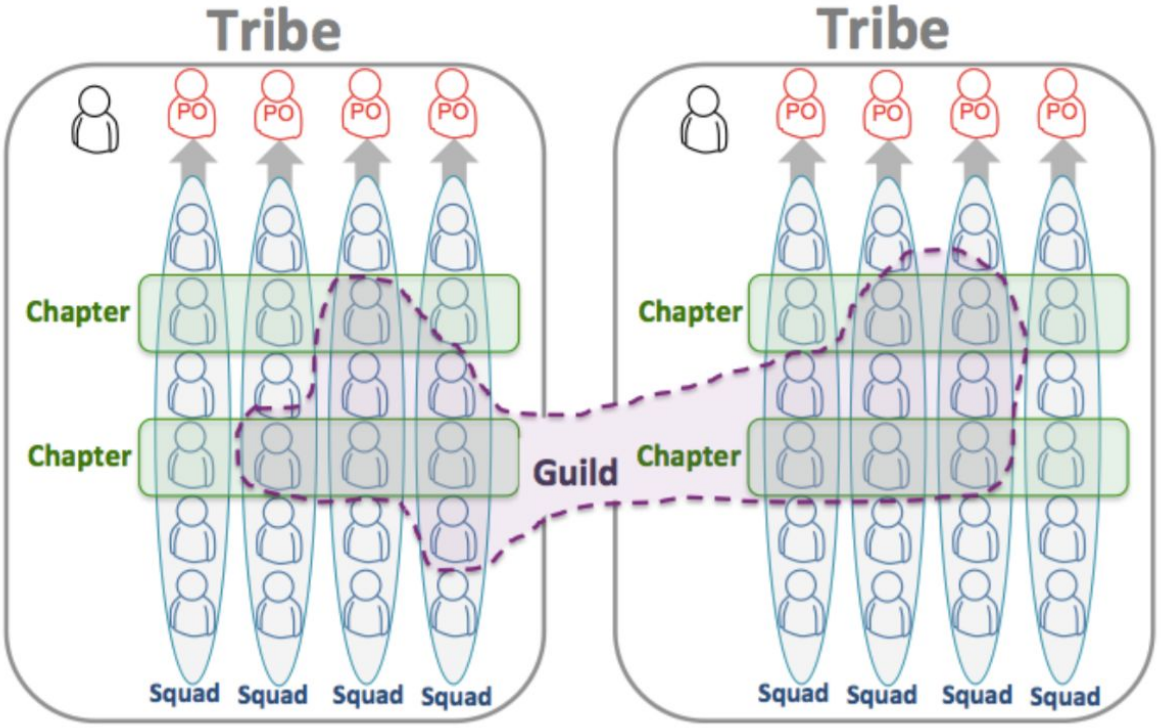
translated by Chad

Scaling Agile @ Spotify

With Tribes, Squads, Chapters & Guilds

Henrik Kniberg & Anders Ivarsson

Oct 2012



제품 개발 조직에서 여러 팀을 상대하는 것은 항상 어려운 일입니다!

지금까지 살펴본 가장 인상적인 사례 중 하나는 Spotify입니다. 이 사례는 3개 도시에서 30 개 이상의 팀으로 확장되었지만 여전히 애자일 사고 방식을 유지했습니다.

Spotify는 음악 산업을 변화시키고 있는 매력적인 기업입니다. 그 회사는 겨우 6년 동안 존재했지만 이미 1,500만 명 이상의 활동적인 사용자들과 400만 명 이상의 사람들이 돈을 지불하고 있습니다. 이 제품은 "세상의 모든 노래를 즉시 찾아서 재생할 수 있는 마법의 음악 플레이어"에 비유될 수 있습니다.

애자일 소프트웨어 개발의 창시자 중 한 명인 Alistair Cockburn은 Spotify를 방문하여 “나이스! 저는 1992 년 이후부터 매트릭스 형식을 구현할 사람을 찾고 있었습니다.”라고 말했습니다.

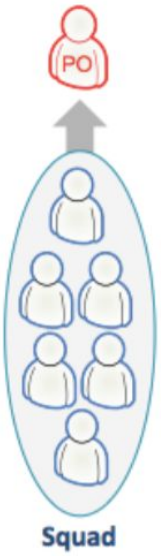
그럼 어떻게 이 일을 해낼 수 있었을까요?

우리는 Spotify에서 일하는 방식과 회사가 수백 명의 개발자와 함께 애자일하게 대처하는 방법에 관하여 컨퍼런스에서 프레젠테이션을 하기도 하고 토론에도 참여했습니다. 많은 사람들이 Spotify의 방식에에 매료되었고, 우리는 이 여정에 대한 이야기를 쓰기로 결정했습니다.

주의 사항: 우리는 이 모델을 발명하지 않습니다. Spotify는 빠르게 진화하고 있고 이 이야기는 우리의 현재 작업 방식을 보여주는 스냅 사진일 뿐, 완성된 여정이 아닙니다. 이 글을 읽을 때쯤이면 이미 상황은 달라져 있을 수 있습니다.

