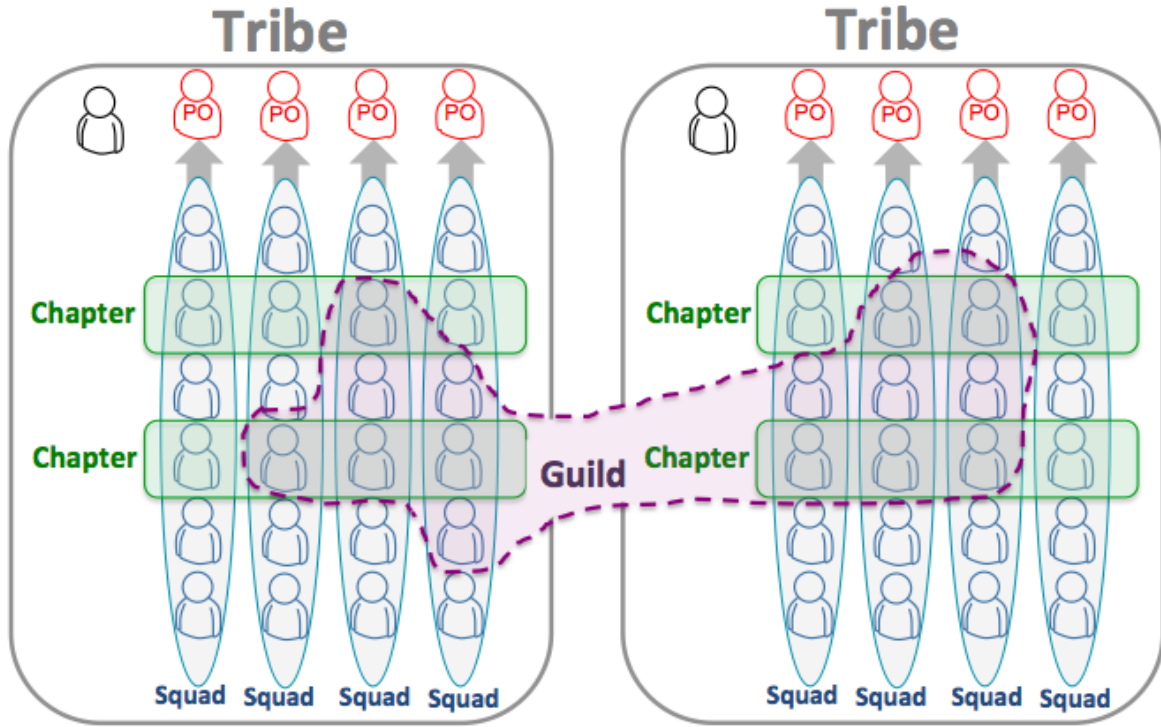


Agile Ölçeklendirme @ Spotify

Tribelar, Squadlar, Chapterlar & Guildler

Henrik Kniberg & Anders Ivarsson
Ekim 2012

baris.bal@acmagile.com
www.acmagile.com



Ürün geliştirme organizasyonlarında bir çok takımla ilgilenmek her zaman zorlu bir iştir!

3 şehirde 30 takıma sahip bir organizasyona rağmen Agile zihniyete sahip olmasından dolayı Spotify, bu zamana kadar gördüğümüz en etkileyici örneklerden birisi.

Spotify, müzik endüstrisini dönüştüren büyüleyici bir şirket. Sadece 6 yıldır var olan şirketin, 15 milyon aktif kullanıcısı ve hizmetlerden yararlanmak için para ödeyen 4 milyon'dan fazla müşterisi bulunmakta. Ürün, "dünyadaki her şarkıyı anlık olarak bulup, dinleyebileceğiniz sihirli bir müzik çalar" 'a benzetilebilir.

Alistair Cockburn (Agile yazılım geliştirmenin kurucularından birisidir), Spotify'ı ziyaret etti ve "Güzel! 1992'den beri bu matriks yapısını uygulayacak birilerini arıyordum; bundan dolayı bunu görmek gerçekten keyifli." dedi.

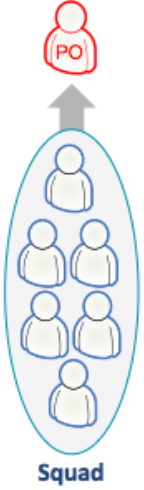
Peki, bu matriks nasıl yönetiliyor?

Bu yöntemi konferanslarda sunduk ve kendimizi, Spotify'da nasıl çalıştığımızı ve şirketin yüzlerce geliştirici ile Agile'ı nasıl idare ettiğini konuştuğumuz tartışmalar içerisinde bulduk. Bir çok insan bundan büyüledi, dolayısı ile bununla ilgili bir makale yazmaya karar verdik.

Uyarı: Bu modeli biz icat etmedik. Spotify (her iyi agile şirket gibi) hızlı geliyor. Bu makale mevcut çalışma yöntemimizin anlık bir resmi – devam eden bir yolculuktur, tamamlanmış değildir. Bu makaleyi okuduğunuzda muhtemelen durum değişmiş olacaktır.

