

1 YAZILIM ÜRÜNÜNÜN TANIMLANMASI VE ÜRETİLMESİ

1.1 BELGELENDİRME

Bahadır ÇINAR

YAZILIM ÜRÜNLERİNDE dokümantasyon çok önemli bir yer tutmaktadır. Bu dokümanlarda kullanıcıya getirilen yenilikler, kolaylıklar açık bir dille anlatılmalı, kullanıcı dokümanlarda her soruya cevap bulabilmelidir. Şirket içinde kalacak dokümanlarda ise ayrıntıya girmeden ürünün üst düzey tasarım ve yazılım yapısı anlatılmalı, tasarım sürekliliği gözetenmelidir.

M.Zafer DOKUZOĞLU

YAZILIM ÜRÜNÜNÜN TANIMLANMASI, üretilmesi ve hatta kullanılması aşamasında belgelendirme özel bir önem taşımaktadır. Bu sayede yazılım bir çok kişinin kullanabileceği, ileride değişen isteklere göre tekrar adapte edilebilecek veya bir kısmı bir başka ürün için de tekrar kullanılacak bir ürün haline gelir.

Yazılım üretilmeden önce yazılımdan beklenen işlevlerin tanımlanması ve bu tanımlamanın herhangi bir belirsizliğe yol açmadan yapılması gerekmektedir. Tanımlanan bu işlevler Yazılım Gereksinim özellikleri adını verebileceğimiz bir dokümanda belgelendirilmelidir.

Yazılım üretilirken Yazılım Gereksinim Özellikleri dokümanındaki özellikleri karşılayacak şekilde tasarlanmalı ve tasarım Yazılım Tasarım Dokümanında belgelendirilmelidir. Bu doküman içeriği o kapsamda olmalıdır ki, daha sonra bu yazılım üzerinde bir diğer kişi veya grup istenilen değişiklikleri yapabilmelidir.

Yazılım kullanıcıya yönelik olarak da belgelendirilmelidir. Bunlar arasında kullanıcı kılavuzları, operatör kılavuzları sayılabilir.

Kemal ENFİYEÇİ

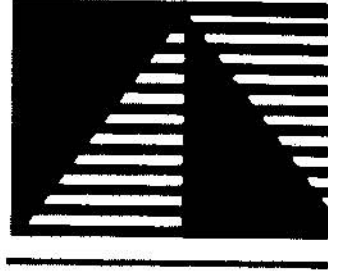
BİR YAZILIM ÜRÜNÜ geliştirirken uygulanması gereken en temel noktalardan birisi de o yazılım ürününün belgelendirilmesidir. Belgelendirme işlemi, yazılım ürününün gelişmesine paralel olarak yürütülmelidir. Yani, yazılımın her aşamasında o aşamayı içeren bir belge hazırlanmalıdır. Bu aşamaları şöyle sıralayabiliriz:

analiz, tasarım, planlama, kod yazılımı, entegrasyon, test ve bakım. Her belge bir sonraki aşama için büyük bir bilgi potansiyeli oluşturmaktadır. Yazılımın güncelliğini koruyabilmesi, varlığını sürdürebilmesi, yeni isteklere cevap verebilmesi bu belgelerin düzenli olarak tutulmasına bağlıdır. Böylece, kolay ve güvenilir bir yazılım ürünü ortaya çıkar.

Belgelerin hazırlanmasında izlenecek yolun da çok iyi tanımlanması gerekmektedir. Bu tanımlamada uluslararası standartlardan faydalanılmalıdır. Bunun sonucu, yazılım ürünü evrensel bir nitelik kazanır. Daha sonra, bu belgelerin yazılımın içeriğine ve uygulama alanına göre uyarlanması gerekmektedir.

Sina KUTLUAY

GELİŞEN TEKNOLOJİYE paralel olarak yazılımın önemi gün geçtikçe daha ön plana çıkmaktadır. Artık donanımların bile programlanabilir olduğu günümüzde, bir sistemin tanımı ve tanımı ister istemez büyük bölümü oluşturan yazılım kısımlarının işlevlerine yönelik olmaktadır. Bir yazılım projesinde, en önemli kısım işlevlerin ve beklentilerin doğru ve tutarlı olarak tanımlanabilmesi olarak ortaya çıkmaktadır. Bu tanımın yeterliliği doğrudan ürünün maliyetini, kalitesini, sonradan değiştirilebilir (maintainable) olmasını etkilemektedir. Aynı öneme sahip bir diğer konu da ürünün belgelendirilmesidir. Tanımlamada ve tasarımda uyulacak belgelendirme standartları ise bütünlüğün korunmasını ve işlevlerin, kodun anlaşılabilir yazılabilmesini getirebilir.