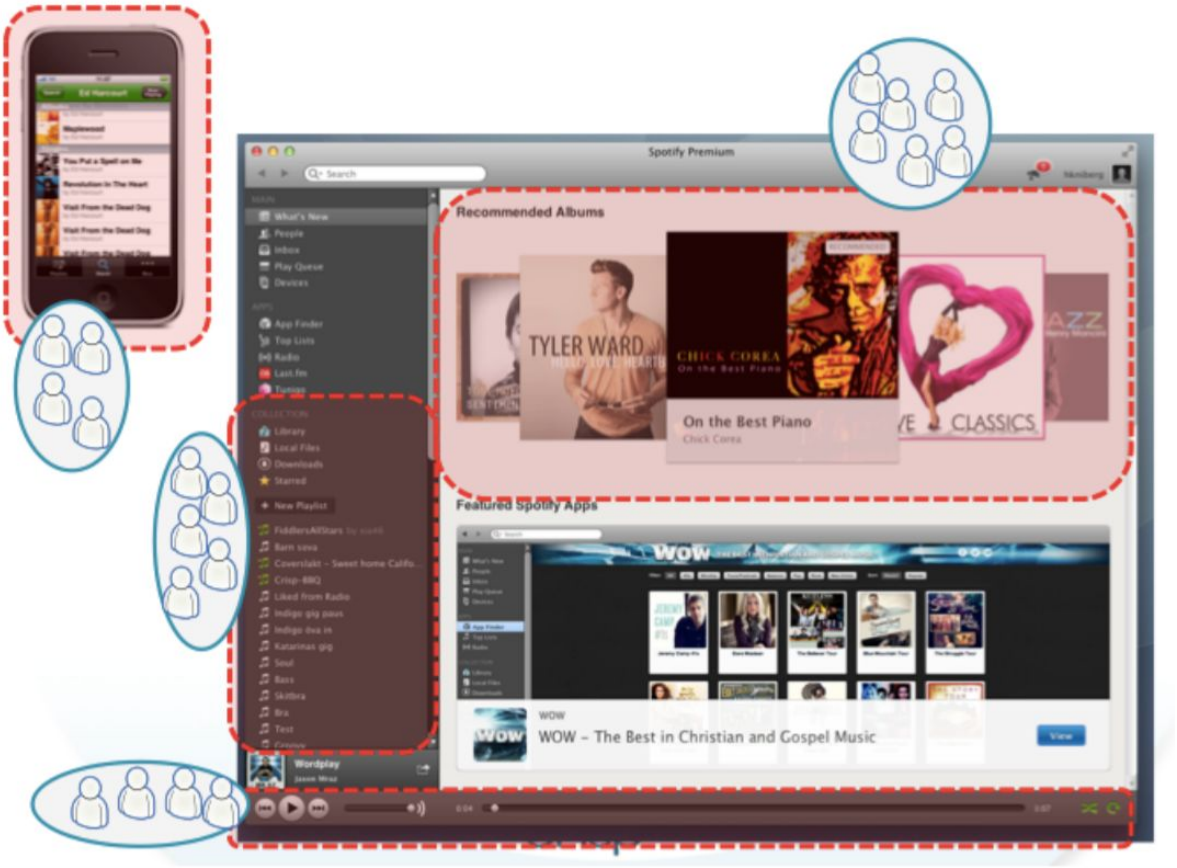
스쿼드(Squad)

Spotify에서 개발의 기본 단위는 Squad입니다.



스쿼드는 스크럼 팀과 비슷하며, 미니 스타트업처럼 느껴지도록 설계되었습니다. 그들은 같은 장소에 함께 앉아있으며, 디자인, 개발, 테스트, 생산에 필요한 모든 기술과 도구를 가지고 있습니다. 그들은 스스로 조직화된 팀이며 그들만의 작업 방식을 스스로 결정합니다. 어떤 이들은 Scrum 스프린트를 사용하고, 어떤 이들은 칸반을 사용하고, 어떤 이들은 이러한 접근법을 혼합하여 사용하기도 합니다.

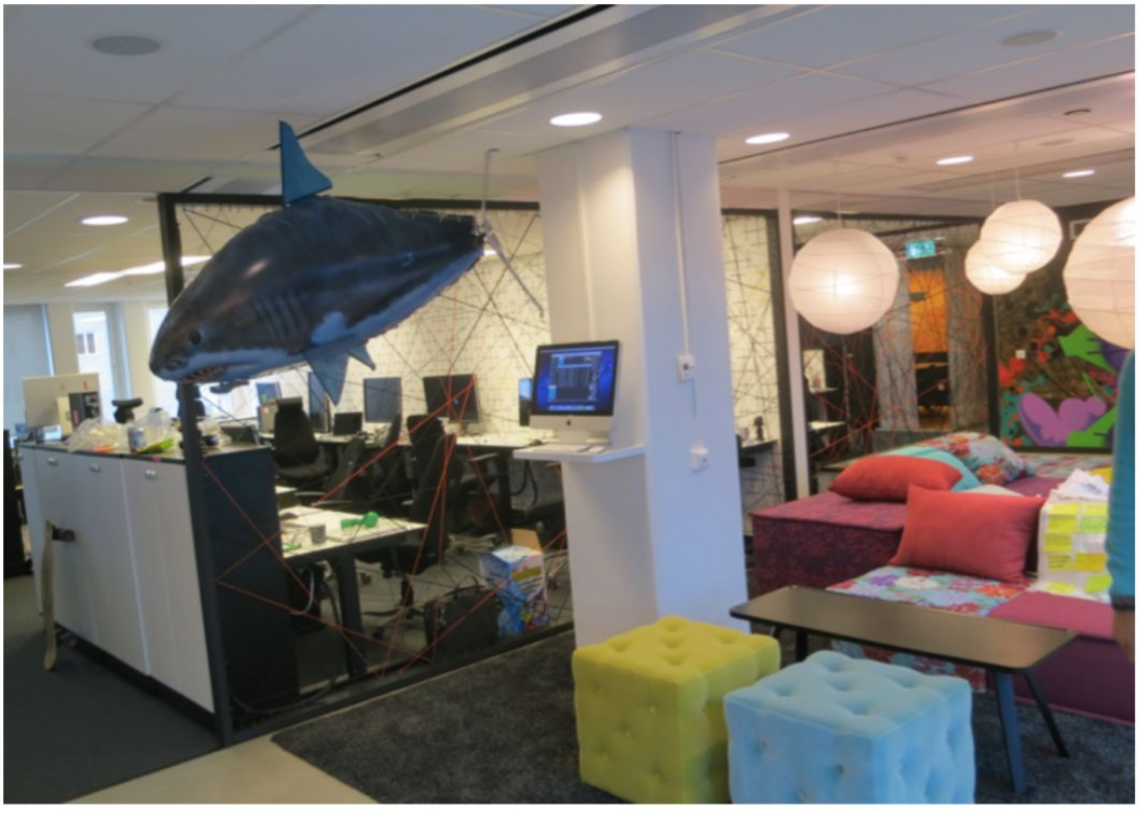
각 스쿼드는 안드로이드 클라이언트 구축 및 개선, Spotify 무선 경험 만들기, 백엔드 시스템 확장 또는 결제 솔루션 제공과 같은 장기적인 임무를 가지고 있습니다. 아래 그림은 다양한 스쿼드가 사용자 경험의 다른 부분을 어떻게 담당하는지 보여줍니다.