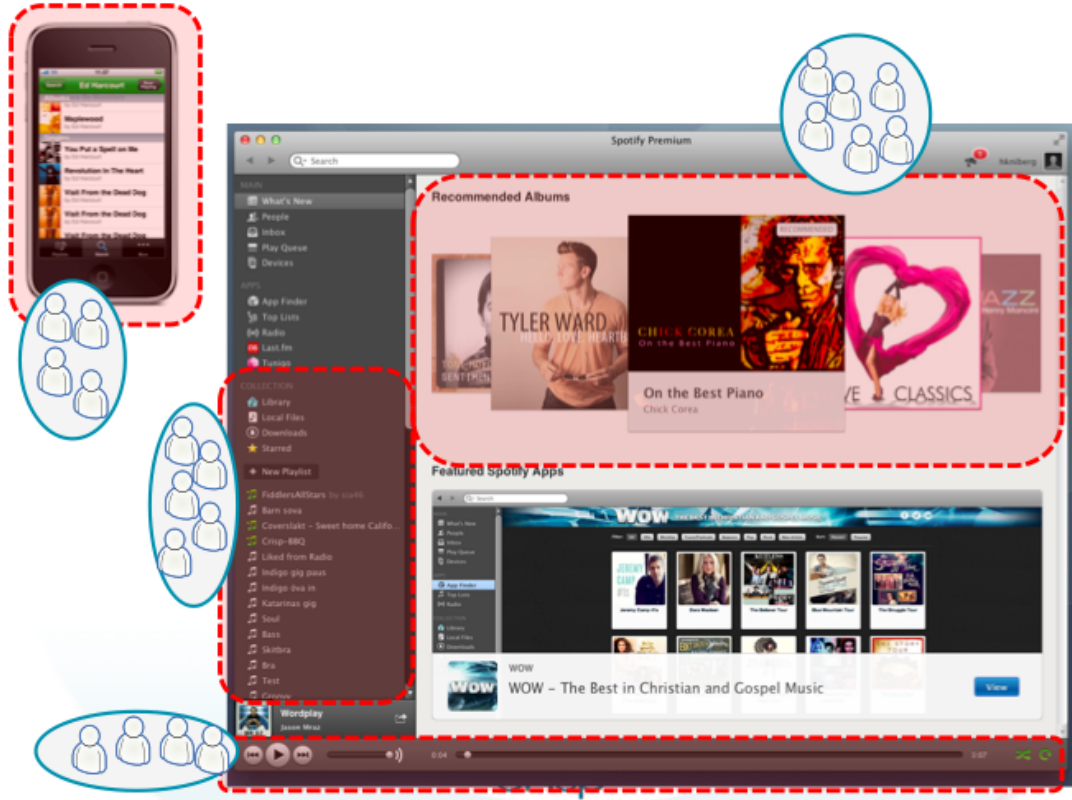
Squadlar

Spotify’da, temel geliştirme birimi Squad’lardır.



Bir Squad, Scrum takımına benzer ve küçük bir start-up gibi tasarlanmıştır. Birlikte otururlar ve tasarım, geliştirme, test ve canlı ortamda yaygınlaştırma yapmak için gerekli tüm yeteneklere ve araçlara sahiptirler. Kendi kendilerine yönetilirler ve çalışma yöntemlerine kendileri karar verirler. Bazıları Scrum sprint’leri kullanır, bazıları Kanban kullanır, bazıları da bu yaklaşımların bir karışımını kullanır.

Her Squad, Android uygulamasını geliştirme ve iyileştirme, Spotify radyo deneyimini yaratma, backend sistemleri ölçeklendirme veya ödeme sistemleri sağlama vb. gibi uzun vadeli bir göreve sahiptir. Aşağıdaki resim farklı Squad’ların, kullanıcı deneyiminin farklı kısımları için nasıl sorumluluk aldığını tanımlamakta.



Squad'lar, Temel Kullanılabilir Ürün (MVP) ve doğrulanmış öğrenim (validated learning) gibi "Yalın Girişim" prensiplerini uygulamaları için cesaretlendirilir. MVP, erken ve sık yaygınlaştırma anlamına gelmektedir. Doğrulanmış öğrenim metriklerinin kullanılması ve A/B testleri ile neyin gerçekten çalıştığı ve neyin çalışmadığı anlamına gelmektedir. Bu yaklaşım: "Düşün, Geliştir, Gönder, İnce Ayar Yap" sloganı ile özetlenmiştir.

Çünkü her Squad'ın sadece bir görevi vardır ve uzun bir süredir ürünün bir parçası ile ilgilenmektedir. Bu nedenle Squad'lar bu alanda uzman olabilirler – örneğin mükemmel radyo deneyimi geliştirme konusunda.

Bir çok Squad, çalışma masası alanı, dinlenme alanı ve kişisel toplanma alanından oluşan mükemmel bir çalışma alanına sahiptir. Neredeyse bütün duvarlar yazı tahtasıdır. Daha önce bundan daha iyi bir işbirliği alanı görmemiştik.



Evet, bu bir uçan köpek balığı. Tamamen normal!

Her Squad, öğrenmeyi ve inovasyonu desteklemek için, zamanlarının kabaca %10'unu "hack günlerine" harcaması yönünde teşvik edilir. "Hack günleri" süresince, insanlar istediklerini yaparlar, genellikle yeni fikirler denerler ve bunu arkadaşları ile paylaşırlar. Bazı takımlar her iki haftada bir "Hack Günü" yaparken, bazıları zamanlarını biriktirerek bütün bir haftayı "Hack Haftası" olarak geçirebilirler. "Hack Günleri" sadece eğlence değildir, aynı zamanda yeni araçlar ve tekniklerle ilgili güncel bilgi sahibi olmanın mükemmel bir yoludur. Bazen önemli ürün inovasyonlarına neden olur.

Squad'ların resmi olarak atanmış bir Squad Liderleri yoktur, ancak bir "Ürün Sahibi" vardır. Ürün Sahibi takım tarafından yapılacak işlerin önceliklendirilmesinden sorumludur, ancak takımın işi nasıl yapacağına dahil olmaz. Ürün Sahipleri, farklı ürün sahipleri ile işbirliği yaparak, Spotify'nın bir bütün olarak hangi yöne gittiğini gösteren bir üst seviye yol haritası dokümanı oluştururlar. Aynı zamanda her Ürün Sahibi, kendi Squad'ları için, üst seviye yol haritası ile uyumlu, bir ürün kapsam dokümanı yönetmekle sorumludur.

