

Katedral ve Pazar

Bilgisayar Mühendisi **Burak OĞUZ**
burakoguzs@yahoo.com

Eric S. Raymond, kitabında kendi açık kaynak projesinde, Linux ile ortaya çıkan yazılım mühendisliği kuramlarının sınanması ile ilgili bir çalışmayı sunuyor ve bunu yaparken iki temel modeli karşılaştırıyor. İlki ticari model olan ve kendisinin "Katedral" dediği kısım ve onun karşısında Linux dünyasının modeli olan "Pazar".

Yazar, Linux yıkıcıdır diyerek başlıyor kitabına. Zaten kim bilebilirdi ki dünyanın dört bir yanında bulunan binlerce yazılımcının Internet'in ince telleriyle birleşsin ve dünya çapında bir işletim sistemi oluştursun? Bunu tahmin etmek elbette çok zordu ama gerçekleştirdiği değişime ve bakış açısına sahip çıkabilmekte bu akımın kazanılması gereken değerlerindedir. Yazar için bu, yazılımda karmaşanın kontrolüne karşı olan değerlerinin değişmesiyle gerçekleşiyor. Linus Torvalds'ın tarzı yani erken ve sık aralıklarla sürüm çıkarmak, mümkün oldukça çok şeyi

sunabilmek ve özellikle karışıklıklara açık olabilmek o dönemler için epey şaşırtıcıydı ve Linux çevresinde yapılanların tutarlı ve kararlı sistemler doğurması Katedral tarzı ihtişam yerine bir Pazar yeri havası oluşturmuştur. Bu karmaşanın içerisinde doğan başarı yazarı cazip eden ve bunun sebeplerini araştırmaya iten en büyük etken olmuştur.

Eric S. Raymond, kendi Pazar projesini oluşturmaya başlamasıyla bunun içerisinde birçok kuralın hüküm sürdüğünü görmeye başlar ve bu kuralları şöyle sıralar.

1. Her iyi yazılım, programcısının kendi yarasını kaşınmasıyla başlar. Aslında bu iyi bilinen bir kural olmakla birlikte yazılımcılar genelde hayatlarını ihtiyaç duymadıkları ve sevmedikleri yazılımları geliştirerek geçirirler. Ancak Linux dünyasında üretilen yazılımların kaliteli olması da bu nedenledir.

2. İyi yazılımcı, ne yazacağını bilen yazılımcıdır. Ama neyi yeniden yazacağını bilen yazılımcı "büyük yazılımcıdır". Örneğin Linus Torvalds da Linux'a başlarken de sıfırdan başlamamış ve Unix benzeri ufak bir işletim sistemi olan Minix kodlarını kullanmıştır. Unix'ten miras alınan kodun yeniden kullanımı felsefesi Linux'ta sınırları aşmış ve insanların benzer işlerdeki çalışmalarına ulaşmakta sarf edilen çaba önemli ölçüde azalmıştır.

3. "Heba etmek hesap dâhilindeyse, bir şekilde heba edersin". Bunu, bir çözüm bulunana kadar, sorunun ne olduğu anlaşılma şekline de ifade edebiliriz. Yani işi hakkıyla yapmayı öğrenmek ikinci denemede ya da üçüncü denemede mümkün olabilecektir.

4. Doğru yoldaysan, birbirinden ilginç problemlerle

karşılaşacaksınız. Belli bir süreden sonra yapılan iş ufak tefek yamalar yapmaktan çıkar. Daha önemli değişiklikler yapılması gerektiğinin farkına varılır ve bir takım kuşku bulutları sekillenmeye başlar.

Pazar projelerinin diğer önemli bir kuvveti de kullanıcılarıdır. Linux, Unix'ten aldığı bu kuvvetini uç noktaya taşımıştır çünkü kullanıcılarının çoğu aynı zamanda da yazılımcıdır. Kaynak kodu el altında olduğu için etkili yazılımcılar olurlar ve hata bulma ve ayıklama sürelerini oldukça kısaltabilirler. Bu durum bizi bir diğer kurala taşır.

