

Yazılım Takımlarında Başarı

Tunca SELBES

Meteksan Sistem, Simülasyon ve Görsel Sistemler

Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği, Ankara, Türkiye

e-posta: tunca.selbes@sgs.meteksan.com.tr e-posta: tunca.selbes@isbank.net.tr

Özet

Günümüz yazılım takımlarının büyük bir çoğunluğu çok iyi yazılım geliştiricilerden oluşmalarına rağmen, projeler başarısızlık ile sonuçlanmaktadır. Yazılım takımları gerekli özen gösterilmeden kurulmakta ve projelere atanmaktadır. Takım performansları unutulmakta, kişisel performanslara önem verilmektedir. Doğru olan ise takım performansının önemli olmasıdır. Yazılım geliştiriminde motivasyon sanıldığından çok daha büyük önem taşımaktadır. Bu makale, yazılım takımı kurarken göz önünde bulundurulması gereken noktaları, kişisel ve takım performanslarının önemini, motivasyonun gerekliliğini, takım içindeki roller ve takım psikolojisi hakkında yapılan literatür araştırmasını, kişisel deneyim ve görüşleri içermektedir.

Anahtar Kelimeler

Kişisel Performans, Takım Performansı, Motivasyon, Roller, Koordinasyon, Takım Psikolojisi

1 . Giriş

Dünyada küçük, orta ve büyük ölçekli birçok yazılım şirketinde, yazılım geliştiricilerin performanslarını artırmak en önemli ve en zor konulardan biridir. Bu nedenle takımları kuran kişilere çok büyük görev düşmektedir. Takımları, birbirleri ile kolaylıkla anlaşabilecek doğru kişilerden kurmak gerekir. Doğru roller, doğru kişiler ve takım ruhu ekipte yer aldığı anda başarı kaçınılmazdır. Kişisel performanstansa takım performansının daha önemli olduğu unutulmamalıdır.

Yazılımda başarıyı sağlayan etkenler arasında kişisel performans, takım performansı, sosyal aktivite, doğru roller ve kullanılan araçlar gösterilebilir. Takım başarısı ise müşteri memnuniyeti, maliyet etkinlik, zamanında ve üretkenlik olarak ölçülebilir.

2 . Kişisel Performans Mı, Takım Performansı Mı?

Kişisel yapılan yazılım projelerinde veya küçük yazılım projelerinde, kişisel performans gerçekten büyük önem taşır. Fakat proje boyutu büyüdükçe ve buna bağlı olarak projede çalışan insan sayısı çoğaldıkça takım performansının önemi ortaya çıkar. Kimse takım çalışması olmadan sadece kişisel performansı ile büyük projeleri başarısızlıktan kurtaramaz. Tabii ki kişisel performans yüksek olmadan takım performansı yüksek olamaz. Ama bu bir kişinin yüksek performansının bütün takım performansını artıracığı anlamına gelmez. Esas önemli olan her bir bireyin kişisel olarak maksimum performansa ulaşip, takım içerisinde bir sinerji yaratmasıdır.

“Neden yetenekli ve özel programcıların oluşturduğu takım, gerçek bir takım olamayabilir?”. Bunun nedeni bazı programcıların birlikte çalışamamaları veya uyum gösterememeleridir. Bu yeteneklerinden veya özel olmalarından kaynaklanmaz; sorun kişisel konulardandır [1]. Takım içindeki insanların iletişimi güçlü olmadan, gerçek bir uyum sağlanamaz.

3 . Motivasyonun Performansa Etkisi

Her meslekte olduğu gibi yazılım sektöründe de performans, motivasyondan direk etkilenmektedir. Çalışanların motivasyonu ne kadar yüksek ise performansları da o kadar yüksek olur. Kişinin motivasyonu; psikolojik faktörlere, yeteneklerine ve iletişim kabiliyetine bağlıdır. Bu noktada yöneticilere büyük görev düşmektedir.

Çalışanların motivasyonunu yükseltmek için yöneticiler, çalışanlarının psikolojik yönlerini keşfetmeli, sorunlarına yardım etmeye çalışmalı, maddi ve manevi yönden onların yanlarında olduklarını hissettirmelidirler. Yöneticiler çalışanların yeteneklerini geliştirmek için onları eğitime göndermeli ve her yönden destek olduklarını göstermelidirler. Bu motivasyon yükseltici faaliyetler gerçekleştirilirken mutlaka olumlu bir iletişim kurulmalıdır.

Bu noktada yöneticiler şirket politikalarını belirlerken yüksek performans bekleyip çalışanlarını motive mi etmelidirler, yoksa önce çalışanlarını motive edip daha sonra yüksek performans mı beklemeliler? Bence bu konuda şirketler her iki yöntemi aynı anda kullanmalı ve çalışanlarına uyguladığı politika şu şekilde olmalıdır: Önce çalışanını motive etmeli, daha sonra bu motivasyondan yüksek performans beklemeli, aldığı yüksek performansa karşı da çalışanını tekrar motive edecek şekilde ödüllendirmelidir. Bu döngü içinde takımın ve şirketlerin daha başarılı olabileceğini düşünüyorum.