387- ELEKTRİK M MM
MÜHENDİSLİĞİ 1 4 1

**Yazılım
geliştirme
süreci
boyunca,
Yazılım
geliştirme
planında
öngörülen
belgeler, her
aşama sonrası
üretilmeli,
denetlenen
ve
düzeltilmelidir.**

çektir. Bu nedenle yazılım ürününün, her aşamada yeterli düzeyde belgelendirilmesi gereklidir. İşlevlerin belirlenmesi, tasarım, test ve kalite kontrolü, entegrasyon ve üretim aşamaları için belgelendi/ilebilen, yeterli düzeyde dokümanları hazırlanabilen yazılım ürünleri ölü doğmayacak ve yaşam süreleri uzun olacaktır.

F. Mehmet ÖZEL

BATI ÜLKELERİNDE ÜRETİLEN yazılımlarla birlikte ayrıntılı açıklama içeren bir el kitabı sunulmaktadır. Ülkemizde bu konuda yeterli ciddiyet yoktur. Yazılım üreticilerinin, kullanım kitaplarının yazılımların bir parçası olduğu gerçeğini bilmeleri gerekir.

Ayrıca, devletin de bu konuda düzenlemeler getirmesi gerekir. Örnek olarak, hiçbir yazılımın el kitabı olmadan piyasaya sunulamayacağı gibi.

Oğuz UNCU

GÜNÜMÜZDE YAZILIM geliştirme olabildiğince karmaşık/aşmış ve bunun sonucu da artan yazılım geliştirme sonrası bakım maliyetleri olmuştur. Bu maliyetleri düşük tutabilmenin ve geliştirilmiş olan yazılımın yaşamını daha sağlıklı olarak tutabilmenin kanımca en doğru yolu yazılım gerektiği şekilde belgelendirmektir. Yazılım belgelendirilmesi işi aslında yazılım geliştirme yaşam sürecinin ayrılmaz ve bütünleyici bir parçası olmalıdır. Yazılım geliştirme süreci boyunca, Yazılım geliştirme planında öngörülen belgeler, her aşama sonrası üretilmeli, denetlenen ve düzeltilmelidir. Kuşkusuz sözkonusu işlemler yazılım geliştirme maliyetini arttıracak, süreyi de uzatacaktır. Bu konuda ASELSAN'da önemli bir yol alınmıştır ve belgelendirmeye dayalı yazılım geliştirme benimsenmiştir.



1.2. TASARIM İLKELERİ

Müjdat ALTAY

AVRUPA TOPLULUĞU içinde bilişim sistemlerinde yazılım faktörü toplam maliyetin 1980'de %45'ini kapsarken, 90'lı yıllarda bu fiğür %80'e ulaşmıştır.

Günümüzde yazılım faktörünü her projede görmek mümkündür. Teknolojik gelişimi sürdüreceğ ürünlerin gerçekleşmesi yazılımın doğru bir şekilde kullanılması, geliştirilmesi ve denetimine bağlıdır.

Şebnem ARIK

BELİRLİ BİR YAZILIM ürününün tasarlanması bir dizi ön çalışmayı gerektirmektedir.

Programın yazılacağı konuda önceden oluşmuş talepleri incelemek, konuya ilişkin gelen teklifleri değerlendirmek, varolan yazılımlardaki kullanıcı problemlerini takip etmek, konu hakkındaki yeni gelişmeleri öğrenmek, programı üzerinde koşacağı donanımdan ve hatta işletim sisteminden mümkün olduğu kadar bağımsız hale getirmek, yazılımın hızı ve çalışabileceği minimum bellek kapasitesi gibi parametreleri önceden değerlendirip önem sırasını saptamak ve buna göre algoritma geliştirmek, gelecekte yapılacak değişikliklere ve eklemelere uygun, modüler yapıda program tasarlamak, programın yazılacağı dili esnek ve yaratıcılığa açık diller arasından seçmek, tasarlanan yazılımın istenilen standartlarda olabilmesi için uygulanması gereken çalışmalar.

Ahmet BELGÜL

YAZILIM ÜRÜNLERİNDE performansı ve güvenilirlik yanında kullanım ve müşteri isteklerine uygunluk gibi unsurlar giderek önem kazanmakta, yazılım ürünlerinin boyutu büyümekte ve karmaşıklaşmaktadır. Bunun getirdiği sorunların üstesinden gelmek için de S/W Teknolojisi, tasarım ilkeleri ve olanakları, bunun yanında geliştirme süreci iyileştirilmeye çalışılmaktadır. S/W ürünlerinin bir sığrama yapılabilmesi ise tıpkı H/W IC'ler gibi S/W IC'leri mümkün kılan teknoloji ilerlemeleri sayesinde olacaktır.

BİR YAZILIM ÜRÜNÜNÜN üretilmesi, bu konuya yakınlığı olan ve olmayan pek çok kimse tarafından yalnızca ya da daha çok "Kodlama" olarak algılanır. Bu doğru değildir. Yazılım ürünü tanımlanması ve üretilmesi birkaç aşamadan oluşur ve bunların hepsi önemlidir.

Tanımlama için öncelikle gereksinimler eksiksiz toplanmalı ve değerlendirilmelidir.

