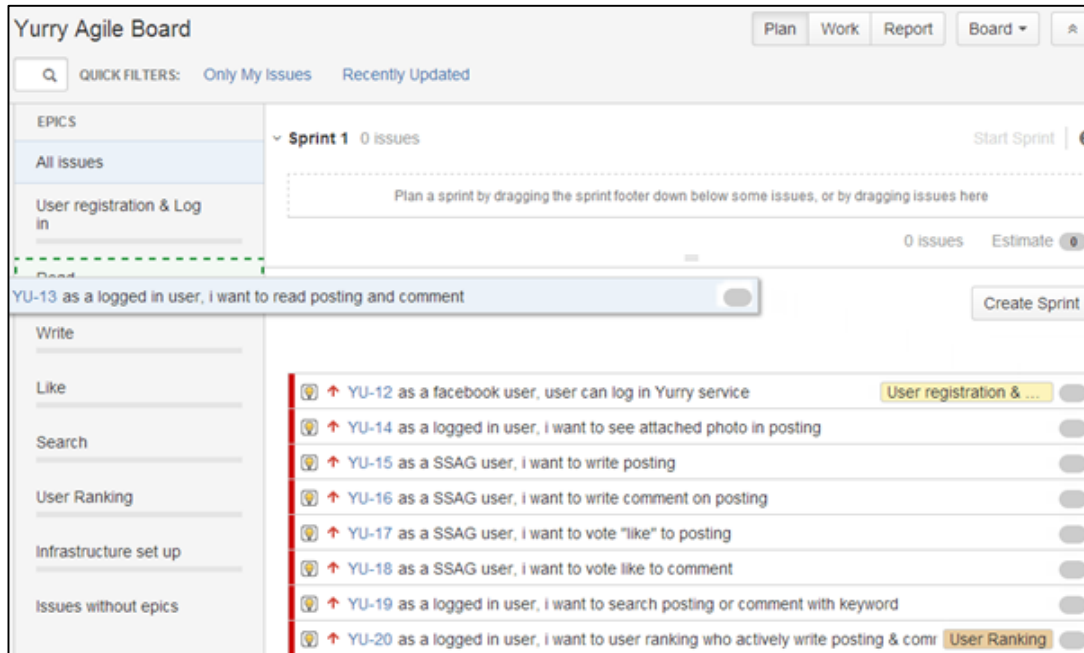
Backlog 5 issues Create Sprint

Issue ID	Description	Estimate	Type	Count
YU-14	as a logged in user, i want to see attached photo in posting	1.0	Read	2
YU-17	as a SSAG user, i want to vote "like" to posting	1.0	Like	0.5

- 스프린트를 생성하고, 백로그에서 스프린트를 끌어 올려서, 스프린트에 맵핑

5. 스프린트 시작

- 이슈 등록 / EPIC 맵핑



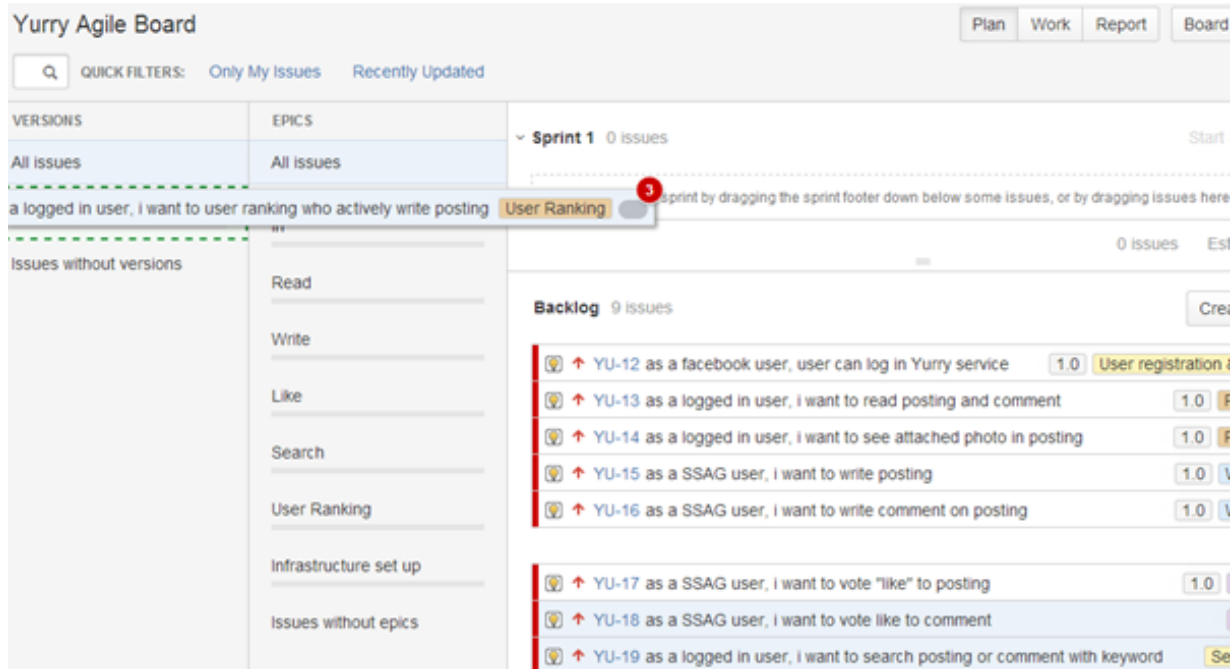
- 등록된 이슈들을 EPIC에 맵핑
- 처음 스토리 작성시 그룹핑을 그대로 사용

