

# Açıklamalı SOA Manifestosu

Thomas Erl tarafından SOA Manifestosunun içyüzü hakkında yorumlar

**Servis yönelimi, yaptığınız seyleri çevreleyen bir paradigmadır. Servis-yönelimli mimari (SOA) ise servis yöneliminin uygulanması sonucu ortaya çıkan bir mimari tipidir.**

İlk baslarda, bu manifesto birbirinden ayrı, ancak birbiri ile yakından alakalı iki konu hakkında olduğu düşünülürdü: servis yönelimli mimari modeli ve mimarinin tanımlandığı bir paradigma olan servis yönelimi. Manifestonun biçimi Çevik Manifestodan sonra modellenmiştir. Bu biçimin içeriği, tutkuları ve değerleri hayata geçirebilmek için açıklayan yol gösterici kısa ifadelerden oluşur. Bu manifesto bir sartname, bir referans model ya da bir beyaz sayfa değildir. Bu önsöz, gerçek tanımlarına değinmeden, varolan terimlerin manifesto belgesinde nasıl ve neden bahsedildiğini aydınlığa kavuşturmak için eklenmeye karar verdik.

## **Servis yönelimi uyguluyoruz...**

Servis yönelim paradigması, en doğru şekilde söylemek gerekirse, stratejik amaç ve faydalar olarak da tanımlanabilecek belli bir hedef durumu gerçekleştirmek için kullanılan bir yöntem ya da yaklaşım olarak görülebilir. Servis yönelim uyguladığımızda, bu hedef durumu gerçekleştirmek uğruna yazılımları ve teknoloji mimarilerini şekillendiririz. Bu, teknoloji mimarilerini servis yönelimli olarak tanımlamamızın nedenidir.

## **...artan çeviklik ve maliyet verimliliği ile sürekli sürdürülebilir iş değeri üreten organizasyonlara yardım edebilmek amacı ile...**

Giriş cümlesinin devami olan bu satır, servis yönelimli yazılımların bazı belirgin ve genel kabul görmüş stratejik faydalarını vurgulamaktadır. Bu faydaları anlamak, servis yöneliminin uygulanması sonucu gerçekleşen yukarıda bahsedilen hedef duruma biraz daha ışık tutacaktır.

İş seviyesinde çeviklik, organizasyonların cevap verebilme yetenekleri ile karşılaştırılabilir. Bir organizasyon iş değişikliklerine ne kadar kolay ve etkili cevap verebilirse, değişimin etkilerine o oranda etkin ve başarılı şekilde uyum sağlayabilir.

Servis yönelimi, servisleri, geliştirilip yayınlanması için gerekli ilk yatırımın çok üzerinde ve sürekli bir değer katması beklenen, bir bilimsel varlığı olarak konumlandırır. Maliyet verimliliği, beklenen yatırım karlılığı ile doğrudan ilişkilidir. Bir çok şekilde, maliyet verimliliğindeki artma çeviklikteki artma ile el ele vermiş gibidir; eğer mevcut servisleri tekrar kullanabilme fırsatı bulunabilirse, yeni çözümler getirebilmek için gerekli olan masraflar genel olarak düşecektir.

Sürdürülebilir iş değeri, servis yöneliminin uzun vade hedefleri ile ilişkilidir. Sürekli olarak yeni çözümlere parçalanarak dönüşebilme esnekliğine sahip olarak gelişen bu yazılımlar, değişken iş gereksinimlerine uyum sağlayabilmek adına evrimleşebilir.

### **...degisen is ihtiyaçlari ile örtüsen bir sekilde**

Bu ifadeler, servis yönelimli yazılım felsefesini anlayabilmenin anahtarlarıdır. Süregelen bir düzende iş değişimine uyum sağlama ihtiyacı, servis yönelimi için temeldir ve herşeyi kucaklayan stratejik bir hedeftir.

### **Çalışmamız ile önceliklerimizi söyle belirledik:**

Birazdan gelecek ifadeler, daha değerli olduğu düşünülenlerin önceliklendirildiği bir grup değeri belirtmektedir. Bu değerlendirme sisteminin amacı, servis yönelimli hesaplamaların stratejik hedef ve faydaları ışığında düzenli olarak yapılması gereken zorlu seçimleri adreslemektir.

### **Teknik strateji yerine iş değerine**

Biraz evvel belirttiğim gibi, iş değişikliklerine uygun hale gelebilmek kapsayıcı stratejik bir hedeftir. Bu nedenle, servis yönelimin uyarlanmasının sonucu olan servis yönelimli mimari ve her tür yazılımın, çözüm ve ekosistemin kalitesi iş odaklı olması ile yakından ilişkilidir. İşin yönünü belirleyen şey teknoloji değil, teknolojiyi en iyi şekilde kullanmayı amaçlayan iş vizyonudur.