4 . Takım Yaratmak

Takımlar projelere göre yaratılır. Projenin içeriği ne kadar büyük ise takım da o kadar büyük olur. Projeler iki bileşenden var olur diyebiliriz. Birincisi teknik bileşenler; süreçler, prosedürler, kullanılan araçlar, metodolojiler, kullanılan diller gibi. İkincisi ise organik bileşenler; insanlar, motivasyon, düşünceler gibi. Projelerde çoğu zaman teknik bileşenler daha çok önem arz etmektedir. Projenin başından sonuna teknik konularda birçok toplantı birçok değişiklik yaşanır.

Organik bileşenler ise gerekli mevkilere insanlar atandıktan sonra unutulmaktadırlar. Gerekli önem sağlanmamaktadır.

Yazılım takımı oluşturulurken üstesinden gelinmesi gereken bazı noktalar unutulmamalıdır. Bunlar;

- İnsanlar farklı farklıdır ve yazılım geliştiriciler farklı insan tipleri ile çalışmaya alışkın olmayabilirler
- İnsan ilişkilerinde zayıf olan yazılım geliştiriciler olabilir. Hâlbuki bu tarz yazılım geliştiriciler teknoloji konusunda gerekli bilgi birikimine sahip olabilirler
- Düşüncelerde farklılıklar olabilir. Bu da sistem kavramada veya çözümünde farklılıklara yol açabilir
- Takım çalışmasına yatkın olmayan çalışanlar olabilir.

Takımları sağlam şekilde kurmak için aşağıdaki noktalar dikkate alınmalıdır. Bunlar;

- Ortak bir dil oluşturulmalı. Müşteri ilişkileri, takım iletişimi, karar verme yeteneği gibi
- Ortak bir hedef oluşturulmalı
- Başarılar edinilmeli ve paylaşılmalı
- Takım üyeleri arasında bilgi alışverişi olmalı. Takımdaki herkes herkesin ne yaptığını bilmeli, nasıl öğrendiklerini paylaşmalı ve nasıl başarılı olduğunu anlatmalı
- Mutlaka bireysel ve takım olarak kazançlar elde edilmeli ve paylaşılmalı

Takım içi koordinasyon da önemli noktalardan biridir. İnsanlar, görevler ve kaynaklar arasında koordinasyonu sağlamak gerçekten zor bir işlemdir. Büyük ölçekli yazılım projelerinde birden çok takım bulunabilir. Aynı ürün üstünde paralel olarak çalışan birçok kişi olabilir. Günümüzün gelişmiş yazılım mühendisliği araçlarına rağmen, koordinasyon insan ve kişisel faktörlerden dolayı problem olmaya devam ediyor [3].

Yazılım takımlarını kurarken veya rollerini analiz ederken R. Meredith Belbin tarafından tanımlanan Belbin Takım Rollerini yöntemi kullanılabilir. Belbin rolleri kullanılarak takım performansı geliştirilebilir, başarılı takımlar kurulabilir [1,2].

5 . Yazılım Takımı ve İçinde Bulundurduğu Roller

Bir yazılım projesinde başarılı olmak için en önemli nokta takım olmaktır. Olması gereken yazılım şirketini anlatırken bir futbol kulübünü örnek göstererek anlatmaya çalışacağım.

En tepeden başlarsak nasıl futbol kulüplerinin başkanları var ise yazılım şirketlerinin de yöneticileri vardır. Futbol kulübü başkanları teknik konularda bilgili olsalar da hiçbir zaman takıma teknik anlamda karışmazlar. Yazılım şirketlerinde de yönetici olan kişiler teknik açıdan yeterli olmalı fakat proje akışına çok fazla müdahalede bulunmamalıdır. Kulüp başkanı oyuncu alımında teknik direktör ile birlikte seçilen oyunculardan kulübün parasının ve oyuncunun yeteneklerinin yeterli olduğuna inandıkları oyuncuyu alırlar. Aynı şekilde yazılım şirketlerinde de şirket yöneticisi ve

proje yöneticisi çalışan yeteneklerini ve maddi durumları değerlendirdikten sonra çalışanları işe alırlar.

Proje yöneticileri aynı birer teknik direktör gibidirler. Teknik direktörler uygulanacak taktikleri verirler, antrenmanları düzenlerler, maçlarda oynayacak oyuncuları seçerler, oyuncu transferlerini yönlendirirler ve daha birçok görevleri vardır. Proje yöneticilerinde de durum aynıdır. Projede çalışacak kişileri seçer, kimin hangi görevi yapacağına karar verir, motive eder, olaylara karşı taktik değişikliklerine karar verir. Unutulmamalıdır ki proje yöneticisi çok iyi bir yönetici olabilir, fakat her işi tek başına yapamaz. Bu noktada Yazılım Kalite Güvence Uzmanları, Yazılım Test Mühendisleri ve Konfigürasyon Yöneticilerinin rolü ortaya çıkmaktadır.