5. 스프린트 시작

- 이슈 등록 / 릴리즈 버전 맵핑

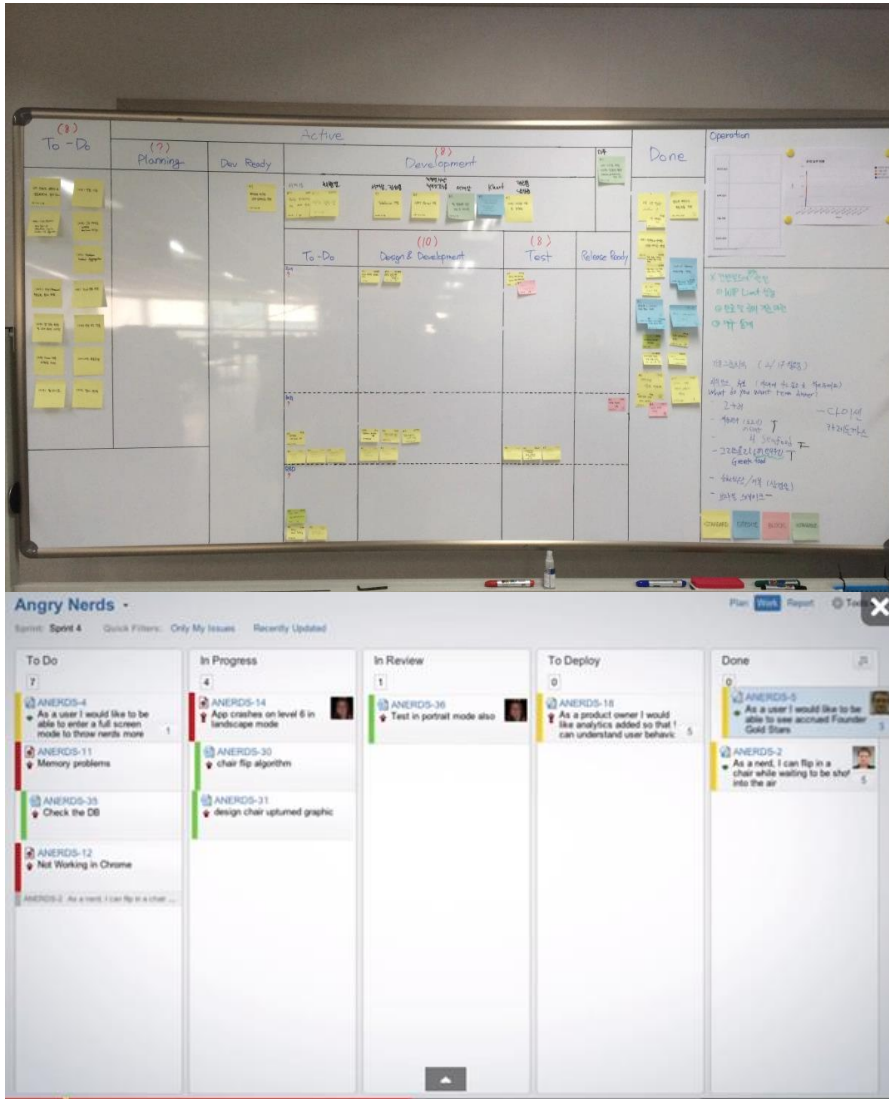


- 릴리즈 버전을 정의해놓고, 버전에 스토리를 맵핑
- 어느 기능이 어느 버전에 들어 가는지 추적 가능
- 어느 버그가 어느 버전에서 발생해서 어느 버전에서 수정이 가능한지 추적 가능



5. 스프린트 시작

- 칸반 보드 세팅



- 진행중인 스크럼 태스크의 상태를 칠판에 표시 해놓는 방식 (포스트잇)
- JIRA 애자일 보드로도 가능하지만, 가시성이 떨어짐. (태스크 & 서브 태스크들이 많아서 잘 안보게됨. 또는 자기것만 보게됨) → JIRA를 쓰더라도 칸반보드는 별도로 운영하는게 좋음
- 칸반 보드의 스타일은 팀과 개발 방식에 맞게 끊임없이 변화해야 함