Bu önceliklendirme, bilimsel kuruluşlarında dalgalanma etkisi yaratır. Nasıl planlayıp otomasyon çözümlerimize kaynak bulacağımızdan, nasıl geliştirip yöneteceğimize kadar proje geliştirme ve yayınlama sürecinin her parçasında değişikliklere neden olur.

### **Projeye özel fayda yerine stratejik hedeflere**

Birçok BT projesi, yalnızca günümüzde mevcut olan iş süreçleri gereksinimlerini otomatize edecek şekilde tasarlanmış uygulamalar geliştirmeye odaklanır. Böylece mevcut (taktiksel) ihtiyaçlar gerçekleştirilir. Ancak tek bir amaca hizmet eden uygulamaların sayısı arttıkça, BT kurumları mantık ve veri adalarından oluşan uygulama "silo"ları ile dolar. Yeni gereksinimler geldikçe, ya yeni silolar yaratılır, ya da silolar arası entegrasyon kanalları oluşturulmaya çalışılır. Ve iste daha fazla değişiklik oluştukça entegrasyon kanalları artırılır, hatta yeni silolar da eklenir. Böylece BT kurumsal manzara daha karmaşık, daha külfetli, daha pahalı ve evrilmesi daha yavaş olur.

Birçok açıdan servis yönelimi bu tür sorunlara cevap olarak ortaya çıkmıştır. Bu paradigma, inatla uzun vadeli stratejik hedeflerin başarılmasını önceliklendirerek, proje bazlı, silo bazlı ve entegre uygulama geliştirmeye bir alternatif getirir. Savundüğümüz servis yönelim ile ortaya çıkan durum, geleneksel uygulama siloları olmak zorunda değildir. Hatta servis yönelim uyguladığımız ortamlarda bulunan eski ve köhne kaynak ve uygulamalar olsa bile, hedeflenen durum bunların daha uyumlu çalışmasıdır.

### **Özelleşmiş entegrasyonlar yerine gerçek anlamda birlikte çalışabilirliğe**

Veri paylaşan yazılımlar beraber çalışabilmelidir. Eğer yazılımlar uyumlu olacak şekilde tasarlanmazsa, büyük bir olasılıkla beraber de çalışmazlar. Birbiri ile

uyumsuz yazılımlar, arasındaki işbirliğini arttırmak için entegre olmaları gerekir. Bu nedenle entegrasyon birbirinden tamamen farklı yazılımlar arası işbirliğini arttırmak için yapılması gereken çabadır.

Gerçi çoğu zaman gerekli olsa da, özelleştirilmiş entegrasyonlar pahalıdır, zaman isterler ve gelişmesi külfetli kırılgan mimarilere neden olurlar. Servis yönelimin bir hedefi de birbiri ile doğal yollarla uyumlu olmayan yazılımları şekillendirerek gereken özelleşmiş entegrasyon ihtiyaçlarını en aza indirmektir. Servis yönelim paradigması, gerçek anlamda birlikte çalışabilirliğe erişebilmek için birçok tasarım şablonunu ile çevrelemiştir.

Yazılımları karakteristiği olan gerçek anlamda birlikte çalışabilirlik, artan maliyet verimliliği ve çeviklik gibi stratejik faydaları gerçekleştirmek için anahtar rol oynar.

### **Özel amaçlı uygulamalar yerine paylaşımlı servislere**

Biraz evvel açıkladığım gibi, servis yönelimi birkaç tasarım şablonundan oluşan bir tasarımsal yaklaşımdır. Anlamlı şekilde uygulandığında, bu prensipler yazılımınızı mesru olarak servis olarak adlandırabileceğimiz servis yönelimli mantıksal birimlere dönüştürür.

Servisler, (ki bazıları gerçek anlamda birlikte çalışabilirliği sağlarlar) daha önce bahsettiğimiz hedef durumu destekleyen somut özelliklerle bezenmiştir. Bu özelliklerden bir tanesi, farklı iş süreçlerinin otomasyonunu desteklemek için paylaşılabilir ve tekrar kullanılabilir, bir çok amaca hizmet eden iş mantıklarını kapsamaktadır. Bir paylaşımlı servis kendisini, harcamaları ve yeni otomasyon çözümleri için gerekli eforu düşürürken, düzenli olarak iş değeri üreten bir varlık olarak ortaya koyar. Her ne kadar planlanmış iş gereksinimlerini çözen geleneksel, tek amaçlı uygulamaların değeri olsa da, paylaşımlı servisleri kullanmak servis yönelimli hesaplamaların stratejik hedeflerini gerçekleştirmesi açısından (ki bu maliyet verimliliği ve çevikliğin artması demektir) daha büyük bir değer üretir.