스쿼드는 MVP(최소 실행 가능한 제품)와 "검증된 학습"과 같은 Lean Startup 원칙을 적용하도록 권장됩니다. MVP는 조기에 자주 출시하는 것을 의미하며 "검증된 학습"이란 메트릭과 A/B 테스트를 사용하여 실제로 의미 있는 것과 그렇지 않은 것을 찾는 것을 의미합니다. 이것은 “생각하고, 만들고, 배포하고, 조정하라”이라는 슬로건으로 요약되어 있습니다

각 스쿼드는 오랫동안 하나의 임무와 제품의 한 부분을 고수하기 때문에, 그들은 실제로 그 분야의 전문가가 될 수 있습니다. 예를 들어, 멋진 라디오 경험을 만들어 나가는 것이 무엇을 의미하는지 알고 있습니다.

대부분의 스쿼드는 데스크 공간, 라운지 공간, 개인용 "허들" 방을 포함한 멋진 작업 공간을 가지고 있습니다. 거의 모든 벽은 화이트보드로 되어 있습니다. 우리는 이보다 더 나은 협업 공간을 결코 본 적이 없습니다.



그렇습니다, 상어입니다. 아무 문제 없습니다. 완벽히 정상입니다.

학습과 혁신을 촉진하기 위해, 각 스쿼드는 시간의 약 10%를 "해킹 데이"에 사용하도록 권장됩니다. 해킹 기간 동안 사람들은 보통 새로운 아이디어를 시도해보고 친구들과 공유하면서 그들이 원하는 모든 것을 합니다. 어떤 팀은 2주마다 1일씩 해킹을 하고, 다른 팀은 전체 "해킹 주간"을 위해 저축을 합니다. 해킹데이는 재미있을 뿐만 아니라, 새로운 도구와 기법을 통해 최신 정보를 얻을 수 있는 좋은 방법입니다. 때로는 중요한 제품 혁신으로

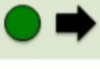
이어집니다!

스쿼드에는 공식적으로 임명 된 스쿼드 리드는 없지만 제품 소유자가 있습니다. 제품 소유자는 팀이 수행해야 할 작업의 우선순위를 정할 책임이 있지만, 작업 방식에는 관여하지 않습니다. 각 부문별 제품 소유자는 전체 Spotify가 어디로 향하는지 보여주는 상위 레벨의 로드맵 문서를 유지하기 위해 서로 협력하며, 각 제품 소유자는 자신의 팀에 맞는 제품 백로그를 유지할 책임이 있습니다.

또한 스쿼드는 애자일 코치의 도움을 받을 수 있으며, 코치는 스쿼드의 업무 방식을 발전시키고 개선하는 데 도움을 줍니다. 코치는 회고, 스프린트 플래닝, 1대1 코칭 등 다양한 활동을 합니다.

이상적으로는 각 스쿼드는 이해관계자와 직접 연락하고 다른 스쿼드에 대한 의존성으로 업무가 중단되지 않으며 완전히 자율적으로 동작합니다. 기본적으로 미니 스타트업과 같습니다. 30개 이상의 팀과 함께하는 것은 도전입니다. 우리는 먼 길을 왔지만 아직 개선해야 할 것이 많습니다.

이를 돕기 위해 우리는 각 스쿼드와 함께 분기별로 설문 조사를 실시합니다. 이를 통해 개선 노력에 집중하고 어떤 종류의 조직 지원이 필요한지 파악할 수 있습니다. 다음은 이러한 설문 조사 중 한 가지를 시각적으로 요약 한 것으로 트라이브 내 5개의 스쿼드를 보여줍니다.

Area	Squad 1	Squad 2	Squad 3	Squad 4	Squad 5
Product owner					
Agile coach					
Influencing work					
Easy to release					
Process that fits team					
A mission					
Org. support					

원은 현재 상태를 나타내고 화살표는 추세를 나타냅니다. 예를 들어, 우리는 세 스쿼드가 릴리즈에 관한 문제를 보고하는 패턴을 볼 수 있고, 개선되지 않는 것 같습니다. - 이 영역에 집중이 필요합니다. 또한 스쿼드 4는 좋은 상황은 아니지만 애자일 코치의 지원으로 이미 개선되고 있음을 알 수 있습니다.

- **제품 소유자** - 전담 제품 소유자가 있어 작업에 우선 순위를 두고 비즈니스 가치와 기술 측면을 모두 고려합니다.
- **애자일 코치** - 팀에는 애자일 코치가 있어 장애를 식별하고 지속적으로 프로세스를 개선하도록 코치합니다.
- **작업에 영향 미치기** - 각 스쿼드 멤버는 자신의 작업에 영향을 미치는 계획에 적극적으로 참여하며 어떤 작업을 수행할지 선택할 수 있습니다. 모든 스쿼드 멤버는 자신의 시간의 10 %를 해킹 업무에

사용할 수 있습니다.

● **손쉬운 릴리스** - 스쿼드는 번거로운 업무와 동기화 작업을 최소화하여 릴리즈를 달성할 수 있습니다.