6. 스프린트 진행

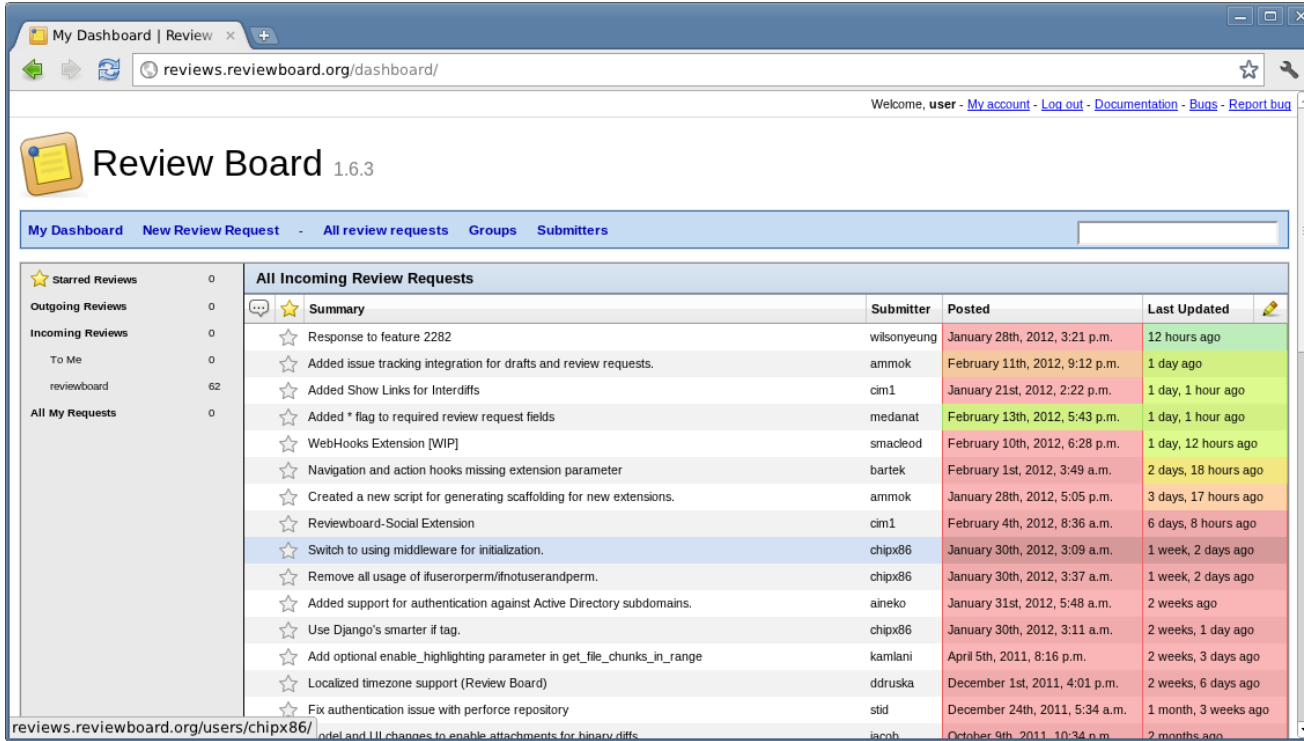
- 일일 스크럼 미팅



- 약속한 시간에 스탠드업 미팅
- 칸반 보드를 보고
- "어제 한일, 오늘 할일, 일처리에 장애 요인" 을 간단하게 이야기 함.
- 끼어들거나 질문 금지... (마이크를 드는것도 좋음)
- 추가 회의가 필요하면 스크럼 미팅 끝나고
- 생각보다 잘 안됨. (회의는 하지만 컨텍스트 공유가 잘 안됨) → 될때까지...

6. 스프린트 진행

- 코드 리뷰



The screenshot shows the Review Board 1.6.3 web interface. The browser address bar displays `reviews.reviewboard.org/dashboard/`. The page header includes a welcome message for a user and links for [My account](#), [Log out](#), [Documentation](#), [Bugs](#), and [Report bug](#). The main navigation bar contains links for [My Dashboard](#), [New Review Request](#), [All review requests](#), [Groups](#), and [Submitters](#). On the left sidebar, there are sections for [Starred Reviews](#) (0), [Outgoing Reviews](#) (0), [Incoming Reviews](#) (0), [To Me](#) (0), [reviewboard](#) (62), and [All My Requests](#) (0). The main content area is titled **All Incoming Review Requests** and contains a table with the following data:

Summary	Submitter	Posted	Last Updated
☆ Response to feature 2282	wilsonyeung	January 28th, 2012, 3:21 p.m.	12 hours ago
☆ Added issue tracking integration for drafts and review requests.	ammok	February 11th, 2012, 9:12 p.m.	1 day ago
☆ Added Show Links for Interdiffs	cim1	January 21st, 2012, 2:22 p.m.	1 day, 1 hour ago
☆ Added * flag to required review request fields	medanat	February 13th, 2012, 5:43 p.m.	1 day, 1 hour ago
☆ WebHooks Extension [WIP]	smacleod	February 10th, 2012, 6:28 p.m.	1 day, 12 hours ago
☆ Navigation and action hooks missing extension parameter	bartek	February 1st, 2012, 3:49 a.m.	2 days, 18 hours ago
☆ Created a new script for generating scaffolding for new extensions.	ammok	January 28th, 2012, 5:05 p.m.	3 days, 17 hours ago
☆ Reviewboard-Social Extension	cim1	February 4th, 2012, 8:36 a.m.	6 days, 8 hours ago
☆ Switch to using middleware for initialization.	chipx86	January 30th, 2012, 3:09 a.m.	1 week, 2 days ago
☆ Remove all usage of ifuserorperm/!notuserandperm.	chipx86	January 30th, 2012, 3:37 a.m.	1 week, 2 days ago
☆ Added support for authentication against Active Directory subdomains.	aineko	January 31st, 2012, 5:48 a.m.	2 weeks ago
☆ Use Django's smarter if tag.	chipx86	January 30th, 2012, 3:11 a.m.	2 weeks, 1 day ago
☆ Add optional enable_highlighting parameter in get_file_chunks_in_range	kamilani	April 5th, 2011, 8:16 p.m.	2 weeks, 3 days ago
☆ Localized timezone support (Review Board)	ddruska	December 1st, 2011, 4:01 p.m.	2 weeks, 6 days ago
☆ Fix authentication issue with perforce repository	stid	December 24th, 2011, 5:34 a.m.	1 month, 3 weeks ago
☆ Model and UI changes to enable attachments for binary diffs	jacob	October 9th, 2011, 10:34 p.m.	2 months ago

[reviewboard]

- 코드 리뷰는 피어 리뷰 방식으로 진행
- 문제 발생시 피어와 커미터 양쪽 공동 책임
- Reviewboard, gitlab, ...

6. 스프린트 진행

- DAILY COMMIT (with TICKET #)

All	Comments	Change History	P4 Changes	Subversion Commits	
Revision	Date	User	Message		
#11	Fri Nov 12 17:30:39 EST 2004	Mike	Big, very exciting commit for TEST-3! Files Changed ADD trunk/moved.txt (from /trunk/copieddocument.txt #10) DEL trunk/copieddocument.txt ADD trunk/NewFile.java DEL trunk/mydocument.txt		
Revision	Date	User	Message		
#10	Thu Nov 11 18:12:59 EST 2004	Mike	Fixed TEST-3 Files Changed MODIFY trunk/mydocument.txt		

- 매일 코드 커밋
- 코드 커밋시에 JIRA TICKET #를 적어 놓으면, JIRA 티켓에서 코드 변경 사항 추적이 용이함.

6. 스프린트 진행

- JIRA 티켓 업데이트
 - 보통 여기서 망함
 - 잘못된 사용예
 - 티켓에 아무 내용이 없음
 - 상태를 업데이트 하지 않음 (나중에 몰아 닫기)
 - 옳은 사용예
 - 티켓을 통해서 요구사항에 질문이 있을때, PO에게 다시 지정, UX가 필요하면 UX 디자이너에게 지정 (또는 서브 태스크 사용)
 - 나한테 ASSIGN 된 티켓은 항상 업데이트
- 스크럼 코치가 있으면 효과적 (티켓 업데이트를 종용)

7. QA

- 근래에 QA 방식은 다양화됨
 - 개발팀이 직접 QA 하는 방식 (별도의 QA를 두지 않음)
: 이상적.빠셈. 다양한 단말 테스트가 쉽지 않음
 - 개발팀내에 QA를 뒀서 스토리가 개발 완료되는 데로 QA
: 막판에 몰리는 현상이 발생함
 - 개발팀 외에 QA를 두고, 개발 종료 시점에 QA를 하는 방식
: 당근 막판에 몰리고, QA팀 일정 조정이 어려움.