### **Optimizasyon yerine esnekliğe**

Bu muhtemelen değer verdiğimiz öncelikler arasında en geniş olanıdır. Bu, tekli ve çoklu servislerin gelişmesi ve yayınlanması sırasında ortaya çıkan düşünceleri en iyi şekilde önceliklendirmemizi sağlayan yol gösterici felsefe olarak da görülebilir.

Optimizasyon öncelikli olarak uygulamanın tasarımında ince ayarlar yaparak ya da ani ihtiyaçların karşılanmasını hızlandırarak planlı kazanımların gerçekleştirilmesi olarak adlandırılabilir. Esnekliğin sağlanması için uygun önceliklendirme yapılmadığında daha önce bahsedilen silo tabanlı ortamların oluşmasına neden olması dışında, bunda hoş karşılanmayan bir şey yok. Örneğin, esneklik özelliği, servislerin verimli (ve gerçek anlamda) veri paylaşılmasının ötesindedir. Gerçek manada değişen iş gereksinimlerine karşılık verebilmek için, servisler nasıl bileşik çözümler oluşturabileceğini bilecek şekilde esnek olmak zorundadır. Öğelerine parçalanabilmesi gerçeğine rağmen çoğunlukla daha statik olan geleneksel dağıtık uygulamalardan farklı olarak, servis bileşenleri sürekli artan bir esneklik ile tasarlanması gereklidir. Bu, ne zaman mevcut iş süreçleri değişse ya da yeni süreçler eklense, asgari entegrasyon eforu harcayarak bileşen tabanlı mimari çerçevesinde yeni servisler ekleyebilmemiz, eskileri çıkartabilmemiz ve mevcutları

genisletebilmemiz demektir. Bu nedenle, bileşenlerine ayrılabilen servisler, anahtar öneme sahip servis yönelimli tasarım prensiplerinden biridir.

### **Baslangıçtaki mükemmellik arayışı yerine evrimsel iyileştirmeye**

Servis yönelim ile alakalı “çeviklik” terimine geldigimizde genelde kafamız karışıyor. Bazı tasarım yaklaşımları anlık kazanımlar için yazılımın hızlı bir şekilde geliştirilip yayınlanmasını savunurlar. Bu, taktiksel ve kısa vadeli faydaya odaklandığı için “taktiksel çeviklik” olarak düşünülür. Servis yönelimi, organizasyonel veya iş seviyelerinde çevikliğin başarılması gerekliliğini savunur. Bunda, bir bütün olarak değişimlere cevap verebilmek için organizasyonları güçlendirme niyeti vardır. Bu tür bir organizasyonel çeviklik “stratejik çeviklik” olarak adlandırılır. Bunu nedeni olarak, geliştirip yayınladığımız her yazılım ile uzun vadeli stratejik değerlere hizmet eden çevik bir ortamda çalışmak gösterilebilir.

BT kurumlarının, organizasyonel çevikliği başarabilmek için müşteri ile koordine bir şekilde evrimleşmesi gerekir. Biz genellikle işin zamanla nasıl gelişeceğini tahmin edemeyiz. Bu nedenle ilk anda mükemmel servisler geliştiremeyiz. Aynı zamanda, yoğunlukla SOA projelerinin analizi ve modellenmesi aşamasında kullanılmaya başlanan organizasyonların mevcut iş aklında geniş ve zengin bir bilgi birikimi de vardır.

Bu bilgi, servis yönelim prensipleri ve ispatlanmış yöntemler ile birlikte zaman içinde işin nasıl değiştiğini görüp ona uyum sağlarken, bugün işin nasıl var olduğunu ve nasıl çalıştığını yakalayan yeni servisleri bulmamıza ve tanımlamamıza yardımcı olur.

### **Soldaki maddelere değer vermekle beraber, sağdaki maddeleri daha değerli bulmaktayız**

Bu değerlerin nasıl önceliklendirildiğini araştırdıkça, servis yönelim ile diğer paradigmalardan arasındaki farkları daha iyi anlıyoruz. Bu tür bir kavrayış bilgisayar çalışanlarına çeşitli şekillerde yardımcı olabilir. Örneğin, verilen organizasyon ya da BT kurumunda servis yöneliminin uygunluğunu anlamamız için gereken temel kriterleri sağlayabilir. Hangi servis yöneliminin uygulanabilir olduğunu, hangisinin uygulanması gerektiğini farketmemizde yardımcı olabilir.