Yazılım projelerinde en önemli görevlerden birkaçını Yazılım Kalite Güvence Uzmanları, Yazılım Test Mühendisleri ve Konfigürasyon Yöneticileri sahiplenir. Bu görevleri futbol kulüpleri ile örneklemek gerekirse, bu görevdeki kişiler birer antrenör gibidirler. Bir futbol takımında değişik pozisyondaki oyuncular için değişik antrenörler bulunur. Takımdaki oyuncuların eksik yönlerini bulup, teknik direktör bilgisinde o eksiklikleri gidermeye çalışırlar. Yazılım projelerinde de durum aynıdır. Antrenörler gibi, Yazılım Kalite Güvence Uzmanları, Yazılım Test Mühendisleri ve Konfigürasyon Yöneticileri yazılım projesinin daha iyi olması için o projenin eksik yönlerini saptarlar, düzenlerler, kontrol ederler ve takibini sağlarlar. Bu örnekleme ışığında, antrenörler futbol takımı için ne kadar önemli ise Yazılım Kalite Güvence Uzmanları, Yazılım Test Mühendisleri ve Konfigürasyon Yöneticileri de yazılım projeleri için o kadar önemlidir. Projenin daha iyi olması için yazılım mühendislerine birçok konuda katkıda bulunurlar.

Yazılım projesinde çalışan programcılar ise aynı birer futbolcu gibidirler. Asıl işi yapan onlardır. Fakat gerekli teknik desteği Proje Yöneticisinden, Yazılım Kalite Güvence Uzmanlarından, Yazılım Test Mühendislerinden ve Konfigürasyon Yöneticilerinden almadan kesinlikle başarıya ulaşamazlar. Bireysel olarak ne kadar iyi programcılara sahip olunursa olunsun takım ruhu olmadan başarıya ulaşamayacağı yazılım takımlarına veya futbol takımlarına bakılarak görülebilir.

6 . Yazılım Takım Psikolojisi

Yazılım projelerinde programcılar genelde Yazılım Kalite Güvence Uzmanlarına, Yazılım Test Mühendislerine ve Konfigürasyon Yöneticilerine ekstra iş yaratıyorlarmış gözü ile bakarlar. Hâlbuki istedikleri ve verdikleri işler o takımın ürettiği yazılımın daha kaliteli olması içindir. Dolayısıyla programcılar ve Yazılım Kalite Güvence Uzmanları, Yazılım Test Mühendisleri ve Konfigürasyon Yöneticileri arasında iletişim çok büyük önem taşımaktadır. Bu noktada iletişimi sağlayacak, takım ruhunu her zaman dinamik tutacak bir göreve ihtiyaç duyulmaktadır. Futbol takımı örneğini vermek gerekirse, futbol takımlarında futbol şube sorumlusu görevi vardır. Bu görevdeki kişiler futbol takımına her türlü motivasyonu sağlarlar, manevi sorunları çözmeye çalışırlar, yöneticilerle iletişimi sağlarlar. Motivasyon ve iletişimin çok önemli olduğu yazılım projelerinde de bu tarz bir görevde çalışacak deneyimli, iletişim yeteneği yüksek, ve motivasyon gücü olan bir çalışan olması yazılım projesi için yararlı olabilir.

7 . Sonuç

Sosyal bir sistemde bazı roller, o sistemin ayakta kalması için kaçınılmazdır. Yazılım takımlarında da aynı durum söz konusudur. Bazı olması gereken roller takımın başarısı için gereklidir. Doğru roller doğru kişilere verildiği takdirde yazılım takımında başarı kaçınılmazdır. Büyük projelerin zamanında bitmemesi ve başarılı sonuçlanmamasının nedenlerinin başında motivasyon eksikliği, takım olamama ve doğru rollerin atanmaması örnek gösterilebilir. Bu noktada çözüm yöneticilere kalmaktadır. Yönetici motivasyon ve performans değerlendirmesini mutlaka doğru şekilde yapmalı ve uygulamalıdır. Ayrıca takım ile olan ilişkisini iyi kurmalı ve sürekli iletişim halinde olmalıdır. Takımı doğru kişilerden sağlam temeller üstüne kurmalıdır. Takım üyeleri de takım içerisinde aynı amaç için çalıştıklarını unutmamalı ve yöneticilerine yardımcı olmalıdırlar.

8 . Kaynakça

- [1] K. Todd Stevens, “The Effects of Roles and Personality Characteristics on Software Development Team Effectiveness”, Mart 1998, ulaşılabilir adres <http://scholar.lib.vt.edu/theses/available/etd-31098-11056/unrestricted/FINAL5.PDF>
- [2] K. Todd Stevens, Sallie M. Henry “Analyzing Software Teams Using Belbin’s Innovative Plant Role”, ulaşılabilir adres <http://www.radford.edu/~kstevens2/ISTall.pdf>
- [3] J. Alberto Espinosa, Robert E. Kraut, Sandra A. Slaughter, Javier F. Lerch, James D. Herbsleb, Audris Mockus, “Shared Mental Models, Familiarity and Coordination: A Multi-Method Study of Distributed Software Teams”, Aralık 2002, ulaşılabilir adres <http://www-2.cs.cmu.edu/~kraut/RKraut.site.files/articles/Espinosa02SharedMMModelsFamiliarityCoordination.pdf>