8. 데모

9. 릴리즈 및 배포

- 코드 프리즈 및 배포
- 앱 릴리즈
 - 안드로이드 단계적 릴리즈 (10%, 30%, ... 100%)
 - 앱 크래쉬 리포팅 (Twitter CrashtyCS)

10. 회고

- 잘한것, 고마운것,잘못한것, 아쉬운점을 정리해서 다음 스프린트에 개선 방향을 정하고 세부 액션 아이템을 정해서 수행
 - 진행자가 있는 것이 좋음
 - 포스트잇으로 잘한것,고마운것, 잘못된것, 아쉬운점을 각자 적어서 붙여놓고 그룹핑
 - 잘한것을 더 잘하게, 미진했던 점을 극복할 수 있는 방안을 써놓고 그룹핑
 - 투표를 통해서 다음 기간에 실행에 옮길 아이템을 선정
 - 다음 회고 프로세스를 개선할 수 있는 방안을 추가로 기술
- 일반적으로 회고때 푸념만 이야기 하고 구체적으로 실행되지 않는 경우가 많음
- 액션 아이템에 대한 오너(OWNER)를 정하는 것이 좋음

10. 회고

- 총 3시간 프레임워크
 - 스프린트에 대한 감상 공유 (10)
 - ESVP (10)
 - 좋았던점, 나빴던점 모으기 (20) : 포스트잇 3장
 - 잘된점,나빴던점,고칠내용,불확정 (30)
 - 화남,기쁨,슬픔 (20) : 포스트잇 3장씩
 - 팀워크 만족도 (20) : LV1~5
 - 점스티커 투표(10) : 인당 스티커 한장씩, 도움이 된거 하나, 방해된거 하나
 - SMART 회의 (20) : 4명씩
 - EXACT 목표 (20) : 4인
 - 회고에 대한 회고 (10)

백로그 회의

백로그 회의

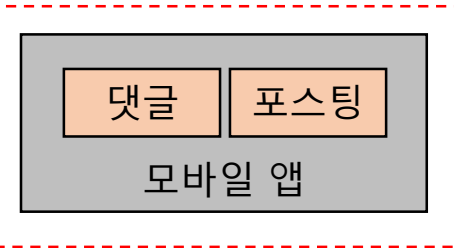
- 목적과 의미
 - 3개월 이상의 장기 제품 개발 로드맵을 정의
- 수행 방법
 - 주요 경영진과, 개발리드가 참여
 - 매주 또는 격주에 한번
 - 우선 순위 조정 및, 주요 기능에 대한 사전 리뷰
 - 에픽 단위의 기능을 정의하고, 우선순위를 비즈니스 상황에 따라서 변동
- 운영
 - JIRA에 백로그 프로젝트를 별도 구성
 - 위아래 순서로 우선 순위 지정
 - 릴리즈 버전을 부여함으로써 릴리즈에 대한 대략적인 일정 예측