Ayrıca, ana değerlerin anlaşılması bizim için belli ortamlarda SOA projeleri başarmanın ne kadar zorlayıcı olduğunu anlamamıza da yardımcı olabilir. Örneğin, buradaki bazı öncelikler, mevcut inanç ve tercihlerimizle kafa kafaya çarpışabilir. Bu durumda, servis yöneliminin faydaları, efor ve (sadece teknoloji üzerinde değil, organizasyon ve BT kültüründe) uyum sağlama üzerindeki etkisine bakarak değerlendirilmelidir.

Birazdan gelecek rehber prensipler, bahsedilen zorlukları adreslemek için oluşturulmuştur.

### **Su ilkeleri takip ediyoruz:**

Su ana kadar, manifesto ile bir vizyon ve vizyon ile ilişkili temel değerlerin altı çizildi. Bildirinin geri kalanı değerlere bağlı kalmak ve vizyonu gerçekleştirmek için gerekli olan rehberliği sunacak prensiplerden oluşmaktadır.

Unutulmamalıdır ki, bu rehber prensipler manifestoya özeldir. Servis yönelim tasarım paradigmalardan oluşan, tamamen farklı bir takım tasarım prensipleri ve servis yönelimi ve servis yönelimli mimariye özel belgelenmiş daha birçok pratik ve sablon halihazırda vardır.

### **Organizasyonların sosyal ve güç yapılarına saygı gösterin**

SAO'nin düstüğü yaygın tuzaklardan biri, benimsenme sürecini teknoloji merkezli bir inisiyatif olarak algılamaktır. Yapıldığında, kaçınılmaz organizasyonel etkilerine hazırlıksız yakalanmamız nedeniyle hemen hemen her zaman başarısızlığa neden olur.

Servis yöneliminin benimsenmesi, iş süreçlerini otomatikleştirme seklimizin dönüşümü ile ilgilidir. Ancak, bu dönüşümü gerçekleştirmek için ne kadar plan yaparsak yapalım, organizasyonu, yapısını, hedeflerini ve kültürünü kavramaya çalışmaktan başlamalıyız. Servis yönelimin benimsenmesi tamamen insanların deneyimleri ile ilgilidir. Yetkili kişilerden desteğe ihtiyaç duyar, ve sonra bilisim, kültürde stratejik, topluluk merkezli bir zihniyetin oluşmasını ister. Bu seviye bir organizasyonel değişimi tanımalı ve planlamalıyız, ki servis yönelimi başarabilmek için gerekli, uzun vadeli taahhütler verilebilsin.

Bu tür düşüneler, bizlerin sadece SOA inisiyatifi ile en iyi şekilde nasıl devam edebileceğimizi anlamamızı sağlamaz, ayrıca en doğru kapsam ve yöntemi tanımlamamıza da yardımcı olur.

### **SOA'nin eninde sonunda birçok seviyede değişiklik talep edeceğinin farkında olun**

Bir söz vardır: "Basarı fırsatlara hazır olmaktır". Sanırım SOA projelerinden edindiğim bir numaralı ders, servis yönelimi sonucunda oluşacak değişikliğin hacmi ve genişliğini tamamen anlamamız, planlamamız ve hazırlıklı olmamız zorunludur. İşte birkaç örnek.

Servis yönelimi, yazılımları uzun vadeli, tekrarlayan iş değeri sahibi varlıklar olarak tanımlayarak, onları geliştirme seklimizi değiştirir. Yapılacak ön-incelemeler bu varlıkları içeren ortamları oluşturmak için, süregelen taahhütler ise değerlerini korumak ve kollamak için gereklidir. Özün sözü, değişim her halükarda BT kurumlarındaki sistemlere kaynak bulmak, ölçmek ve bakımını yapmak için gereklidir.

Buna ek olarak, servis yönelimin kurum içinde kaynak olarak yer bulmasının asıl nedeni, sistemin farklı parçalarında olan değişiklikler ve tasarım ve kullanımdaki düzenlemelerdir. Altyapıda sürekli ölçeklenebilirlik ve güvenilirliği garanti etmek için gerekli değişiklikleri dile getirmek asla değildir.

### **SOA farklı şekillerde benimsenebilir. Çabaları yönetilebilir ve sınırlar içinde anlamlı değerlerde tutun**

Yaygın efsanelerden biri de servis yönelimli hesaplamayı gerçekleştirebilmek için servis yönelimin kurum seviyesinde benimsenmiş olması gerekliliğidir. Bu, tasarım ve endüstri standartlarının kurum içinde yerleşmesi ve kullanılmasına zorlanması

demektir. Her ne kadar bu fikirde bir yanlışlik yoksa da, özellikle büyük çaplı birçok organizasyonlar için gerçekçi bir hedef değildir.