5. Programın oluşumunda kullanıcıların katkısına başvurmak, kod geliştirme ve hata ayıklamada en etkili ve en rahat yoldur. Aslında bu etkinin karmaşıklık ve kullanıcı sayısı ile artacağı, Linus Torvalds aksini ispat edene kadar, açık kaynak dünyası tarafından da göz ardı edilmiştir. İşte bu yüzden ki Torvalds'ın inşa ettiği en iyi şey Linux çekirdeği değil Linux geliştirme yöntemidir.

Linux'un yıktığı bir diğer kural da Katedral tarzı yazılım geliştirme sürecidir. Linux özellikle ilk çıktığı dönemde piyasaya erken ve sık sürüm çıkarmıştır. Elbette bu yazılım geliştirme sürecinin dinamiklerine aykırıydı. Çünkü ilk sürümler hep hatalı ve eksik olurdu. Ancak bir yıl kadar sonra, Linux gözle

görülür olmaya başladığında farklı ve sağlıklı olduğu görülmeye başlandı. Bu Torvalds'ın kullanıcılarını geliştirme ortağı olarak görmesine bağlanıyordu.

6. Piyasaya erken ve sıklıkla sür. Müşterilerinin sesine kulak ver.

Hızlı sürümler ve Internet ortamını ayaklandırmak, Torvalds'ın en dolaysız yolu bulduran kurgulama dehasını tamamlayıcı olarak öne çıkartmaktadır. Torvalds'ın dehasıyla maksimize ettiği şey aslında arıza giderme ve geliştirme süreci için harcanan adam-saat miktarydı. Yazar buna "Linus Kanunu" imini koyuyor ve şöyle özetliyor : Ne kadar gözden geçirirsen; hatalar o kadar azalır.

7. Geliştirme ortağı ve beta sınavıcı havuzu yeterince büyükse sorunlar çabucak belirlenir ve halledilir.

Karışıklığı gidermek her zaman zor olanıdır ve bunun için kullanıcıya en büyük rol düşer. Kaynaktan uzak kullanıcıların geri beslemelerinin çoğu zaman anlamsız ve yanlış yönlendirici olması Katedrallerin en büyük sorunudur. Linux'un geliştirilmesinde karışıklığın bu kadar kolay çözülebilir hale gelmesi en çok kullanıcıların doğrudan koda müdahale edebilmesinin bir sonucudur. O zaman açık kaynak projeleri için şunu diyebiliriz.

8. Beta sınavıcılarınıza en değerli

kaynaklarınız olarak muamele ederseniz, en değerli karşılığı verecektirler.

9. İyi fikirlerle ulaşmada bir sonraki en iyi adım, kullanıcılarından gelen iyi fikirleri elde etmektir. Bazı durumlarda ise bir sonraki daha iyidir.

Ayrıca yazar genel olarak böyle hızlı yazılım geliştirme süreçlerinde teknik olarak incelendiğinde çeşitli örneklerle dayanarak şu tavsiyelerde bulunuyor.

10. Genellikle en çarpıcı ve yenilikçi çözümlere, problem konseptinizin yanlış olduğunu fark etmekle kavuşursunuz.

11. "Mükemmele, eklenecek bir şey kalmadığında değil, eksiltilecek bir şey kalmadığı zaman ulaşılır."

12. Her bileşenin işlevi doğrultusunda iş görmesi gerekir; ama müthiş bir bileşen, işlevlerinin ötesinde iş görür.

13. Bir güvenlik sistemi ancak sırrı kadar güvendedir.

Peki Pazar tarzı geliştirmenin ön koşulları nelerdir? Elbette Pazar tarzı içinde kod geliştirme yani üretim yapmak zordur. Zaten Torvalds da o zaman için denememiştir. Büyüyen geliştirme topluluğunun oynayacağı, işleyen sınavabilir bir şeye ihtiyacı vardır. Bununla birlikte topluluğu kurabilmeniz ve

yürütebilmeniz için makul bir vaade ihtiyaç vardır. Ayrıca Pazar tarzı bir sürecin içerisinde koordinatörün ihtiyaç duyacağı müstesna bir tasarım yeteneği değil, asıl olarak başka fikirlerin parlaklığını teşhis edebilme yeteneğidir.