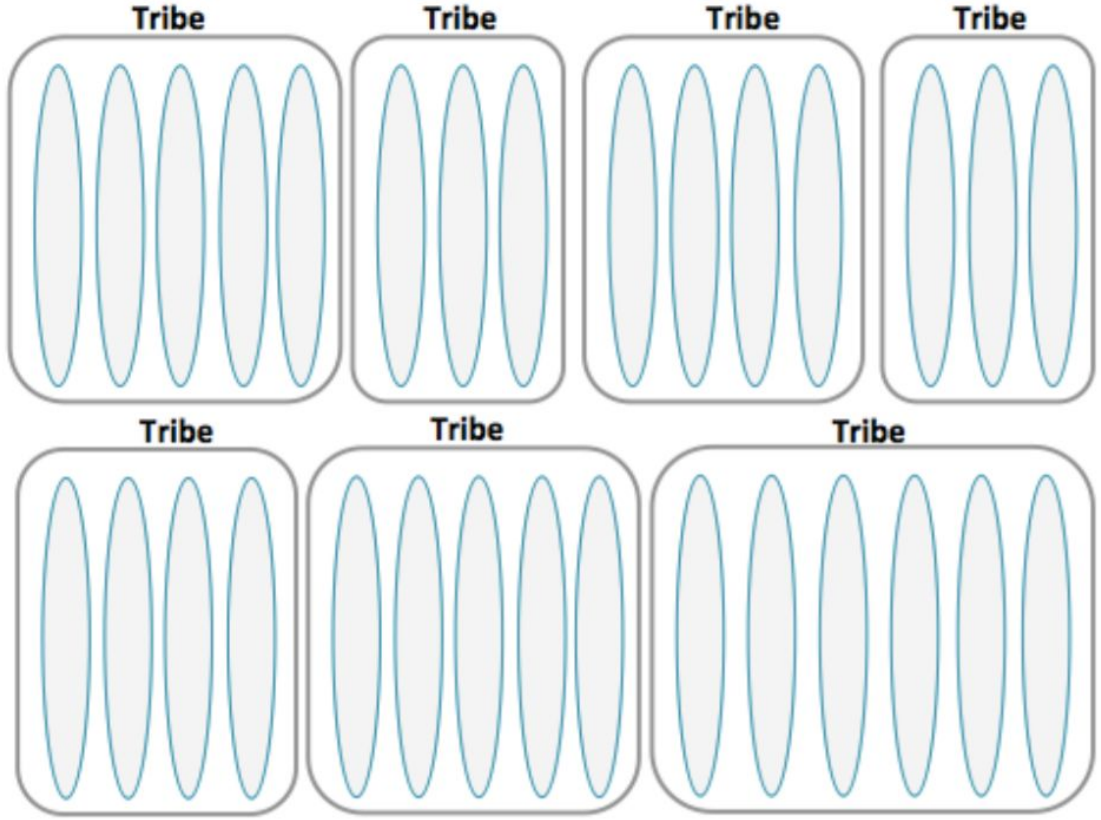
● **팀에 맞는 프로세스** -스쿼드는 프로세스의 소유권을 가지고 지속적으로 개선합니다.

● **미션** - 스쿼드는 모든 사람이 알고 있고 중요하게 생각하는 미션을 가지고 있으며 백로그의 스토리는 미션과 연결되어 있습니다.

● **조직 지원** – 스쿼드는 기술 문제 및 "소프트"한 문제에 대해 해결 지원을 받을 수 있는 곳을 알고 있습니다.

트파이브(부족, Tribe)

트라이브는 음악 플레이어 또는 백엔드 인프라와 같은 관련 분야에서 일하는 스쿼드들의 집합입니다.



트라이브는 스쿼드 미니 스타트 업을 위한 "인큐베이터"로 볼 수 있습니다. 트라이브는 자유와 자율성을 갖습니다. 각 트라이브에는 그 트라이브 속해 있는 스쿼드에게 가능한 최고의 업무 환경을 제공할 책임을 맡고 있는 트라이브 리더가 존재합니다. 한 트라이브의 스쿼드들은 모두 같은 사무실에 있고, 보통은 바로 옆에 있으며, 근처의 라운지 구역은 스쿼드들 간의 협력을 촉진하게 됩니다.

트라이브는 대부분의 사람들이 100 명 이상의 사람들과 사회적 관계를 유지할 수 없다고 말하는 “던바 의 수”개념에 따라 규모가 정해집니다 (실제로 Spotify의 경우 생존 압력이 강하지 않은 그룹의 경우 실제로는 더 큽니다. 믿거나 말거나 ...). 그룹이 너무 커지면 제한 규칙, 관료주의, 정치, 추가 관리 계층 및 기타 낭비와 같은 더 많은 것들이 보이기 시작합니다.

그래서 트라이브는 100명 정도 이하로 설계되어 있습니다.

트라이브는 정기적으로 모임을 엽니다. 이 비공식 모임을 통해 그들이 트라이브의 나머지 부분 (또는 누구에게나 나타나고 있는지)이 무엇을 하고 있는지, 그들이 무엇을 제공했는지, 현재하고 있는 일에서 다른 사람들이 배울 수있는 것은 어떤 것들이 있는 지를 보여줍니다. 여기에는 작동하는 소프트웨어의 라이브 데모, 새로운 도구 및 기술, 멋진 해킹 데이 프로젝트 등이 포함됩니다.

스쿼드 의존성

여러 스쿼드가 존재하게 되는 경우 항상 의존성을 가지게 됩니다. 의존성은 반드시 나쁜 것은 아닙니다. 스쿼드는 때때로 정말 멋진 것을 만들기 위해 협력해야 합니다. 그럼에도 불구하고, 우리의 목표는 스쿼드가 가능한 한 자율적이며 특히 스쿼드을 멈추게 하거나 속도를 늦추는 의존성을 최소화하는 것입니다.