SAO'nin benimsenmesi açısından en uygun kapsam, pragmatik tartışmalar ışığında yapılan plan ve analiz sonucunda belirlenmelidir. Bu tartışmalar, yetki alanları ve gelişen kültürel değişiklikler gibi organizasyonel yapıda oluşan etkiler üzerinde olabilir.

Bu tür etkenler, bizim yönetilebilir bir dönüşümün kapsamını belirlememize yardımcı olur. Bilisim kurumunu umduğumuz stratejik duruma dönüştürmek için gerekli her efor için, kapsam anlamlı da olmalıdır. Bir başka deyişle, kapsam silolar arasında da anlamlı olmalı. Böylece birbiri ile ilişkili servis grupları, belirlenmiş sınırlar içinde geliştirilip yayınlanabilir. Yine bir başka deyişle, biz "servis kitaları" oluşturmak istiyoruz, "servis adaları" değil.

Kurum içinde birbirinden bağımsız geliştirilen ve yönetilen servis öbekleri oluşturma kavramı, yaygın olarak bilinen ismiyle "bing-bang" SOA projelerinin riskini azaltırken, organizasyonel ve teknik değişikliklerin etkilerini de azaltır. Bir seferde tek servis öbeğinin oluşması nedeniyle, bu bir kademeli dönüşüm yöntemidir.

## **Ürünler ve standartlar size ne SOA'yi, ne de servis yönelim paradigmasını verir**

Bu prensip iki farklı birbiri ile ilişkili efsane ile ilgilidir. İlki, SOA'yi modern teknolojik ürünler kullanarak satın alabilirsiniz der, ikincisi ise (XML, WSDL, SCA gibi) endüstri standartlarına geçişin doğal bir sonucu olarak SOA dönüşümün başarılacağına düşünür. Ürün sağlayıcı/satıcı ve endüstri standartları toplulukları tescilli olmayan framework'ler ve platformlar ile modern servis teknolojilerinde yenilikler geliştirebileceklerini sanmaktadırlar. Sanallastirmeden bulut hesaplamaya ve grid hesaplamaya kadar herşey, bize gelişmiş ve karmaşık servis yönelimli potansiyel çözümlerin gelişimi hakkında yardımcı olur. Ancak bilinmelidir ki, bu teknolojilerden hiçbiri SOA'ya özel değildir. Silo tabanlı sistemleri -kendi özel sunucularınızda olduğu gibi- bulut üzerinde rahatlıkla oluşturabilirsiniz.

"Kutusunda SOA Çözümü" gibi birşey yoktur. Servis yönelimli mimariye sahip olabilmek için, servis yönelimi başarılı bir şekilde uygulanmalıdır. Bu, tasarladığımız ve geliştirdiğimiz herşeyin tek bir yön, vizyon ve iş gereksinimi tarafında yönlendirilmesini gerektirir.

## **SOA çok çeşitli teknoloji ve standart kullanarak gerçekleştirilebilir**

Servis yönelimi teknoloji-bağımsız ve satıcı/sağlayıcı-bağımsız bir paradigmadır. Servis yönelimli mimari ise teknoloji-bağımsız ve satıcı/sağlayıcı-bağımsız bir mimari modeldir. Servis yönelimli hesaplama dağıtık hesaplamanın özel bir türü olarak görülebilir. Bu nedenle servis yönelimli çözümler ancak dağıtık hesaplamaya uygun teknolojiler ve endüstri standartları ile geliştirilebilir.

Bazı teknolojilerin (özellikle endüstri standartları temelli olanların) servis yönelimi tasarım prensiplerini uygulayabilme potansiyelinin artmasına rağmen, en uygun teknoloji ve endüstri standardını aslında iş gereksinimlerini gerçekleştirme potansiyeli belirler.

## **Endüstri temelli, genel geçer kabul gören, topluluk standartlarına uygun kurumsal standartlar ve politikalar geliştirin**

Endüstri standartları, kişiye ya da kuruma özel olmayan, mimari için kabul görmüş temel niteliklerin (veri nakli, arayüz ve mesaj formatı gibi) oluşmasına yardımcı eden teknoloji sarnameleridir. Endüstri standartlarını kullanmak servislerin karşılıklı çalışabilir olmasını garanti etmez.

İki yazılımın tamamen uyumlu olabilmesi için, (veri modeli ve ilkeleri gibi) ek kurallarına bağlı kalmak gerekir. Bu nedendir ki, bilişim sektörü tasarım standartlarını oluşturmali ve kullanmaya zorlamalıdır. Verilen bir alan içinde, standartlaştırma ve standartlaştırma süreçlerini düzenlemede yaşanan sıkıntılar, birçok stratejik faydanın dayandığı karşılıklı çalışabilme özelliğinde yıkıma neden olmaya başlar.

