

Yazılımların Kaynakları (Sources of Software)

Sadi Evren SEKER

American University of Middle East, Kuwait, academic@sadievrenseker.com

Bir işletmede kullanılan yazılımların geliştirme sürelerine, kaynak ayırımına veya yönetsel kararlara göre farklı kaynaklardan gelmesi mümkündür. Bu yazı, işletme gözü ile yazılımların kaynaklarını inceleyecek ve açıklamaya çalışacaktır.

1. Bilgi Teknolojileri Firmaları

Öncelikle bilgi teknolojileri firmaları özellikle yazılım almak yahut kullanmak isteyen kurumlar tarafından genelde yeterli yazılım tecrübesi bulunmadığı zamanlarda tercih edilir. Kurumun yeterli yazılım yazdıracak tecrübesi, elemanı, yönetim becerisi veya bazı durumlarda bütçesi bulunmayabilir. Bu durumda kurum, daha tecrübeli bir yazılım firmasına başvurarak yazılım ihtiyacını karşılamak ister. Aşağıda, 2009 yılındaki yazılım firmalarının dünya çapındaki en yüksek ciroya sahip 10 firması listelenmiştir (kaynak : [Software Magazine])

- IBM \$74,934
- Microsoft \$50,820
- HP \$38,265
- Oracle \$23,252
- Accenture \$21,577
- Computer Sciences Corp. \$16,740
- SAP AG \$15,235
- EMC Corp. \$14,026
- Hitachi \$12,254
- Lockheed Martin Corp. \$12,130

Yukarıdaki listedeki değerler, Amerikan Doları cinsinden olup milyon dolar olarak verilmiştir. Listenin en üstündeki IBM firması, donanım üretimi ile tanınmakla birlikte son on yıl içerisinde faaliyet alanını donanımdan yazılıma kaydırmış olup özellikle orta katman yazılımlar (middleware) konusunda çok ciddi yatırımlar yapmıştır. Liste 2009 yılında hazırlandığı için listedeki diğer önemli orta katman yazılım geliştiricisi olan Oracle henüz alt sıralarda görülmektedir ancak geçtiğimiz 3-4 yıl içerisinde Oracle firmasının da ilgili alanda oldukça yoğun yatırımları bulunmaktadır. Liste yazılım firmalarının genel cirosunu gösterdiği için ne yazık ki konumuz ile ilgili olan kurumsal düzeyde bilgi teknolojisi yatırımını tam yansıtamamaktadır. Örneğin Microsoft firması ikinci sırada olmakla birlikte firmanın birinci gelir kaynağı Office yazılımları ikincisi ise işletim sistemi olan Windows yazılımıdır. Ancak geçtiğimiz 5-10 yıllık dilimde Microsoft firması da bu alandaki yatırımlarını arttırmış olmakla birlikte çeşitli firmaları satın alarak bünyesine katmıştır.

Burada bilgi sistemleri satan firmaların müşterilerine uyarlanmış özel yazılımlar yaptığını unutmamalıyız. Aynı firmaların paket yazılımları da mevcuttur ancak bu yazılımların en büyük farkı firma ayırımı gözetmeksizin genel bir yazılımın firmaya adaptasyonudur. Yapılan araştırmalar göstermiştir ki bir paket yazılım, en fazla %70 oranında bir kurumun ihtiyaçlarını sağlayabilmektedir. Bu oran en iyi ihtimalle yaşanan durumdur ve yazılımın kuruma yeterli uyumu sağlamayarak bir çöp olması ihtimali de küçümsenmeyecek kadar yüksektir.

Bilgi teknolojileri firmaları için en önemli adım, firmanın kurumu tanıması, ihtiyaçlarını belirlemesi ve kurum ile ortaklaşa olarak yazılım geliştirme sürecini tamamlamasıdır. Bu açıdan bilgi teknolojisi firmalarının sunduğu hizmetleri birer anahtar teslim sistem olarak düşünmek mümkündür. Ancak kurumun da firmaya pek çok aşamada insan gücünü ve bilgi kaynaklarını açması ve geliştirme sürecine dahil olması gerekmektedir.