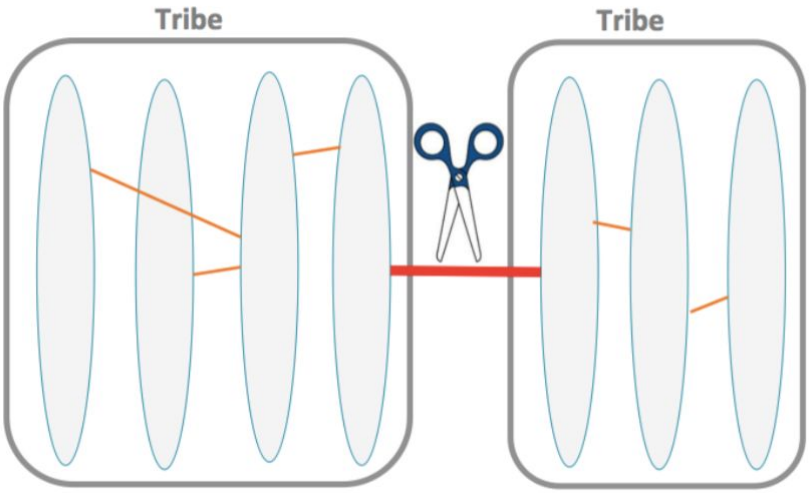
이것을 달성하기 위해, 우리는 정기적으로 모든 우리 팀에게 그들이 어느 팀에 의존하고 있는지, 그리고 그 트라이브가 스쿼드를 멈추게 하거나 느리게 하는 정도를 묻는다. 예를 들면 다음과 같습니다.

Text

	A	B	C	D	E
1	**Squad**	**Depends on**	**Dependency**	**Comment**	**Same tribe?**
2	**Music Player**				
3	Content	Ops	Slowing	Need machines, connections, help set-up things etc. Works really well in general, but at times the workload on operations causes the lead times to grow and slow us down	No
4	Content	NeXT	No problem	Storage. Not big, mostly information/communication needs to happen.	No
5	Content	BFS	No problem	Replacement service	Yes
6	Content	Team 2	No problem	Communication around next story	No
7	Content	Team 1	Future	Content ingestion	No
8	BFS	UX	Slowing	Need UX to discuss, review and provide mock-ups.	No
9	BFS	Content	No problem	Normal dependencies, sprint work.	Yes
10	BFS	Mobile	Slowing	No internal mobile developers within Squad.	No
11	BFS	Analytics	Slowing	A/B test results slowing down roll outs of features	No
12	BFS	Team 3	Blocking	Waiting for data dumps	No
13	BFS	Team 1	Future	Waiting for data dumps	No
14					

그런 다음 우리는 문제가 있는 의존성, 특히 차단 및 교차 의존성을 제거하는 방법에 대해 논의합니다

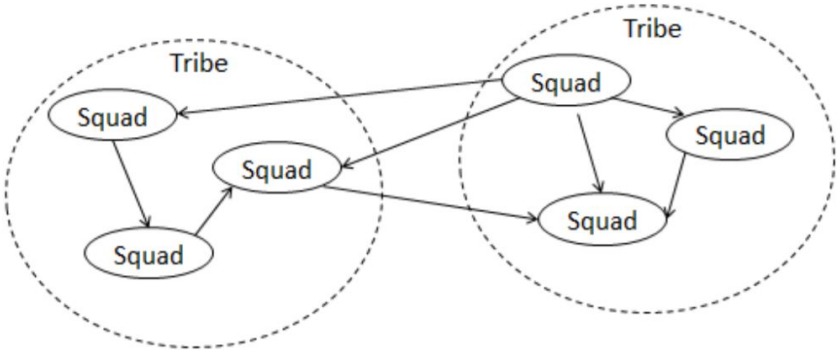
이것은 종종 우선 순위 변경, 재구성, 아키텍처 변경 또는 기술 솔루션으로 이어집니다.



또한 이 설문 조사는 스쿼드가 서로 의존하는 방식에 대한 패턴을 확인하는 데 도움이 됩니다. 예를 들어 점점 더 많은 스쿼드가 운영으로 인해 둔화되는 것으로 보입니다. 우리는 다양한 유형의 의존성이 시간이 지남에 따라 어떻게 증가하거나 감소하는지 추적하기 위해 간단한 그래프를 사용합니다.

스크럼은 "스크럼의 스크럼"이라고 불리는 실천법이 있는데, 이것은 각 팀의 한 사람이 모두 만나 의존성에 대해 논의하는 동기화 모임입니다. 우리는 Spotify에서 보통 스크럼 오브 스크럼을 하지 않는데, 주로 대부분의 스쿼드는 상당히 독립적이어서 그러한 조정 회의가 필요하지 않기 때문이다.

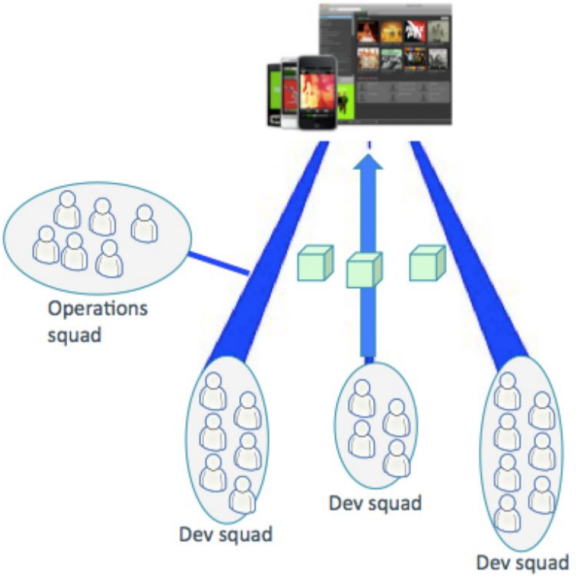
대신에 “스크럼의 스크럼”은 "요구에 따라" 발생합니다. 예를 들어, 우리는 여러 스쿼드간의 몇 달간의 조정된 작업을 필요로 하는 대규모 프로젝트를 진행한 적도 있습니다.