Bu prensip sadece kurumsal tasarım standartlarını savunmaz, ayrıca bizlere ne zaman mümkün ve olası olursa, özelleşmiş tasarım standartlarının genel olarak endüstri ve topluluklarda halihazırda kullanılan standartları içermesi gerektiğini de savunur.

## **İçeride çeşitliliğe izin verirken dışarıda benzerliğin pesinde kosun**

Birleşiklik birbirinden tamamen farklı varlıkların birleşmesine denir. Her varlık kendi içinde bağımsız bir şekilde yönetilirken, hepsi bir ortak, birleşik bir önyüze bağlı kalırlar.

Servis yönelimli mimarinin temel parçalarından biri de, uygulamadaki detayları soyutlaştıran birleşik bir iletişim noktası katmanı oluşturmaktır. Bunu başarmak için çoğunlukla endüstri ve tasarım standartlarını tek potada eritmek gerekir. Gerçek anlamda karşılıklı çalışabilmek için bu birleşimin tutarlılığı anahtar rol oynar.

Birleşik iletişim noktası katmanı satıcı/sağlayıcılar arasında seçme fırsatının artmasına yardımcı olur. Örneğin bir servisin diğerlerine nazaran tamamen farklı bir platform üzerine inşa edilmesi gerekebilir. Servislerde uyumlu iletişim noktaları olduğu sürece, uygulamaların yönetimi de birbirinde bağımsız olacaktır. Bu sadece servislerin farklı geliştirme ortamları (EJB, .NET, SOAP, REST gibi) üzerinde inşa edilebileceğini vurgulamaz, ayrıca farklı aracı platform ve teknolojinin birbiri ile gerektiğinde uyumlu çalışabileceğini de gösterir.

Bilinmelidir ki, bu tip bir farklılığın bir ücreti vardır. Bu prensip farklılaşmayı savunmaz, basit bir şekilde gerektiğinde farklılaştırmamızı da önerir. Böylece son teknoloji ve platformlar iş gereksinimlerini gerçeklerken de kullanılabilir.

## **İş ve teknoloji paydaşları ile işbirliği ile servisleri belirleyin**

Teknik çözümlerin iş odaklı olabilmesi için teknolojinin iş ile senkronize olması gerekir. Bu nedenle, servis yönelimli hesaplamanın bir başka hedefi de teknoloji ve iş ortak noktalarda buluşturmadır. Bu buluşmanın gerçekleşmesi, çoğunlukla servis geliştirme ve yayına almadan çok daha önce olan analiz ve modelleme süreçleri sırasında olur.

Servis yönelimli analizi çıkarabilmek için gerekli önemli bir etken de, iş ve teknoloji uzmanlarının olası aday servisleri beraberinde ele tanımlamaya çalışmasıdır.

Örneğin, is uzmanları işlevleri doğru bir şekilde çıkarmaya yardımcı olurken, teknoloji uzmanları kavramsal servislerin tanımlarından ve içindeki tanecikli yapıdan emin olabilmek için pragmatik girdiler sağlayabilirler.

### **Mevcuttaki ve gelecekteki kullanım şeklini de göz önünde bulundurarak servis kullanımını en üst seviyeye çıkarın**

Verilen bir SOA projesinin kapsamı kurum çapında yada kurumun bir alanıyla sınırlı olabilir. Kapsam ne olursa olsun, bir servis envanteri çıkarabilmek için ön-tanımlı sınırlar belirlenir. Bu servisler geliştirilmeden önce kavramsal olarak modellenmelidir. Birbiri ile alakalı birden fazla servisi modellerken, sonunda geliştirdiğimiz servislerin bir planını çıkarırız. Bu modelleme egzersizi farklı çözümler arasında paylaşılan servisleri de belirler ve tanımlarken kritik bir öneme sahiptir.

Servis yönelimli analiz aşamasını gerçekleştirmek için birçok farklı yöntem ve yaklaşım mevcuttur. Hepsini için ortak yöntemlerden biri servislerin işlevlerinin gereksizlerden ayıklanması için normalize edilmesidir. Normalize servislerin yeniden kullanılabilir servisler olmasına da gerek yoktur. İşin içine servislerin tanecikli yapısı, özerkliği, durum yönetimi, ölçeklenebilirliği, birlesikliği ve yeniden kullanılabilirliği girer.