Pazar yazılımları toplumsal olarak incelendiği zaman ise hareket noktası alabileceğimiz iki kaynak vardır. Bunlardan ilki The Mythical Man-Month adlı kitabında Fred Brooks, projelerde iletişim maliyetinin ve karmaşıklığın geliştirici sayısının karesiyle arttığını öne sürmüştü. Ancak bu kabul gören fikri Linux tamamen geçersiz kılmıştır. Weinberg de The Psychology of Computer Programming adlı klasiğinde "egosuz programlama" tartışmasında, yazılımcıların kodları konusunda mülkiyetçi olmaması, aksine hata ayıklama konusunda insanları teşvik etmesi gerektiğini öne sürmektedir.

Ucuz Internet, Linux'un evriminde gerekli bir önkoşul olsa da diğer kritik faktör de liderlik tarzının ve geliştiricilerin ortak geliştiricileri cezbederek ortamdaki maksimum desteği almayı sağlayan işbirliği alışkanlıklarının gelişmesi olmuştur. İşte Linux projesi de "birbirine yakın niyetlerin ciddi çabası"dır. Linus Torvalds, kendini esas olarak başkalarının yapıldığı bir projenin gözetmeni olarak başarıyla konumlandırmakla ve projeye ilgiyi sürdürülebilir hale gelene kadar beslemekle, "ortak kavrayış temeli"ni derinlemesine oluşturmayı

başarmıştır. Bu da Pazar modeli için gerekli koordinatörün tanımını oluşturmuştur. Brooks fikirlerine ters olarak:

14. En az Internet seviyesinde bir iletişim ortamıyla donatılmış bir geliştirme koordinatörü, dayatmacı olmadan önderlik yapabilirse; çok sesliliğin tek seslilikten iyi olacağı kesindir.

Tabii ki Pazar teze- gelenekçi yazılım geliştirme yöneticilerinin itirazları da olmuştur. Bu itirazlar genelde açık kaynak projelerinin oluşum, değişim ve dağılıma durumlarındaki rahatlığın grubun herhangi bir kapalı kaynak karşısında sayısal avantaj yitirmesi odaklıdır. Kapalı kaynak yöneticileri belirli hedefleri olduğu, yapılan işin tüm ayrıntılarıyla takip edildiği, isteklendirme, verimlilik için örgütlenme ve kaynakları daha iyi düzenlemek gibi savları vardır. Ancak kapalı kaynak yöneticilerinin hatalarını görmek için basit örnekler vermek yeterlidir. Açık kaynak projeleri, kapalı kaynak ekiplerinin dar yazılımcı havuzunun tersine, yazılımcıların bilgili oldukları projelerde gönüllü olmalarının getirdiği uzmanlaşmış yoğunlaşmayla birleşen ve Internet yardımıyla tüm dünyayı bütünleştirebilen bir kaynak havuzuna sahiptir. Ayrıca açık kaynak projelerinde örgütlenme sosyal ortamdaki rekabet süreci içerisinde maksimum verimliliği

destekler. Motivasyon ise örgütlenmenin getirdiği sosyal rekabet ortamının doğal sonucudur. Yapılan işin takibi ise açık kaynak dünyada tek bir kişi veya ekibe bağlanmaz. Bunun aksine dünyadaki herkesin denetimindedir ki bu karşılaştırmaya bile gerek olmayacak bir konudur. Son olarak hedefleri belirlemek için kapalı kaynak dünyasındaki yönetim komiteleri ve şirket yol haritalarından açık kaynak dünyasındaki proje liderlerinin daha başarılı olduklarını göstermek gerekir. Bu kolay değildir ancak Torvalds'ın ve benzer dünyayı sarsan projelerin liderleri "dünya egemenliği" açısından geleneksel mekanizmalar üzerinde nasıl bir baskı oluşturduğunu siz düşünün.

Elektronik olarak kitabı aşağıdaki linkten indirebilirsiniz:
[Http://www.emo.org.tr/resimler/ekler/d362d95ed1876eb_ek.pdf](http://www.emo.org.tr/resimler/ekler/d362d95ed1876eb_ek.pdf)