이를 위해 팀들은 매일 동기화 회의를 열어 스쿼드 간의 의존성을 파악하고 해결했으며, 포스티잇이 달린 보드를 사용하여 해결되지 않은 의존성을 추적하였습니다.

많은 회사에서 공통적인 의존성 문제의 원인은 개발과 운영간에 발생합니다. 우리와 함께 일한 대부분의 회사에서 개발자에서 운영부서로 핸드오프될 때 마찰과 지연이 발생합니다.

Spotify에는 별도의 운영 팀이 있지만, 그들의 일은 스쿼드를 위해 릴리즈를 하는 것이 아닙니다. 그들의 일은 스쿼드가 스스로 코드를 릴리스하는 데 필요한 지원을 제공하는 것이다. 인프라, 스크립트 및 루틴 형태의 지원을 제공하는 것입니다. 그들은 어떤 의미에서 “Production으로 가는 길을 만드는 것” 입니다.



세부적인 프로세스 문서화보다는 직접 대면 커뮤니케이션에 기반한 비공식적이지만 효과적인

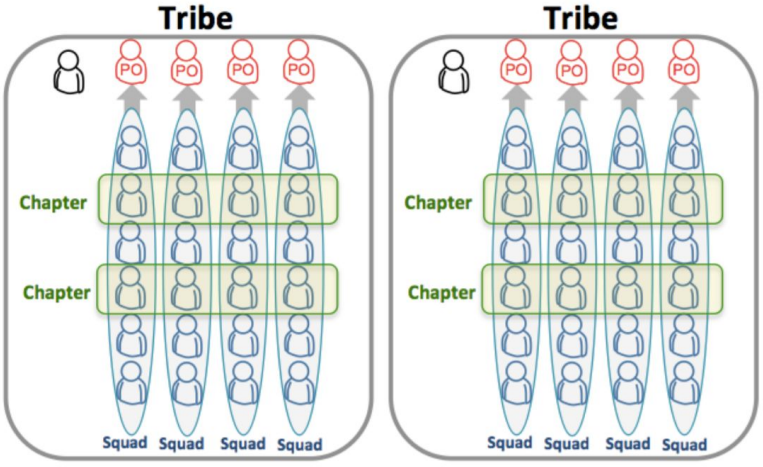
협업방식으로 일을 합니다.

챕터(Chapter)와 길드(Guild)

모든 것에는 단점이 있으며, 완전 자율성의 잠재적 단점은 규모의 경제에 대한 손실입니다. 스쿼드 A의 테스터가 지난 주 스쿼드 B의 테스터가 해결 한 문제로 씨름하고 있을 수 있습니다. 모든 테스터가 스쿼드와 트라이브에 걸쳐 모일 수 있다면 모든 스쿼드를 위해 지식을 공유하고 도구를 만들 수 있습니다.

각 스쿼드가 완전히 자율적이고 다른 스쿼드와 의사 소통을하지 않는다면 회사를 유지하는 이유가 무엇일까요? Spotify는 30 개의 다른 소기업으로 나누어 질 수도 있을 겁니다.

그래서 우리는 챕터와 길드를 가지고 있습니다. 이것은 회사를 하나로 묶는 접착제이며, 너무 많은 자율권을 희생하지 않으면서 규모의 경제를 제공합니다. 챕터는 비슷한 기술을 가지고 동일한 트라이브 내에서 동일한 역량 영역 내에서 일하는 소규모 가족입니다.

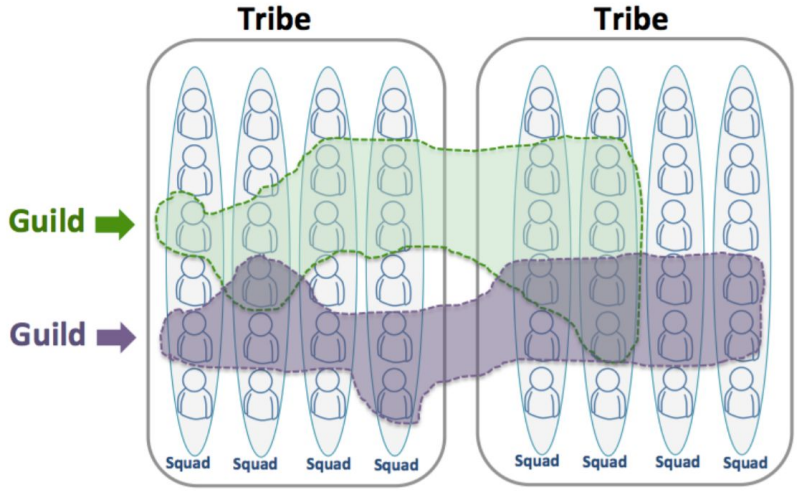


각 챕터은 정기적으로 만나 전문 분야 및 특정 과제 (예 : 테스트 챕터, 웹 개발자 챕터 또는 백엔드 챕터)에 대해 논의합니다.

챕터 리드는 직원 개발, 급여 설정 등과 같은 모든 전통적인 책임을 가진 챕터 멤버의 라인 관리자입니다. 그러나 챕터 리드 역시 스쿼드 소속으로 일상 업무에 관여하고 있어 현실과의 접촉을 유지하는 데 도움이 된다.

이제 현실은 항상 위의 그림과 같은 예쁜 그림보다 더 지저분합니다. 예를 들어, 챕터 멤버는 스쿼드에 균등하게 분배되지 않습니다. 일부 스쿼드에는 많은 웹 개발자가 있고 일부 스쿼드에는 없습니다. 그러나 그림은 일반적인 아이디어를 제공하기 위해 그려 졌습니다.