İs ve teknoloji ihtisası tarafından yönlendirilen bu tip düşünceler, gelecek değişikliklere uyum için gereken esnekliğe sahip olurken mevcut kullanım gereksinimlerini tespit ederek servisleri tanımlamamıza fırsat verir.

### **Servislerin iş gereksinimlerini ve hedeflerini yerine getirdiğini teyit edin**

Herşeyde olduğu gibi servisler de yanlış kullanılabilir. Bir servis portföyü geliştirir ve yönetirken, iş gereksinimlerini gerçekleştirmedeki verimliliği ve kullanım şekli ölçülmeli ve doğrulanmalıdır. Bir çok araç, servisin nasıl kullanıldığının gözlemlenmesi ve ölçülmesi için ortam sunar. Ancak bu göz ardı edilmemelidir ki, servisler sadece varolduğu için kullanılmaz, iş ihtiyaçlarını ve beklentileri karşılayıp karşılamadığının doğrulanması için de kullanılır.

Bu birden fazla bağımlılığı bulunan ortak servisler için de özellikle doğrudur. Ortak servislerin, onları kullanan tüm çözümlerde ölçeklenebilirlik ve güvenilebilirliği garanti altına alabilmek için uygun bir altyapıya sahip olmaları yetmez, ayrıca işlevlerin bozulmaması için büyük bir dikkat ile tasarlanması ve genişletilmesi de gereklidir.

### **Gerçekteki kullanımına bakarak servisleri ve servislerin düzenini evrimleştirin**

Bu rehber prensip, iş ve teknolojinin ortak bir noktada buluşabilmesini hedefleyen bir amaç olduğu kadar, doğrudan “Baslangıçtaki mükemmellik arayışı yerine evrimsel iyileştirmeye” değer ifadesi ile ilişkilidir.

Servislerin tanecikli yapısını, işlevlerini ya da nasıl bir kompozisyonda olacağını belirlerken varsayımlara asla itimat etmemeliyiz. İlk başta hazırladığımız analiz neye dayanırsa dayansın, bir servis bir işlevi belirler ve bir ya da birden fazla servis bileşimini ilgilendiren bir ya da birden fazla işlevsel özellik içerir.

Gerçek hayattaki is gereksinimleri ve ortamları degistikçe, servislerin arttirilmesi, genişletilmesi, yeniden yapılandırılması ve hatta degistirilmesi gerekebilir. Servis yönelimi tasarım prensipleri, servis mimarisine dogal bir esneklik katar. Böylece gerçek dünya kullanım sekline bir cevap olarak degisebilmek için, servisler de kendi kendine toparlanırlar ve degisim ile uyumlu olurlar.

### **Farkli oranlarda farkli yönleri degisen sistemleri ayirin**

Yekpare ve silo tabanlı sistemleri böylesine sert ve kırılğan yapan şey, mevcut kullanım üzerinde önemli etkisi olan degisimin kendisidir. Bu nedenle çoğunlukla, yeni silo tabanlı uygulamalar yaratmak, mevcut olanları arttirip genişletmekten daha kolaydır.

İslerin ayrılığı (“separation of concerns”, genel olarak bilinen bir yazılım mühendisliği teorisi) altında yatan mantıklı açıklama sudur. Büyük bir problem daha ufak is ve problemlere parçalandığında daha verimli çözümler. İslerin ayrılığına servis yönelimi uyguladığınız zaman, her bir isı çözebilmek için çözüm mantığını içeren ilgili birimleri yaratmış oluruz. Dolayısıyla bu yöntem, farklı ayarlar ile farklı problemleri çözebildiği gibi, daha büyük problemleri çözmek için birimleri bir araya getirmeye izin vermiş de olur.

Servis yeniden-kullanılabilirliğini teşvik ederken, bu yöntem birçok soyutlaştırma katmanı oluşturarak, servis sistemlerini degisimin etkilerinden korur. Bu tür bir soyutlaştırma farklı seviyelerde bulunabilir. Örneğin, eğer eski kaynakları içeren bir servisin degistirilmesi gerekirse, servisin orijinal iletişim noktaları ve işlevsel davranışlarını koruduğu ölçüde degisimin etkileri de azaltılmış olur.

Bir başka örnek ise bağımlıdan bağımlı olmayan mantığın ayrışmasıdır. Eğer çok amaçlı ise ve degisim ihtimali az ise, bağımlı mantığın yeniden kullanılma potansiyeli fazladır. Diğer taraftan, bağımlı olmayan mantık ise tek amaçlı iş süreç mantığını temsil eder, ki bu çoğu zaman geçicidir. Farklı servis katmanlarında bu mantık tiplerini ayırtmak yeni soyutlaştırma katmanlarının tanımlanmasına neden olur. Bu ise servis ve onları kullanan çözümleri degisimin etkilerinden korurken, servis yeniden kullanılabilirliğini arttirir.