Bir Squad'ın, aynı zamanda, çalışma yöntemlerini geliştirmede ve iyileştirmede yardımcı olacak bir Agile Coach'a erişimi bulunmaktadır. Coach'lar süreç gözden geçirme toplantıları, sprint planlama toplantıları düzenlerler, bire bir mentörlük yaparlar, vs.

İdeal olarak her squad, paydaşları ile doğrudan iletişim halinde olacak şekilde tamamen özerktir. Diğer Squad'ları bloklayan bağımlılıkları yoktur. Temel olarak her biri küçük birer start-up'tır. 30'un üstünde takımla bu bir zorluk. Uzun bir yol kat ettik, ancak hala yapılacak pek çok iyileştirme var.

Bunu desteklemek için, yılın her çeyreğinde Squad'larla bir anket yapıyoruz. Bu anketler iyileştirme eforlarımızı odaklamak ve ne çeşit bir organizasyonel desteğe ihtiyaç duyulduğunu belirlememiz konusunda yardımcı oluyor. Aşağıda bir Tribe içerisindeki, 5 Squad için yapılmış anketlerden birinin görsel özeti var:

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					

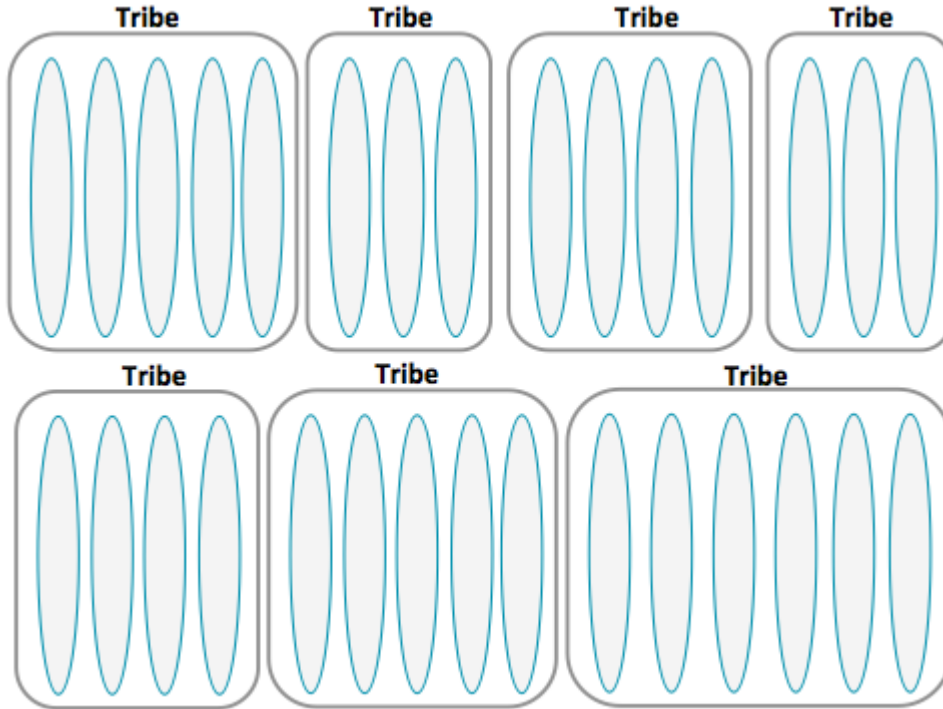
Daireler mevcut durumu, Oklar trendi gösteriyor. Örneğin, 3 Squad'ın yaygınlaştırma ile ilgili problemleri olduğunu ve bu durumun iyileşmediğini görebiliriz. Bu alan acil odaklanma gerektiriyor. Ayrıca 4 Squad'ın Agile Coach desteği ile ilgili mükemmel bir durumları olmadığını, ama bu durumun iyileştiğini görebiliyoruz.

- **Ürün Sahibi** – Squad, iş değerini ve teknik boyutları göz önünde bulundurarak iş önceliklendirmesi yapan, adanmış bir Ürün Sahibini içerir.
- **Agile Coach** – Squad, engellerin belirlenmesinde yardımcı olan ve sürecin sürekli iyileştirilmesi için mentörlük yapan bir Agile Coach'a sahiptir.
- **Etkileyen İş** – Her Squad üyesi kendi işini etkiler. Planlamada aktif rol alır ve hangi iş üzerinde çalışacağını seçer. Her Squad üyesi zamanının %10'unu, "Hack Günü" için haralayabilir.

- **Kolay Yaygınlaştırılabilir** – Squad, geliştirdiği işi minimum güçlük ve senkronizasyon ihtiyacı ile canlıya alabilir (ve alır!).
- **Takıma uyan süreç** – Squad, süreçle ilgili sahiplenme hisseder ve sürekli olarak bunu iyileştirir.
- **Görev** – Squad'ın herkes tarafından bilinen ve önemsenen bir görevi vardır. Kapsamdaki tüm hikayeler bu görevle ilişkilidir.
- **Organizasyonel Destek** – Squad ne zaman bir problem çözümü (teknik problemlerin yanı sıra ufak problemler için de) için desteğe ihtiyacı olduğunu bilir.

Tribe

Bir *Tribe*, müzik oyuncısı veya backend sistem altyapısı gibi ilişkili alanlarda çalışan Squad gruplarından oluşur.