길드 (Guild)는 지식, 도구, 코드 및 실습을 공유하려는 사람들의 모임으로 유기적이고 광범위한 "관심있는 커뮤니티"입니다. 챕터는 항상 트라이브에 국한된 반면, 길드는 일반적으로 조직 전체에 적용됩니다. 웹 기술 길드, 테스터 길드, 애자일 코치 길드 등이 그 예입니다.

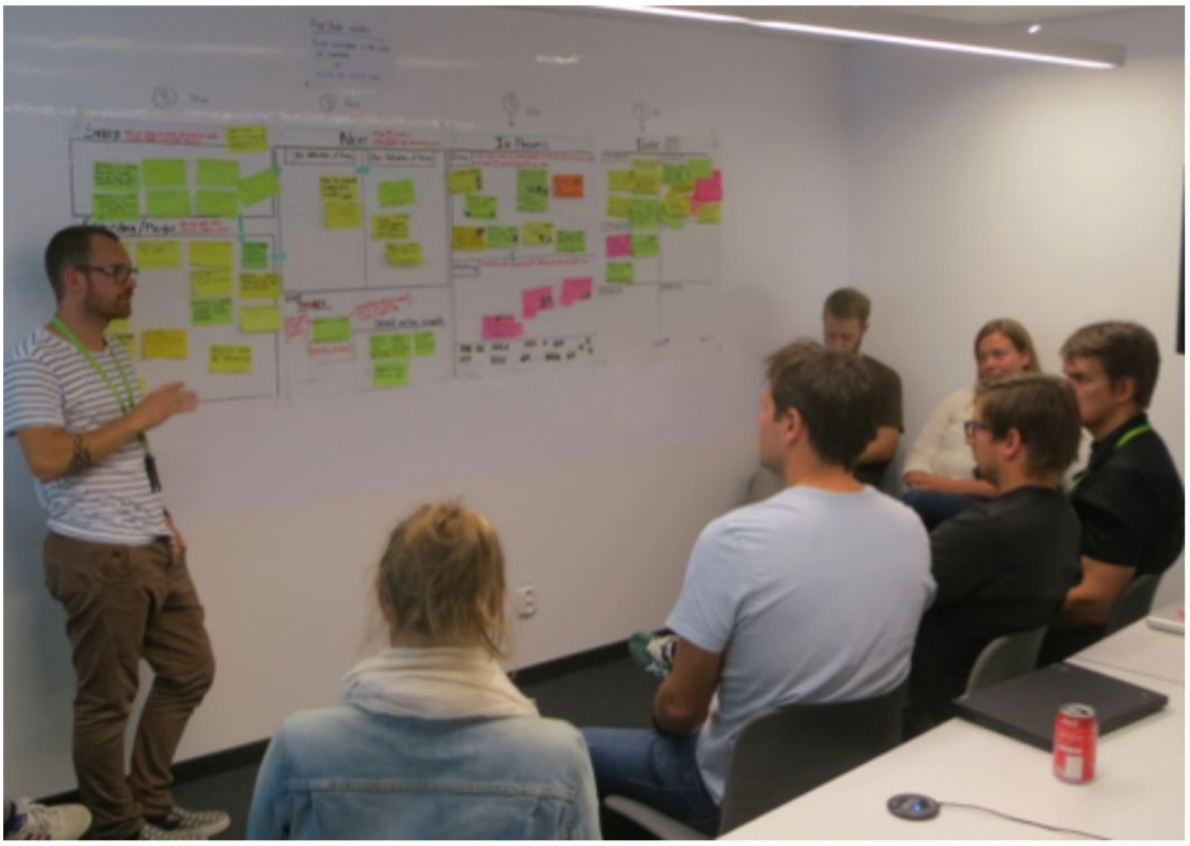


길드에는 종종 해당 지역에서 일하는 모든 챕터와 그 멤버가 포함됩니다. 예를 들어, 테스트 길드에는 모든 테스트 챕터의 모든 테스터가 포함되지만 관심 있는 사람은 모든 길드에 참여할 수 있습니다.

각 길드에는“길드 코디네이터”가 있습니다.

길드 작업의 한 예로, 우리는 최근 "웹 길드 컨퍼런스"를 개최했는데, 그것은 Spotify의 모든 웹 개발자들이 스톡홀름에 모여 그들의 분야 내의 도전과 해결책을 토론하는 오픈 스페이스 행사였습니다.

또 다른 예는 애자일 코치 길드입니다. 코치는 조직 전체에 퍼져 있지만 지속적으로 지식을 공유하고 정기적으로 만나 개선 위원회에서 추적하는 높은 수준의 조직 개선 영역에서 협력합니다.



잠깐만, 이건 단순한 매트릭스 조직 아닌가요?

음. 글쎄요. 그것은 우리 대부분이 익숙한 것과는 다른 유형의 매트릭스입니다.

많은 매트릭스 조직에서 유사한 기술을 가진 사람들은 기능 부서로 "풀링"되고 프로젝트에 "할당"되고 기능 관리자에게 "보고"됩니다.

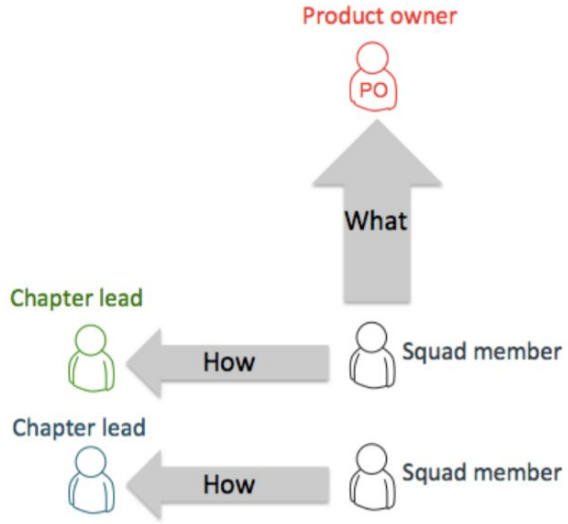
Spotify는 거의 이 작업을 수행하지 않습니다. 우리의 매트릭스는 딜리버리를 더욱 중요하게 생각합니다.