Büyük ölçekli yazılım sağlayıcılarına gelince, bu firmalar tarafından sunulan paketlere büyük ölçekli kaynak planlama sistemleri (enterprise resource planning (ERP) software) ismi verilmektedir. Bu çözüm paket yazılım ile kuruma özel yazılım geliştirme kaynaklarının ortasında bir çözüm olarak görülmeyebilir.

2. Büyük ölçekli çözümler (Enterprise software solutions)

Bu başlığın altında aslında diğer başlıklardaki yaklaşımları kullanmak mümkündür. Yani bir yazılım müşterisinin büyük ölçekli olması, bulut bilişimden veya açık kaynak koddan veya paket yazılımlardan faydalanamayacağı anlamına gelmez. Fakat buradaki ayırımın amacı daha çok entegre olan büyük ölçekli yazılımları ayrı bir başlık altında incelemektir. Örneğin bir firmanın stok yazılımı farklı bir kaynaktan, muhasebe yazılımı farklı bir kaynaktan veya personel yazılımı daha farklı bir kaynaktan olabilir. Bu yazılımların, birbiri ile iletişim içerisinde ve daha da önemlisi entegre olarak çalışması ise büyük ölçekli çözümler altında incelenmektedir. Bu yaklaşımı daha özel olarak ERP (Enterprise Resource Planning, Kurumsal Kaynak Planlama) başlığı altında incelemek de mümkündür. ERP yazılımlarının en önemli farkı, diğer yazılımlarda olan fonksiyonel yaklaşım yerine süreç yaklaşımı getirmesidir. Örneğin bir cari kart hazırlanması, muhasebe yazılımının özelliği olabilir veya bir bordro bastırılması bir personel yazılımının özelliği olabilir ancak ERP yaklaşımında bunlardan daha farklı olarak süreç incelenir. Bu yaklaşım sayesinde ERP sistemlerinin kalitesi (Seker, 2014) ve kullanıcı kazanımları (Seker, 2014) ölçülebilmektedir. Örneğin üretim yapan bir firmadaki ürün sevkiyatı bir süreçtir. Bu süreç sırasında, ürünün satın alınması (SCM, supply chain management), depolanması (stok yönetim yazılımı), üretime dahil edilmesi (operations management, production management, MRP), maliyet hesabı (muhasebe yazılımı), bütün bu aşamalarda en verimli personel istihdamı (personel yazılımı), satış ve satış sonrası destek, pazarlama, gibi çok sayıda yazılımı ilgilendiren aşamalardan geçilir. İşte ERP, sürece odaklanarak gerekli olan yazılım ihtiyaçlarını karşılamayı hedefler.

3. Bulut Bilişim (Cloud Computing)

Bulut bilişim, bu yazı yazılırken henüz yeni bir yaklaşımdı. Çoğu firma bu konu ile ilgili araştırma aşamasındaydı ve hizmet sağlayıcılar henüz ürün/servis olarak piyasaya yeni çıkmaktaydı. Ancak genel olarak yaklaşım hakkında bilgi verilmesi gerekirse şunları söyleyebiliriz. Öncelikle, bulut bilişimin amacı, müşterilere farklı seviyelerde hizmet sağlamaktır. Örneğin en alt seviyede sadece donanım ve internet bağlantısı hizmeti sunulurken en üst seviyede anahtar teslim firmanın bütün ihtiyaçlarını karşılayan entegre yazılımlar sunulabilir. Bulut bilişim kapsamında bu iki seviye arasındaki herhangi bir seviyede hizmet sunulması hedeflenir. Örneğin klasik olarak bilgisayarlara kurulan ofis yazılımlarının yerine bütün işlemlerin Internet üzerinden yapıldığı "google docs" bir bulut bilişim yaklaşımıdır. Buna göre kullanıcı görüntülemek istediği dokümanlarını internet üzerinden hizmet veren ilgili siteye yükler (ya da zaten farklı bir sayfasında hizmet vermekte olan mail uygulamasından bu siteye transfer eder) ve ofis yazılımı sunucu tarafında çalışarak sonucu bir HTML sayfası olarak kullanıcıya gösterir. Bu sayede kullanıcının ilave bir yazılıma ihtiyacı olmadan, kendi bilgisayarındaki CPU ve RAM gibi kaynakları kullanmadan dokümanı görüntülemesi sağlanır. Elbette ofis uygulaması oldukça basit bir uygulamadır ancak konunun anlaşılması için yeterli olacağını düşünüyorum. Kurumsal seviyeye geçildiğinde de aslında ölçeğin büyümesi ve ilave tonla problem çıkması dışında, yaklaşımda çok da büyük bir farklılık yoktur.