애자일 팀 모델의 변화

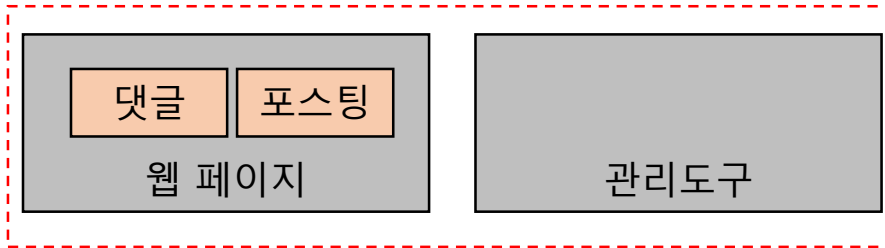
팀모델의 변화

- 기술 분류에 따른 팀 (Functional Team)
 - 프론트, 백엔드, 앱, 운영, QA 등

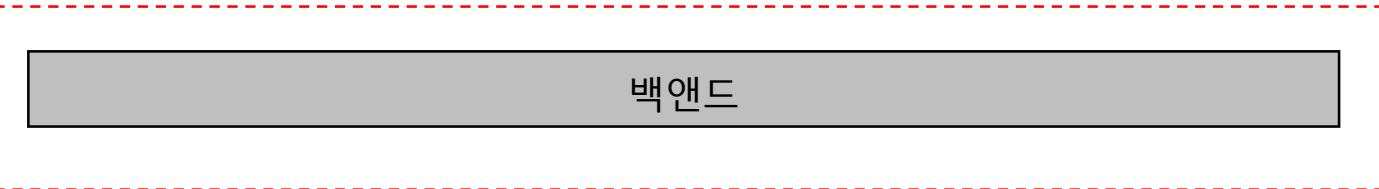
앱개발팀



프론트 개발팀

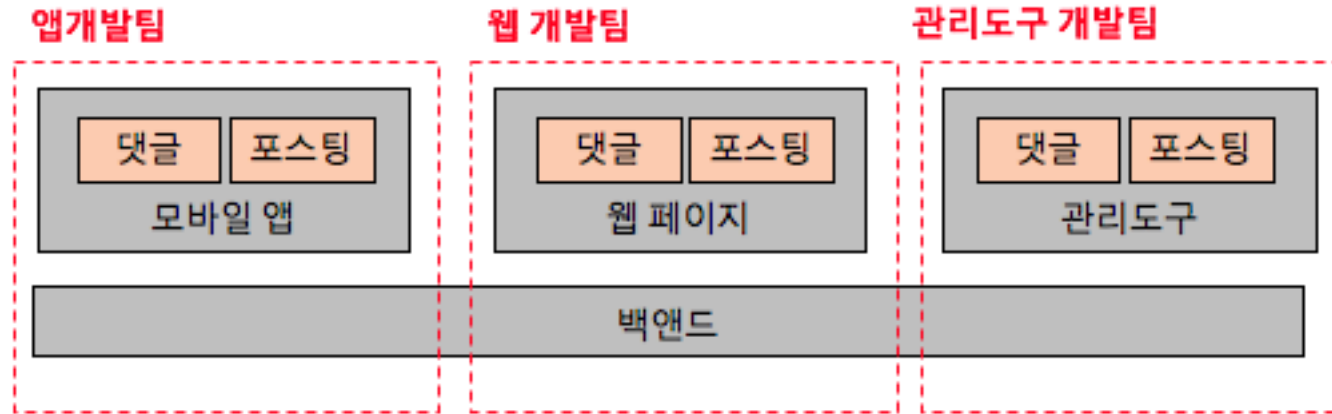


백엔드 개발팀



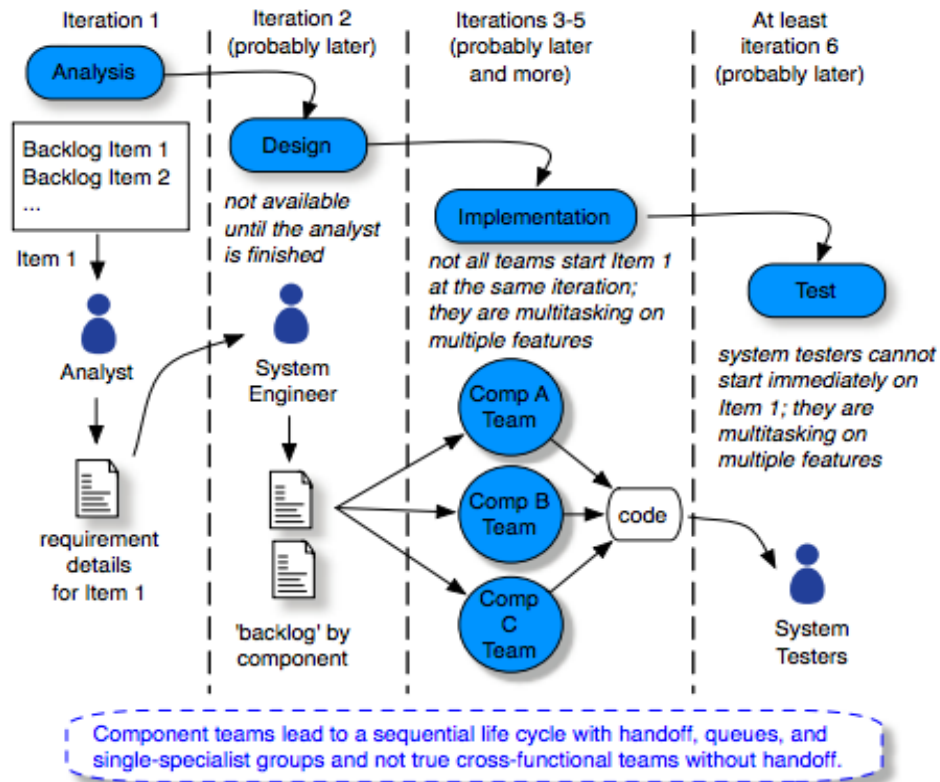
팀모델의 변화

- Cross functional Team (Product based)
 - 한팀내에 프론트,백엔드,앱,운영,QA를 다 넣음
 - MSA에 적용되-