Tribe, küçük Startup Squad'ları için kuluçka merkezi olarak görülebilir. Tribe'lar adil bir seviye de özgürlüğe ve özerkliğe sahiptir. Her Tribe, Tribe içerisinde bulunan Squad'lara mümkün olan en iyi yaşam alanını sağlamaktan sorumlu bir Tribe Lideri'ne sahiptir. Bir Tribe'a bağlı Squad'lar fiziksel olarak aynı ofis binasında bulunurlar, normal olarak yan yanadırlar. Yakınlarındaki dinlenme alanları Squad'lar arasındaki işbirliğini teşvik eder.

Tribe'ların büyüklüğü "Dunbar Sayısı" konsepti ile belirlenir. Dunbar Sayısı konsepti, bir çok insanın, 100 kişiden fazla insanla sosyal ilişki kuramayacağını söyler. (bu sayı gerçekte hayatta kalma baskısı altındaki gruplar için daha fazladır. İster inanın ister inanmayın, gerçekten Spotify'daki vaka bu şekilde değildir.) Gruplar çok büyüdüğünde, kısıtlayıcı kurallar, bürokrasi, politika, ekstra yönetim kademeleri ve diğer faydasız uğraşlar gibi durumlar görmeye başlıyoruz.

Bundan dolayı Tribe'lar 100 kişiden az olacak şekilde tasarlanmıştır.

Tribe'lar düzenli olarak toplantılar düzenler. Bu toplantılar resmi değildir. Squad'lar, Tribe'in geri kalanına (veya kimler geldiyse onlara) ne üzerinde çalıştıklarını, ne teslim ettiklerini ve şu an yaptıkları şeyden diğer ekiplerin ne öğrenebileceklerini gösterirler. Bu toplantılarda çalışan yazılımın canlı demosu yapılabilir, yeni araç ve teknikler tartışılabilir, ve mükemmel "Hack Günü" projeleri gösterilebilir.



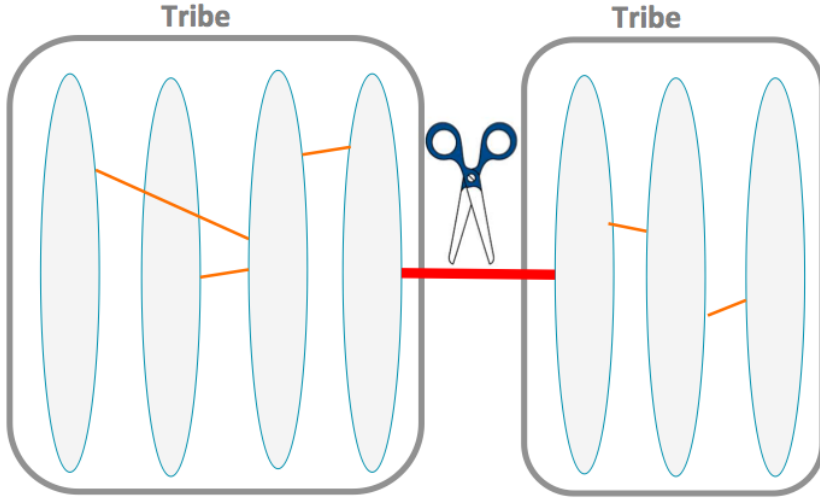
Squad Bağımlılıkları

Bir çok Squad'ın bulunduğu bir ortamda her zaman bağımlılıklar olacaktır. Bağımlılıklar mutlaka kötü değildir. Squad'lar bazen gerçekten mükemmel bir şey geliştirmek için birlikte çalışma ihtiyacı duyabilirler. Bütün bunlara rağmen, amacımız mümkün olduğu kadar özerk Squad'lara sahip olmaktır. Özellikle bir Squad'ı yavaşlatan veya bloklayan bağımlılıkları azaltmaya çalışıyoruz.

Bu duruma yardımcı olmak için, düzenli olarak Squad'larımıza hangi Squad'larımıza bağımlı olduklarını ve bu bağımlılıkların ne ölçüde bloklayıcı veya yavaşlatıcı olduğunu soruyoruz. Aşağıda bir örnek görebilirsiniz:

	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					

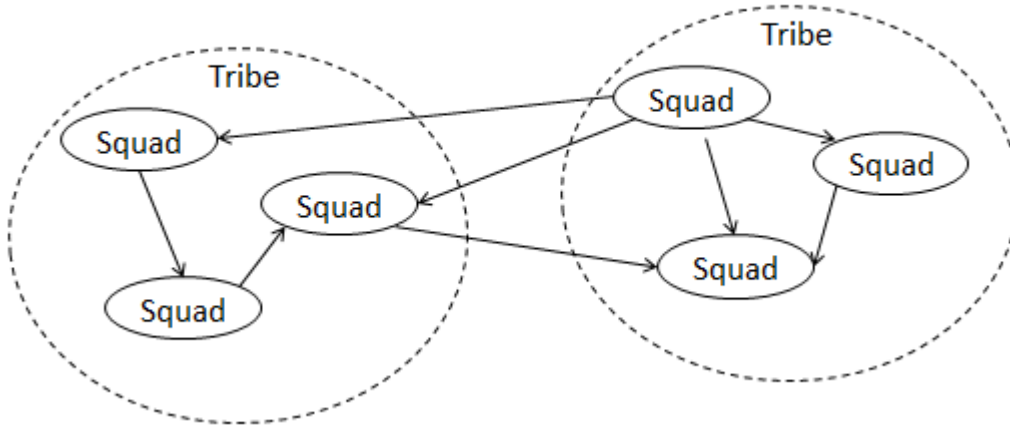
Sonrasında problemlili bağımlılıkları (özellikle bloklayan ve bir başka tribe ile ilgili olanları) ortadan kaldırmak için gerekli yöntemleri tartışıyoruz. Bu genellikle tekrar önceliklendirme, tekrar organize olma, mimari değişiklikler veya teknik çözümler ile sonuçlanıyor.