4. Açık Kaynaklı Yazılımlar (Open Source Softwares)

Bu yazılımların özelliği, kaynak kodlarının açık olması ve müşteri tarafında kaynak kodlarının görülebilir olmasıdır. Örneğimizi yine ofis yazılımı üzerinden verelim. Bir ofis yazılımı kullanacaksınız ve bu yazılımın bazı kısımlarını özelleştirmek istiyorsunuz. Şirket logosu eklemek, diğer yazılımlarınızla entegre etmek veya tamamen firmamıza özel bazı raporlama işlemleri yapmak için koduna müdahale etmek istiyor olalım. Açık kaynak kodlu yazılımlar burada kodu bize vererek bu ve benzeri geliştirme imkanlarını sağlamaktadır. Ayrıca güvenlik gibi sebeplerden dolayı da bazı firmalar bu yazılımları tercih ederler. Ancak bir piyasa tecrübesi olarak bu yazıyı yazarken belirtmek isterim ki, açık kaynak kodlu yaklaşımın bir zamanlar yılmaz savunucularından olmama karşılık son dönemde bu yaklaşımdan uzaklaşmış bulunuyorum. Bunun en önemli sebeplerinden birisi, açık kaynak kodlu yaklaşımın ilk çıktığı dönemlerdeki gibi iyi niyetli ve özgür yazılım fikrinden daha çok bir pazarlama enstrümanı haline dönüşmesidir. Çoğu yazılım evi, profesyonel olarak pazarlayamayacağı yazılım ürünlerini, ne yazık ki açık kaynak kodlu adı altında pazarlamaktadır. Burada çok farklı hilelerin kullanıldığını üzülerek görüyorum. Örneğin yazılımın en son sürümünü ticari tutup eski sürümünü açık kaynak bırakmak, yazılımı ücretsiz verip dokümantasyon satmak veya hizmet satmak gibi. Elbette bu insanlar da bir yerlerden para kazanacaklar, ancak bu para kazanma yolu, firmaya ücretsiz bir yazılımı verip, ardından firmayı bağımlı hale getirdikten sonra, ücretli yazılımlardan çok daha yüksek maliyetler talep etmek olmamalı. Bu konuda yazılım kullanıcılarının dikkatli davranmasını ve yazılım alımı sırasında en az 5-10 yıllık bütçe planları çıkarmasını tavsiye ederim. Bu ufak serzenişten sonra konumuza devam edelim.