팀모델의 변화

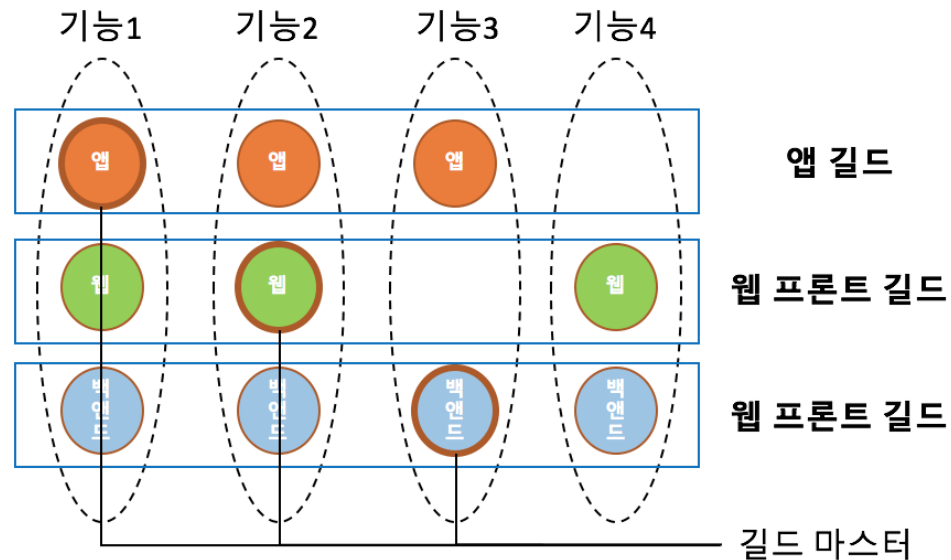
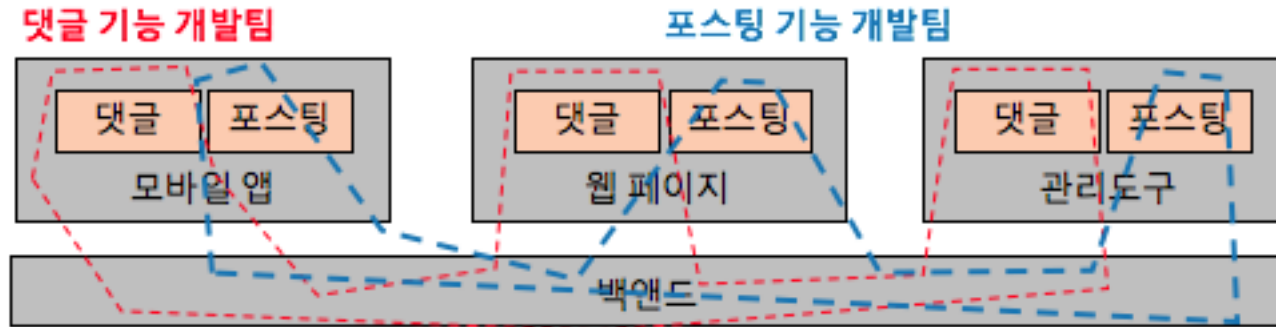
- Cross functional Team (Product based)
 - Product 간에 복잡한 코디네이션이 필요함
 - Program manager,



팀모델의 변화

- Cross functional team (Feature based)

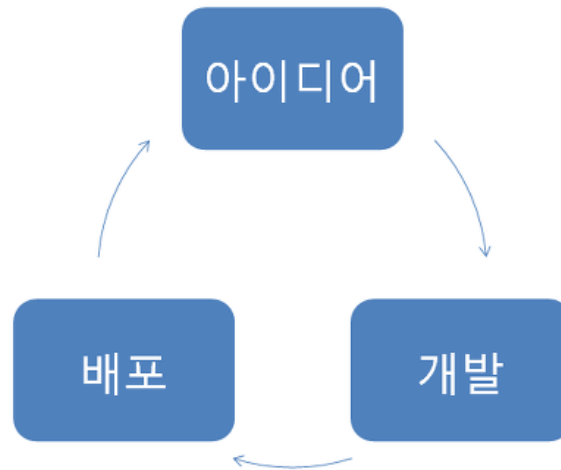
- 기능 단위로 팀을 묶고 하팀이 전체 컴포넌트를 개발
- 플랫폼화 필수