Anketler aynı zamanda Squad'ların diğerlerine nasıl bağımlılık gösterdikleri çerçevesindeki modeli görmemize yardımcı oluyor - örneğin Squad'lar, Operasyon ekipleri nedeniyle yavaşlıyor görünüyor. Zaman içerisinde hangi tip bağımlılıkların azalıp arttığını izlemek için basit bir grafik kullanıyoruz.

Scrum, "Scrum of Scrums" adı verilen bir pratiği barındırır. Bu, takımlar arasındaki bağımlılıkları tartışmak için her takımdan bir kişinin katıldığı bir senkronizasyon toplantısıdır. Biz Spotify'da genellikle "Scrum of Scrums" yapmıyoruz. Squad'ların bir çoğunun makul bir seviyede bağımsız olması nedeniyle bu tip koordinasyon toplantısına ihtiyaç duymuyoruz.

Bunun yerine, "Scrum of Scrums" ihtiyaç duyulduğunda gerçekleşiyor. Örneğin, yakın dönemde bir kaç ay boyunca bir çok Squad'ın koordineli çalışmasını gerektiren büyük bir projemiz vardı.

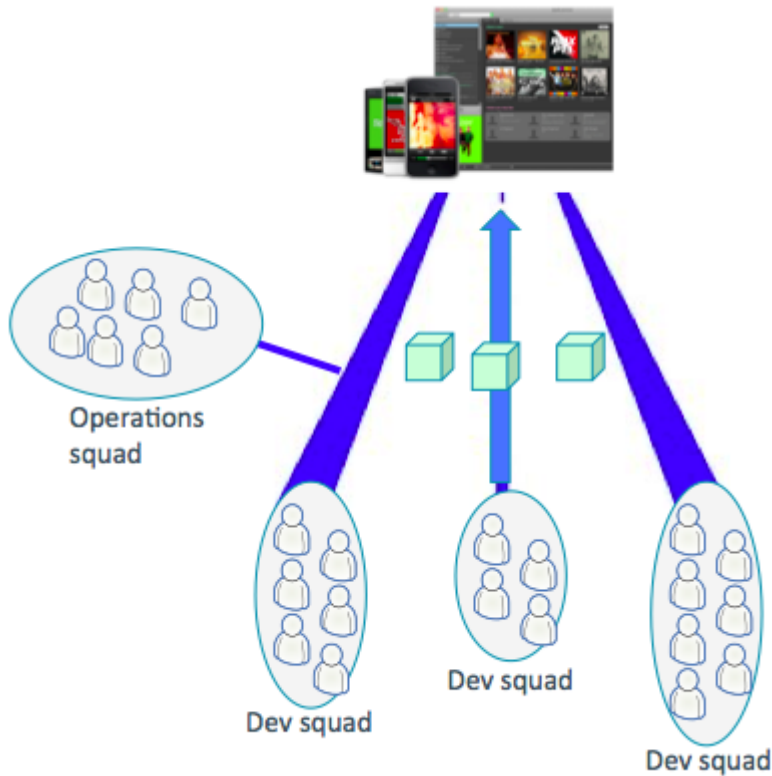


Bu durumun çalışmasını sağlamak için, takımlar, Squad'lar arasındaki bağımlılıkları belirledikleri ve çözdükleri günlük senkronizasyon toplantısı yaptılar. Ve pusulacık kullandıkları bir tahta ile çözülmemiş bağımlılıkları takip ettiler.



Çoğu şirketteki bağımlılık problemlerinin ortak kaynağı Geliştirme ve Operasyondur. Birlikte çalıştığımız bir çok firmada, geliştirme ile operasyon arasında, ilişkili uyumsuzlukları ve gecikmeleri ile birlikte, bir çeşit el değiştirme süreci vardır.

Spotify'da ayrı bir operasyon takımı var, ancak bu ekibin işi Squad'lar için yaygınlaştırma yapmak değildir. Bu takımın işi, Squad'lara yaygınlaştırmalarını kendi kendilerine yapabilmeleri için ihtiyaç duyulan desteği sağlamaktır. Sağladıkları destek altyapısal, script ve routine'ler şeklindedir. Operasyon takımı, "üretime giden yolu inşa eder".



Bu resmi olmayan ama etkili bir işbirliğidir, detaylı süreç dokümantasyonu yerine yüz yüze iletişim üzerine kuruludur.