5. Kurum İçi Yazılım Geliştirme (In-House Development)

İşte en önemli ve en dikkatle ele alınması gereken başlık budur. Şimdiye kadar anlatılan kaynakları ne kadar verimli kullanırsa kullansın, firma ölçeği belirli bir büyüklüğe ulaştıktan sonra, mutlaka içeride yazılım geliştirilmesi veya en azından mevcut yazılımlara ek yapılması ihtiyacı doğmaktadır. Hele çok sayıda farklı kaynaktan yazılım kullanan bir kurumdan bahsediliyorsa veya çeşitli rekabet avantajları yüzünden firma yazılımlarını kendisi yazmaya karar veriyse, bir firmanın en çok kaynak ve enerji ayıracağı yazılım maddesi bu başlıktır. Bu başlık aslında kitaplar dolusu yazıların yazılabileceği ve dünya üzerinde en çok denenmiş başlıklardan birisidir ve ne yazık ki bu yazı kapsamında kısa tutmam gerekiyor. Ancak çok kısaca şunları söyleyebilirim. Bir yazılımı, bir firma, kendisi geliştirmek istiyorsa, mutlaka bu konudaki uzman kişilere ve hatta firmalara danışmalı ve detaylı bir analiz sürecinden geçmelidir. Bir yazılımı satın almak veya yazdırmak için harcanması gereken analiz emeğinin aynı olduğu bilinmelidir. Diğer bir deyişle firma bir yazılım almak için yapacağı bütün çalışmaları, yazılımı yazmak için de yapmalıdır. Yazılımı içeride geliştirme fikri, stratejik bir karar olup, uzun vadedeki hesaplarının yapılması gerekir. Örneğin 10'larca yıllık maliyet hesapları ve firmaya katkı / zararları tartışılmalı, risk analizi çalışmaları bu vadede yapılmalıdır ve her stratejik kararda olduğu gibi bu kararda da %100 doğru bir karar olmadığı bilinmeli ancak hata yapılması halinde, yine her stratejik kararda olduğu gibi maliyetin yüksek olacağı da bilinmelidir. Kişisel olarak tavsiyem, duruma ve kuruma göre çok değişmekle birlikte mümkünse yazılım işini, yazılımdan anlayan kişilere devretmek ve firmanın çalışma alanına konsantre olmasıdır. Aynı bir personel servisi işinin bu konudaki uzman firmalar tarafından daha ucuza ve daha kaliteli veriliyor olması gibi, yazılımın da ilgili firmalara havale edilmesi en doğru karardır. Ancak stratejik öneme sahip yazılımları, kurumların bünyesinde tutması da tavsiye edilmelidir. Daha iyi anlaşılması için uç örnekler vermeye çalışacağım. Örneğin ofis yazılımını hazır alıp kullanmak yerine, bu yazılımı yazmaya çalışmanın neredeyse hiçbir firma için anlamı yoktur. Öte yandan bir satış ve pazarlama firmasının bütün satış stratejilerini, tahmin ve hedeflerini ve planlamalarını geliştirmek için kullandığı kendisine özel yöntemler ve yılların tecrübesi ile geliştirilen yöntemleri de dışarıya açması beklenemez. Her ne kadar günümüzde bu amaçla hizmet veren ve oldukça iyi çalışan yazılımlar mevcut olsa da, kişisel görüşüm bu yazılımların kullanılmadan önce iki kere düşünmesi gerektiği yönündedir. Yukarıda sıralanan yazılım kaynakları arasında seçim yapılırken dikkat edilecek kriterler aşağıdaki şekilde sıralanabilir.

- Maliyet
- Fonksiyonellik
- Destek Sağlanabilmesi

- Desteğin Ulaşılabilirliği
- Tepki Süresi
- Esneklik
- Doküman
- Kurulum Kolaylığı