### **Gizli bağımlılıkları azaltın ve tüm dış bağımlılıkları dayanıklılığı arttirmek ve degisimin etkilerini azaltmak için yayınlayın**

Bilinen en tanınmış servis yönelim tasarım prensibi, gevsek bağlaşımdır (loose coupling). Servislerin içeride nasıl yapılandırıldığı ve onları kullanan programlar ile nasıl bir ilişkide olduğu, hepsi servis mimarisinin parçaları üzerindeki bağımlılıklar ile ilgilidir.

Soyutlaştırma katmanları, degisimin etkilerini kontrollü bölgelere yerleştirerek evrimsel degisimi kolaylaştırır. Örneğin, geliştirme bağımlılıklarını en aza indirmek için, geliştirmeyi daha soyut hale getirdiği için mimari içinde servis facade’leri kullanılabilir.

Diğer taraftan, yayınlanan teknik sözleşmeleri, servisler ile etkileşimde bulunabilmek için müşterilere bağımlılıkları bildirmelidir. Degisim oluştugunda teknik sözleşmeleri etkileyecek iç bağımlılıklardaki azaltarak, müşteriler üzerindeki degisimin etkilerini çoğaltmaktan kaçınmış oluruz.

## **Tüm soyutlama seviyelerinde, her servisi birbirine bağlı ve yönetilebilir islev birimleri çevresinde düzenleyin**

Her servis iyi tanımlı bir isleve ihtiyaç duyar. Bu islev servisin islev sınırları içinde ne gibi bir mantığın olması ya da olmaması gerektiğine karar verir. Kapsamın belirlenmesi ve islev sınırlarının tanecikliği servis geliştirme ve yayınlanma süreçleri içindeki en kritik sorumluluklardan biridir.

Büyük islevsel sorumlulukları olan servisler, verimlilik açısından daha sert ve kırılabilirler, özellikle de yeniden kullanılması beklendiğinde. Diğer taraftan, asiri küçük sorumluluklara sahip servisler, sayıları artan bileşenleri içerebilmek için altyapı külfetine neden olabilirler.

İslev kapsamı ve taneciklik arasındaki doğru dengeyi bulabilmek, is ve teknoloji ihtisasının bileşimine ihtiyaç duyar. Ayrıca, servislerin verilen sınırlar içinde diğerleri ile nasıl ilişkide olduğunun anlaşılmasını gerektirir.

Bu manifestoda bahsedilen birçok rehber prensip, her servisi bilisim kurumlarını hedef duruma itme yeteneğine sahip bir bilisim varlığı olarak tanımlayarak ve servis yönelimli hesaplanmanın stratejik faydalarının farkına vararak, ilgili kararları verebilmemize yardımcı olur.

Son olarak, sunu belirtebiliriz. Kavramların belirlenmesinden yayın aşamasına, oradan tekrarlanan kullanım şekillerine, bu, servis yönelimli islev birimlerinin evrimsel yolunu belirleyen gerçek dünya is değerlerinin bir başarısı olacaktır.

Tesekkür: Ben bu sayfadaki içeriği 21-22 Kasım 2009 tarihlerindeki haftasonunda, o zaman için halen üzerinde çalıştığım “Yeni Nesil SOA” kitabı için yazdım. Orijinal SOA Manifestosuna ilave olarak bu sitede yayınlamama izin veren Prentice Hall’a tesekkür ederim. Ayrıca, ilettikleri geribildirimler ve açıklamalardaki yardımları için SOAPatterns.org topluluğuna ve çalışma grubu üyelerine de tesekkür ederim. Birçok çalışma grubu üyesi kendi yazılarını, bloglarını ve makalelerini yayınladı. İleri yorum ve düşünceleri görmek için onları incelemenizi tavsiye ederim.

*Tesekkür: Ben bu sayfadaki içeriği 21-22 Kasım 2009 tarihlerindeki haftasonunda, o zaman için halen üzerinde çalıştığım “Yeni Nesil SOA” kitabı için yazdım. Orijinal SOA Manifestosuna ilave olarak bu sitede yayınlamama izin veren Prentice Hall’a tesekkür ederim. Ayrıca, ilettikleri geribildirimler ve açıklamalardaki yardımları için SOAPatterns.org topluluğuna ve çalışma grubu üyelerine de tesekkür ederim. Birçok çalışma grubu üyesi kendi yazılarını, bloglarını ve makalelerini yayınladı. İleri yorum ve düşünceleri görmek için onları incelemenizi tavsiye ederim.*

- Thomas Erl