개발 운영 프로세스의 변화

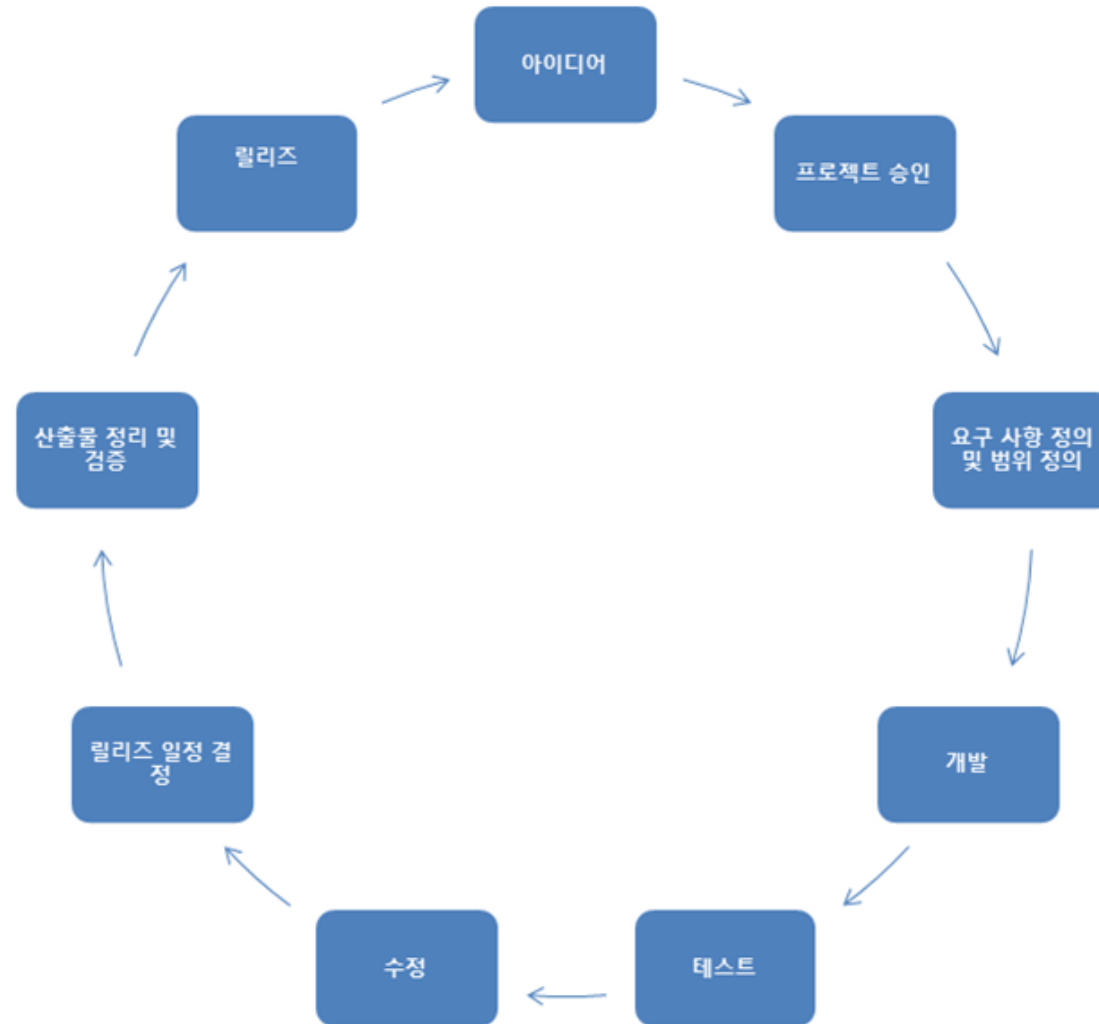
개발 운영팀의 변화 모델

- 1단계. 스타트업



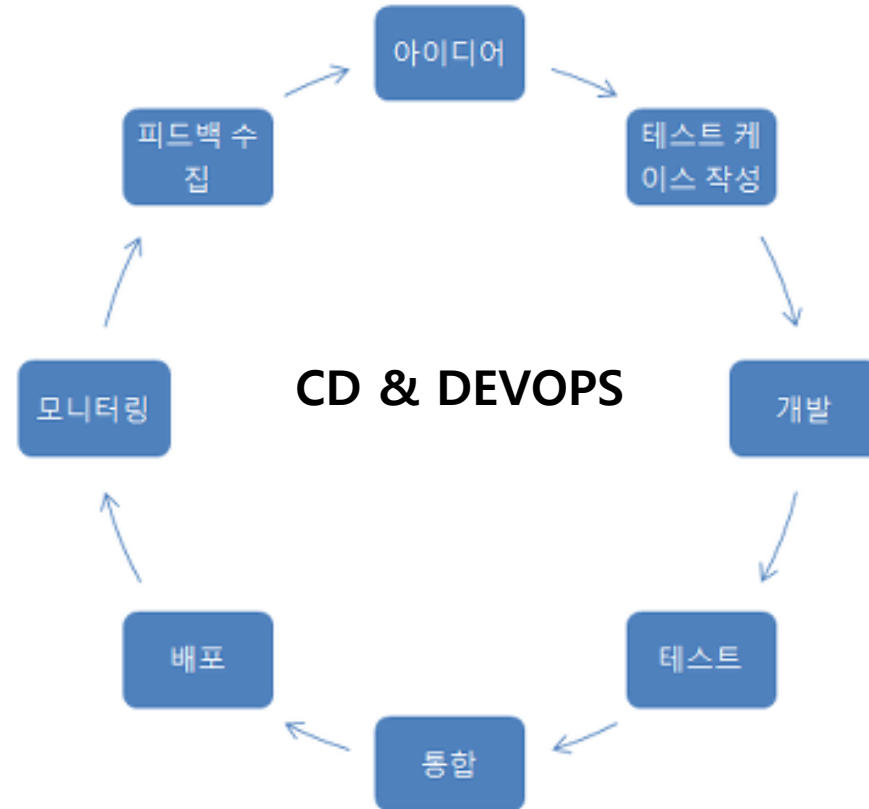
개발 운영팀의 변화 모델

- 2단계. 펀딩 받았다



개발 운영팀의 변화 모델

- 3단계. 깨달았다.



End of document