Yukarıda sıralanan özelliklerin önemi, aslında firmalara göre farklılık gösterebilir. Ancak hepsi, firmaların göz ardı edemeyeceği önemdedir. Örneğin yazılımı paket yazılım olarak almak veya içeride geliştirmek arasında karar vermek istiyor olalım. Bu durumda maliyet hesaplaması, yazılımın zaten firmamız için sunulan fiyatının ötesinde, yazılımın bakım maliyeti, zaman içerisinde çıkacak ilave eklentilerinin maliyeti, ve yazılımı çalıştırmak için gereken diğer donanım ve yazılım ihtiyaçlarının maliyetleri gibi farklı yönlerden incelenmelidir. Örneğin her yazılım bir işletim sistemi ihtiyacı duyar, bu işletim sisteminin lisans maliyeti veya işletim sisteminin yönetiminden sorumlu kişilerin maliyeti bu anlamda hesaplanabilir. Yazılımın içeride geliştirilmesi durumunda esneklik kazanımının çok daha fazla olacağını söyleyebiliriz. Benzer şekilde, yazılımdaki problemlere daha hızlı müdahale edileceği de söylenebilir. Neticede içeride kendi elemanlarımız tarafından geliştirilen bir yazılım ve bu yazılıma istenilen müdahaleyi yapabilen bir ekipten bahsediliyor. Kısacası, yukarıdaki listesi verilen kriterlere göre kurumun yazılım ihtiyacında bir önem listesinin hazırlanması ve limitlerin belirlenmesi gerekir. Örneğin maliyet olarak çıkılabilecek azami maliyet veya şirketin faaliyet alanı göz önüne alındığında, şirketin destek ihtiyacı için geçecek maksimum zaman gibi limitler belirlenebilir. Bu limitlere göre bir yazılım teminine gidilmesi veya içeride geliştirilecek yazılım için bir yapılanmaya gidilmesi daha kolay olacaktır. Örneğin çok hassas zamanlaması olan iş yerlerinde belki de bakım ve problemlere müdahale için ikinci bir yedek veri merkezi (data center) kurulması ve sırf bu işten sorumlu bir ekibin istihdam edilmesi istenebilir. Ayrıca yazılım sağlamak isteyen kurumların genelde RFP (Request for proposal, Teklif talebi) şeklinde gelen taleplere cevap vermesi, piyasada alışıldık bir durumdur. Basitçe ihtiyaç duyulan yazılım, ve bu yazılımın çalışacağı ortam (donanım, ilave yazılımlar) ve gerekli destek ekibi gibi maliyete dahil olan kalemler için bir teklif hazırlanmasını istenebilir. Bu teklifler genelde hata payı içermekle birlikte maliyet hesabı için gereken araştırma aşamasını, yazılım satmak isteyen firmaya yaptırmak açısından da avantajlıdır. Hatta çoğu yazılım evi, kendi ürünlerini kullanan müşterileri ile bu ürünleri kullanmayı düşünen müşterileri buluşturarak tecrübe paylaşımına gidilmesine imkan sağlar. Burada bir parantez açmak istiyorum ve yazılım evlerinin elbette başarılı olan müşterilerini öne sürdüğünü ve yazılım alma aşamasında olan diğer kurumlara bu müşterileri tanıstracağını tahmin etmek çok da zor değildir. İşte tam da bu noktada, eğitimin ve tecrübenin önemi ortaya çıkar. Kişisel olarak belirtmeliyim ki, hayatınız boyunca arkanızda duracak en iyi destek ekibi, yıllarca beraber okuduğunuz ve ödev / proje yaptığınız öğrencilik yıllarınızdaki arkadaşlarınızdır. Şayet iyi bir eğitim aldıysanız, iyi bir de destek ekibiniz olmuş demektir. Elbette iyi eğitimin sonucunda bu arkadaşlarınızın da sizin gibi kilit noktalarda olma ihtimalinin yüksek olduğunu söylemek mümkündür. Bunlara ilave olarak yıllarca piyasa beraber gülüp ağladığımız arkadaşlarınız da bu destek ekibinin bir parçasıdır. Sonuçta firmalar ve ürünleri hakkındaki en iyi istihbarat bu adı konmayan, herkes için gizli bir istihbarat teşkilatı gibi çalışan ve yıllar süren ilişkilerle şekillenen ekip sayesinde sağlanır. Bu yüzden iyi bir eğitim ve iş yerinizdeki insanlarla iyi iletişim önemlidir. İşin mühendislik özellikle de bilgisayar mühendisliği boyutundan bakıldığında, kişisel görüşüm, en önemli ve en çok dikkat edilmesi gereken parametrenin yeniden kullanılabilirlik (reusability) olduğudur. Bu özellik ne yazık ki yukarıdaki listede yok. Bunun sebebi yukarıdaki listenin bir yönetici gözünden hazırlanmış olmasıdır. Ancak teknik seviyede kod yazılacağı kesindir ve yazılacak kodun mümkün olan en yüksek oranda, tekrar ve tekrar kullanılabilir olması aslında yapılan işin ne kadar az emek ve maliyetle çözülebileceğidir.

Kaynak

Laudon K ve Laudon J, (2011) Management Information Systems, 12. Baskı, Prentice Hall, 2011, ISBN: 0132142856

Seker, S. E. (2014), Bilişim Sistemleri Eşitlik Teorisi (Equity Theory), YBS Ansiklopedi, v. 1, is. 2, pp. 17 – 20

Seker, S. E. (2014), KANBAN, YBS Ansiklopedi, v. 1, is. 4, pp. 21 - 25