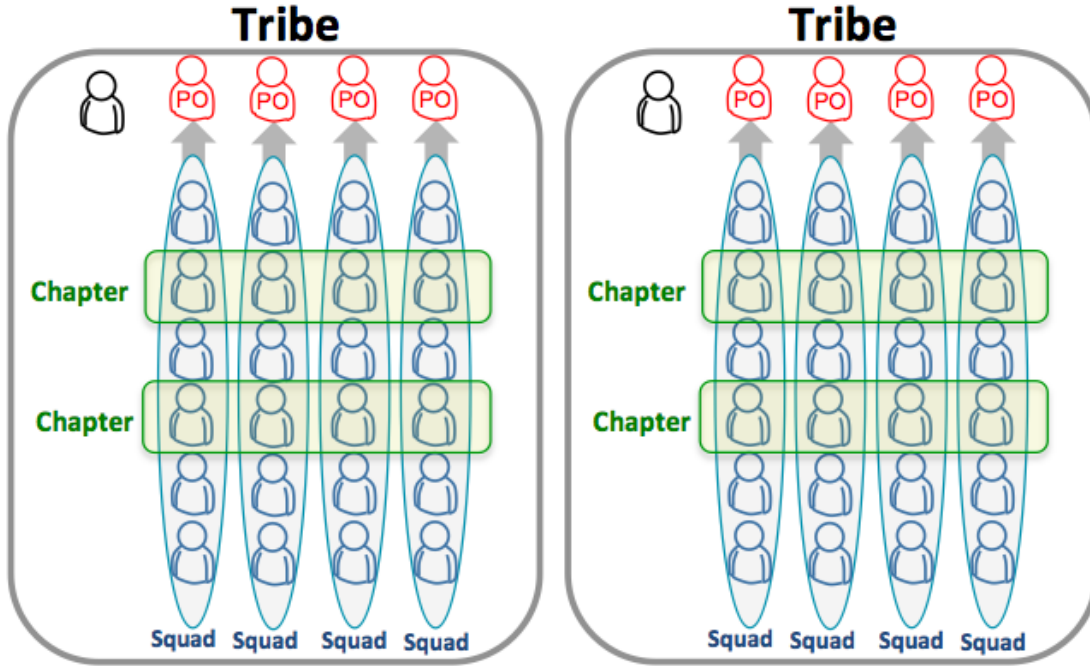
Chapter'lar ve Guild'ler

Herşeyin bir dezavantajı vardır, ve tam özerkliğin potansiyel dezavantajı, ölçek ekonomisi kaybıdır. Squad A'daki bir test mühendisi, Squad B'deki test mühendisinin geçen hafta çözdüğü bir problemle boğuşuyor olabilir. Eğer Squad'lardaki ve Tribe'lardaki tüm test mühendisleri bir araya gelebilirse, bilgi paylaşımı yaparlar ve tüm Squad'ların yararlanabileceği araçlar yaratabilirler.

Eğer her Squad tam özerk olsaydı ve diğer Squad'lar ile hiç iletişimi olmasaydı, o zaman ortada bir şirketin olmasının anlamı ne olurdu? Spotify bile 30 farklı küçük şirkete bölünebilirdi.

Bundan dolayı, Chapter'lar ve Guild'lere sahibiz. Bu şirketi bir arada tutan yapıştırıcıdır. Bu bize çok fazla özerklik feda etmeden, bir miktar ölçek ekonomisi veriyor.

Chapter, aynı Tribe içerisindeki benzer yeteneklere sahip ve aynı uzmanlık alanında çalışan kişilerden oluşan küçük ailedir.

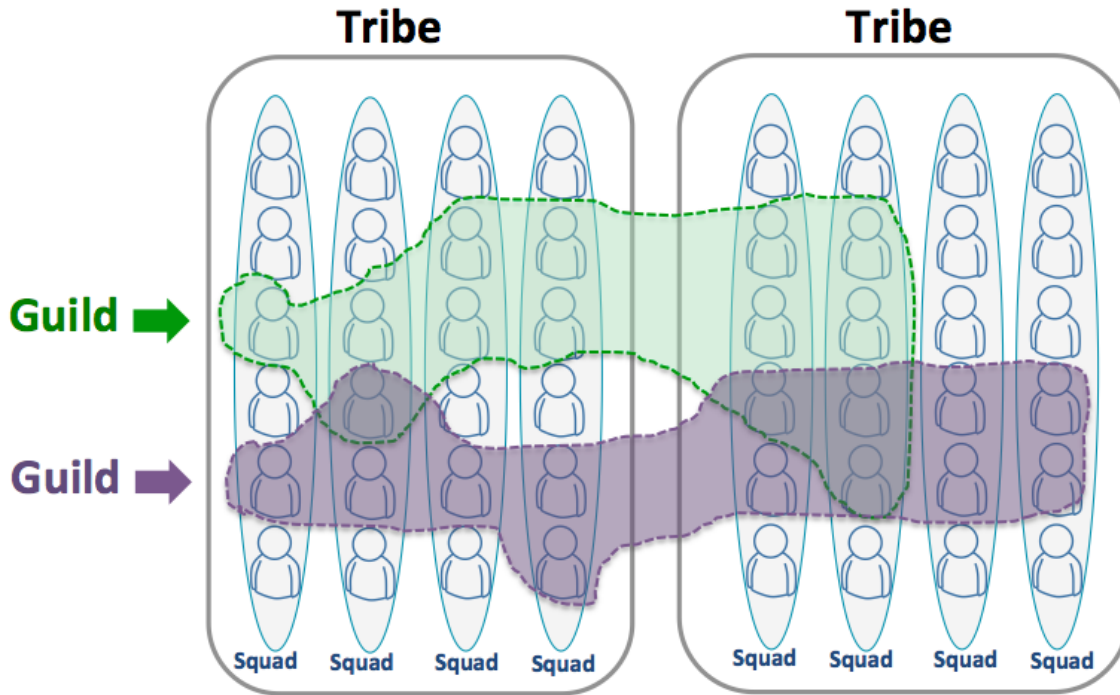


Her Chapter düzenli olarak bir araya gelerek, kendi uzmanlık alanı ile ilgili ve belirli problemleri üzerinde tartışır. Örneğin Test Mühendisleri Chapter'ı, Web Geliştiricileri Chapter'ı, veya Backend Chapter'ı gibi.

Chapter'ın lideri, kişileri geliştirmek, maaşlarını ayarlamak vb. gibi geleneksel sorumluluklara sahip olan ve bu alanda uzman olan müdürlerdir. Ancak, Chapter lideri aynı zamanda bir Squad'ın parçasıdır ve gerçekten kopmamasına yardımcı olmak için günlük işlere de dahil olur.

Artık, gerçek, yukarıdaki sevimli resimlerden her zaman daha karmaşıktır. Örneğin, Chapter üyeleri Squad'lara eşit olarak dağılım göstermemektedir. Bazı Squad'lar bir çok Web Geliştirici sahibiyken, bazılarında hiç yoktur. Ancak yukarıdaki resim size genel fikri vermiştir.