즉, 사람들은 서로 다른 기술 세트를 가진 사람들이 협업하고 자체 구성하여 훌륭한 제품을 제공하는 안정적인 공동 배치 팀으로 그룹화됩니다. 그것은 매트릭스의 수직 차원이며, 사람들이 물리적으로 모이며 대부분의 시간을 보내는 장소이기 때문에 더 중요한 그룹입니다.

수평 차원은 지식, 도구 및 코드를 공유하기 위한 것입니다. 이 챕터의 임무는 이를 촉진하고 지원하는 것입니다.

매트릭스 용어로 세로 영역을 "무엇"으로, 가로 영역을 "방법"으로 생각하십시오.

매트릭스 구조는 각 스쿼드 멤버가 "다음에 무엇을 만들 것인가"와 "잘 만드는 방법"에 대한 지침을 얻을 수 있도록 합니다.



이것은 Mary와 Tom Poppendieck이 권장하는 “교수 및 기업가”모델과 일치합니다. PO는 훌륭한 제품을 제공하는 데 중점을 둔 "기업가" 또는 "제품 챔피언"이며, 챕터 리드는 기술적 우수성에 중점을 둔 "교수"또는 "역량 리더"입니다.

기업가가 속도를 높이고 지름길을 찾고 싶어하는 반면 교수는 속도를 늦추고 물건을 제대로 만들고 싶어하는 경향이 있기 때문에 이러한 역할 사이에는 건전한 긴장이 있습니다. 두 가지 측면이 모두 필요하므로 이것이 “건강한”긴장이라고 표현된 이유입니다.

설계는 어떠한가요?

Spotify 기술은 서비스 지향적입니다. 우리는 100 개가 넘는 개별 시스템을 보유하고 있으며 각 시스템은 별도로 유지 관리 및 배포 할 수 있습니다. 여기에는 재생 목록 관리 또는 검색 또는 결제와 같은 백엔드 서비스와 iPad 플레이어와 같은 클라이언트, 라디오와 같은 특정 구성 요소 또는 음악 플레이어의 "새로운 기능" 섹션이 포함됩니다.

기술적으로는 누구나 시스템을 편집 할 수 있습니다. 스쿼드는 효과적으로 일하는 기능 팀이므로 일반적으로 여러 시스템을 업데이트하여 새로운 기능을 프로덕션에 적용해야 합니다.

이 모델의 위험성은 아무도 시스템 전체의 무결성에 중점을 두지 않으면 시스템 아키텍처가 엉망이 된다는 것입니다.

이러한 위험을 완화하기 위해 "시스템 소유자"라는 역할이 있습니다. 모든 시스템에는 시스템 소유자 또는 시스템 소유자 페어가 있습니다 (페어링 권장). 운영상 중요한 시스템의 경우 시스템 소유자는 개발자 관점을 가진 한 사람과 운영 관점을 가진 사람으로 이루어 집니다.

시스템 소유자는 해당 시스템과 관련된 기술 또는 구조적 문제에 대한 집중하는 사람입니다. 그는 코디네이터이며 시스템에서 코딩하는 사람들이 서로 걸려 넘어지지 않도록 안내합니다. 그는 품질, 문서, 기술 부채, 안정성, 확장성 및 릴리스 프로세스와 같은 것에 중점을 둡니다.

시스템 소유자는 모든 것이 거쳐가야 하는 통로나 상아탑의 설계자가 아닙니다. 그는 개인적으로 모든 결정을 내리거나 모든 코드를 작성하거나 모든 릴리스를 수행 할 필요는 없습니다. 그는 일반적으로 시스템 소유권 외에 다른 일상적인 책임을 가진 스쿼드 멤버 또는 챕터 리드입니다. 그러나 때때로 그는 "시스템 소유자의 날"을 수행하며 해당 시스템을 정리하는 작업을 수행합니다. 보통 우리는 이 시스템 소유권을 사람의 10분의 1 이하로 유지하려고 하지만, 물론 시스템마다 많이 다릅니다.

우리는 또한 여러 시스템을 가로지르는 높은 수준의 설계 문제에 대한 작업을 조정하는 치프 설계자 역할도 가지고 있습니다. 그는 새로운 시스템의 개발을 검토하여 그들이 일반적인 실수를 방지하고, 새로운 시스템의 개발이 우리의 아키텍쳐 비전과 일치하는지 확인합니다. 피드백은 항상 제안과 입력일 뿐 입니다. 시스템의 최종 설계에 대한 결정은 여전히 시스템을 구축하는 팀에 달려 있습니다.

이 모든 것이 어떻게 작동합니까?

Spotify는 매우 빠르게 성장했습니다. 3년 동안 30명에서 250 명으로 기술 분야에서 성장했습니다. 우리는 성장의 고통을 공유합니다! 스쿼드, 트라이브, 챕터, 길드가 포함된 이 스케일링 모델은 지난 1년 동안 점진적으로 도입되었기 때문에 사람들은 여전히 이 모델에 익숙해지고 있습니다. 그러나 지금까지 설문 조사와 회고를 바탕으로 스케일링 모델이 잘 작동하는 것 같습니다! 그리고 그것은 우리에게 “성장”할 무언가를 줍니다. 고속 성장에도 불구하고 직원 만족도는 지속적으로 증가했고, 2012년 4월에는 5점 만점에 4.4점이었습니다.

그러나 성장하는 조직과 마찬가지로 오늘날의 솔루션은 미래의 문제를 낳습니다. 계속 지켜봐 주십시오. 이 이야기는 끝나지 않았습니다. : o)

/ 헨릭 & 앤더스

henrik.kniberg@spotify.com www.crisp.se/henrik.kniberg

anders.ivarsson@spotify.com