Guild, daha organik ve daha geniş katılımlı; bilgi, araçlar, kod ve pratik paylaşımı yapmak isteyen insanlardan oluşan bir “İlgi Grubu” dur. Guild’ler genellikle bütün organizasyonu kapsarken, Chapter’lar her zaman, Tribe lokalinde olan bir gruptur. Bazı Guild örnekleri: Web Teknoloji Guild’i, Test Mühendisleri Guild’i, Agile Coach’lar Guildi vb.



Bir Guild genellikle o alanda çalışan tüm Chapter ve üyelerini içerir. Örneğin Test Mühendisleri Guild’i, tüm Test Mühendisliği Chapter’larındaki Test Mühendislerini içerir. Ancak herkes ilgi duyduğu guild’e katılabilir.

Her Guild’in bir Guild Koordinatörü vardır ve bu kişi sadece bu işi yapar :o).

Bir Guild’in yaptığı işe örnek olarak, daha yakın zamanda “Web Guild Unkonferans”ı düzenledik. Bu etkinlik açık ortam etkinliği olarak gerçekleştirildi ve Spotify’daki tüm Web Geliştiriciler Stockholm’de bir araya gelerek, kendi alanlarındaki zorlukları ve çözümleri tartıştılar.



Bir diğ er  rnek Agile Coach Guildi. Coach'lar t m organizasyona dağılmıřtır, ama s rekli olarak bilgi paylařırlar ve " yileřtirme Tahtası"nda takip ettiğimize  st seviye organizasyonel iyileřtirme alanları konusunda iřbirliğı yapmak i in d zenli olarak toplanırlar.



Bir saniye, Bu bir Matriks Org. Değil mi?

Evet. Yani, bir  eřidi. Bu, bir  oğumuzun alıřkın olduėundan, farklı bir maktriks organizasyon  eřididir.

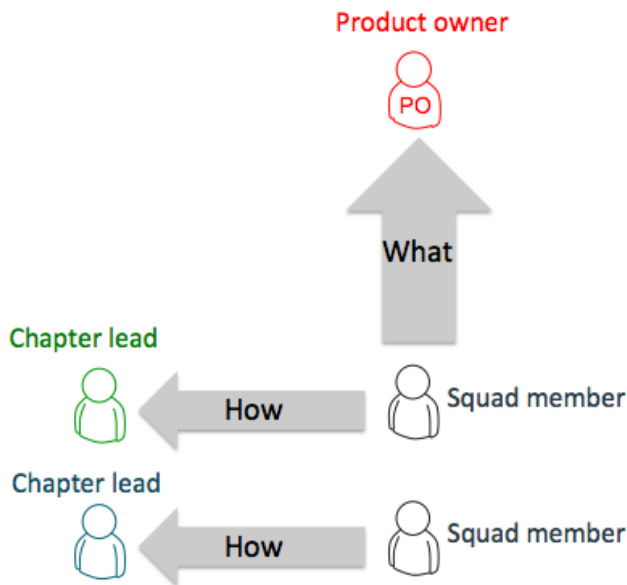
Bir çok matriks organizasyonda, benzer yeteneklere sahip insanlar fonksiyonel bölümlerde havuz içerisinde değerlendirilir ve projelere atanırlar ve bölüm yöneticilerine rapor verirler.

Spotify nadiren böyle bir şey yapar. Bizim matriks yapımız, ağırlıklı olarak teslimata yönelmiştir.

İnsanlar aynı yerde oturan, sabit Squad'lar şeklinde gruplanmıştır. Bu gruplarda farklı yeteneklere sahip insanlar işbirliği ve kendiliğinden organize bir şekilde mükemmel bir ürün ortaya çıkarmaya çalışırlar. Bu matriksteki dikey boyuttur. Fiziksel olarak insanların gruplandığı ve zamanlarının çoğunluğunu geçirdikleri Squad'lar olduğu için, bu birincil yapıdır.

Yatay boyut, bilgi, araç ve kod paylaşımı içindir. Chapter liderinin görevi bunu desteklemek ve olanak sağlamaktır.

Matriks terminolojisinde, dikey düzlemi "Ne" ve yatay düzlemi "Nasıl" olarak düşünün. Bu matriks yapısı, her Squad üyesinin "Bir şeyi nasıl iyi geliştirebileceği" nin yanı sıra "Bir sonra ne geliştirileceği" konusunda da yardım almasını kesinleştirir.

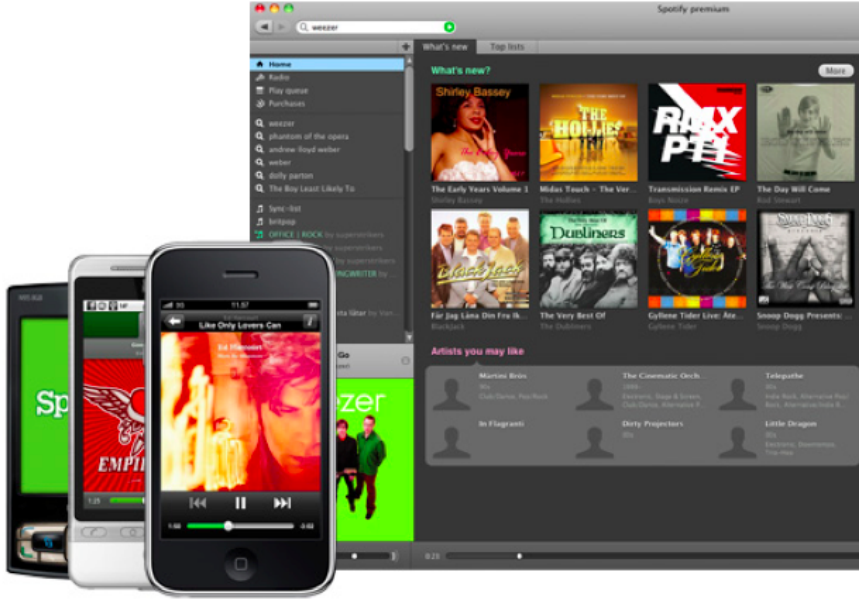


Bu, Mary ve Tom Poppendieck tarafından önerilen "Profesör ve Girişimci" modeliyle uyum gösterir. Chapter lideri, teknik mükemmeliyete odaklanmış "Profesör" veya "Yetkinlik Lideri" iken, Ürün Sahibi, mükemmel ürün teslimatına odaklanmış "Girişimci" veya "Ürün Şampiyonu"dur.

Bu roller arasında sağlıklı bir gerilim vardır. Girişimci, köşeleri kesmekle ve hızlanmakla meşgul olurken, Profesör yavaş giderek, işlerin uygun geliştirilmesi ile meşgul olmaktadır. Bu iki yönde gereklidir, bundan dolayı "sağlıklı" bir gerilimdir.

Peki ya Mimari?

Spotify teknolojisi yüksek derecede servis tabanlıdır. 100'ün üzerinde ayrı sisteme sahibiz ve her biri ayrı ayrı yaygınlaştırılabilir ve idame ettirilebilir. Bu sistemler, Oynatma Listesi Yönetimi, Arama veya Ödeme gibi backend servislerini içerdiği gibi son iPad oynatıcı gibi son kullanıcı servislerini de içermektedir. Ayrıca radyo, müzik oynatıcının “yeni ne var” bölümü gibi belli bileşenleri de içermektedir.



Teknik olarak herkes her sistem üzerinde işlem yapabilir. Squad'lar etkin yetenek takımları olduğundan, normal olarak canlıya yeni yetenek yaygınlaştırmaları için bir çok sistemde güncelleme yapma ihtiyacı bulunmaktadır.

Bu modelin riski, hiç kimse sistemin bir bütün olarak entegre olmasına odaklanmaz ise, sistemin altyapısının bozulmasıdır.

Bu riski azaltmak için, “Sistem Sahibi” dediğimiz bir rol bulunmaktadır. Bütün sistemlerin bir sistem sahibi veya iki sistem sahibi (eşleşmeyi özendiriyoruz) vardır. Operasyonel kritik sistemler için, Sistem Sahibi, bir Dev-Ops eşidir, bu kişilerden birisi geliştirme yaklaşımına sahipken, diğeri operasyon yaklaşımına sahiptir.

Sistem sahibi, sistemle ilgili teknik veya mimari problemlerde gidilecek kişi(ler)dir. Bu kişi koordinatördür ve sistem üzerinde kod geliştiren kişilerin, sistemin dengesini bozmadığından emin olmaları için yol gösterir. Bu kişi, kalite, dokümantasyon, teknik borç, kararlılık, ölçeklenebilirlik ve yaygınlaştıra süreci gibi konulara odaklanır.

Sistem Sahibi bir darboğaz veya hayaller aleminde yaşayan bir mimar değildir. Tüm kararları kendi başına almak, kodu geliştirmek veya yaygınlaştırmaları yapmak zorunda değildir. Sistem sahipliği yanında, tipik olarak günlük sorumlulukları olan bir Squad üyesidir veya Chapter lideridir. Ancak, zaman zaman bu kişi bir “Sistem Sahibi Günü” yapar ve sistem üzerinde çalışır. Normal olarak, sistem sahipliği işini toplam zamanın %10'undan az tutmaya çalışıyoruz. Ancak bu bir çok sistem olmasından dolayı değişkenlik gösteriyor.

Ayrıca Şef Mimari rolüne de sahibiz. Bu kişi bir çok sistemi etkileyen üst seviye mimari problemleri işinin koordinasyonunu yapar. Ortak hataların engellenmesi için yeni sistemlerin geliştirmelerini gözden geçirir. Aynı zamanda bu sistemlerin, mimari vizyonla aynı doğrultulda olduğundan emin olur. Geri bildirim her zaman girdi veya öneriler şeklindedir. Sistemlerin son tasarım kararı halen Squad'ların kararıdır.

Bunların hepsi nasıl çalışıyor?

Spotify çok hızlı büyüyor – 3 yıl içerisinde teknik birimlerde 30'dan 250 kişiye çıktık. Dolayısı ile bu büyümenin acısını yaşıyoruz. Bu Squad, Tribe, Chapter ve Guild'lerden oluşan ölçeklendirme modeli, geçen yıl oluşturulmuştur. Bu nedenle insanlar hala alışmaya çalışıyorlar. Ancak şimdiye kadar, yapılan anketler ve süreç gözden geçirmelerin sonucuna göre, bu ölçeklendirme modelinin çalıştığını gözlemliyoruz. Bu bize içine sığabileceğimiz bir yapı veriyor. Hızlı büyümeye rağmen, çalışan mutluluğu sürekli artıyor (Nisan 2012'de 5 üzerinden 4.4'tü).

Ancak, her büyüyen organizasyonda olduğu gibi, bugünün çözümleri, yarının problemlerinin doğmasına neden oluyor. Bu nedenle bizi izlemeye devam edin, hikaye henüz bitmedi :o)

/Henrik & Anders

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

anders.ivarsson@spotify.com

* Bu çeviri ACM Agile tarafından yapılmıştır.

Barış BAL

baris.bal@acmagile.com

www.acmagile.com