Gereksinimlerden yararlanılarak yazılım ürününün işlevsel özellikleri belirlenir. Belirlenen bu özellikler, bilgisayar ortamına aktarılarak, ürünün sistem tasarımı gerçekleştirilir. Sistem tasarımında, yazılım ürününün terminolojisi, ürün oluşturacak bölümlerin işlevsel tanımları, veri girişi, veri saklama, erişim ve paylaşım yöntemleri ile ürünü destekleyecek yayınlar ve standartlar saptanır ve belgelenir.

Tasarımdan sonra, önceden saptanan programlama yöntemleri/dilleri kullanılarak, ürünün kodlanması aşamasına geçilir. Kodlamada, ürünün ayrı program birimlerinden ya da alt-programlardan oluşmasına dikkat edilir. Her program birimi ya da altprogramın tek bir işlevi olmalı ve başında işlevi ile ilgili bilgi yer almalıdır. Kodlamada da uygun ve standart bir belgeleme yöntemi kullanılmalıdır,

Kodlama işleminin ardından sinama işlemi yapılır. Sinama, iki aşamada gerçekleştirilir, önce, program birimleri sinanır. Bu sinamada, program biriminin, tasarımında tanımlanan işlevlerini eksiksiz gerçekleştirmesine ve kullanıcı arabiriminin (ekran tasarımları, iletiler, yönergeler, yardım gibi..) standart olmasına dikkat edilmelidir.

Sinamanın ikinci aşaması sistem sinamasıdır. Bu aşamada, yazılım ürünü yayınlarıyla birlikte bir bütün olarak sinanır. Yazılım ürününün yayınlarıyla tutarlı olması önemlidir. Yazılım ürününde, hatadan sakınma ve korunma, doğrulama yöntemlerinin de

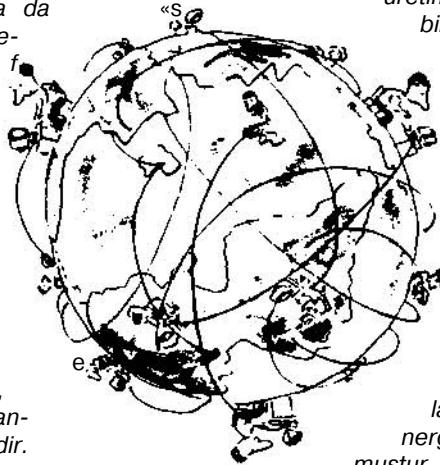
kullanılmasına özen gösterilmelidir.

Belgelendirmeye ilgili olarak, yazılım ürününün kullanıcıyı destekleyen yayınları olmalıdır. Ayrıca, ürünün geliştirme aşamalarını içeren yayınlarının da olmasında yarar vardır.

YAZILIM, kazandığı öneme paralel olarak gittikçe daha karmaşık ve fazla satır sayısına sahip hale gelmektedir. Bu durumda boyuta bağlı olarak çeşitli sayıda kişinin bir arada koordineli olarak çalışmasını gerektiren yazılım tasarımının bazı ilkelere bağlı olmadan başarıya ulaştırılabilmesi düşünülemez. Dolayısı ile çeşitli düzeylerde ilkelerin önceden ortaya konulması ve bunlara uyulmasının denetlenmesi önemli bir zorunluluktur. Bu ilkeler; proje veya firma bazında olabileceği gibi analiz, tasarım, kodlama veya test düzeylerinde kuralları da içermelidir.

HIZLA GELİŞEN YAZILIM ve dona-

mm teknolojisi nedeni ile yazılım üretiminde kullanılmak üzere bir çok yöntem, yönerge ve standart geliştirilmiştir. Bu gelişim içinde, gerek askeri gerekse sivil projeler konusunda çalışan ASELSAN Bilgisayar Hizmetleri ve Yazılım Müdürlüğü, zaman içinde yazılım geliştirme konusunda varolan yaklaşımları özümseyerek, özelleştirmiştir ve ASELSAN içinde kullanılabilir standartlar, yönergeler ve yöntemler oluşturmuştur. Askeri projelerinde zorlaması ile DOD 2167A standardı incelenmiş ve ASELSAN'da kullanılmak üzere uygulanmıştır. Ortaya konulan bu yöntem, yönerge ve standartlar bir projede uygulanmış, halen gündemde olan bir diğer projede de uygulanmaktadır.



**Geleceğin
bilgi
toplumunun
gereksinimlerini
karşılatabilmek,
yazılım
teknolojisinin
ürün yazılımları
ve bunlara ait
yazılım destek
sistemleri için
geliştirilmesine
bağlıdır.**

1.3. KALİTE, PERFORMANS, GÜVENİLİRLİK

Bahadır ÇINAR

SANAYİ ÜRÜNLERİ ne kadar iyi test edilirse edilsinler, kullanım sırasında bazı problemlerin çıktığı çok görülen bir olaydır. Yazılım alanında bu problemler kullanıcı hatalarından kaynaklanabildiği gibi, tasarımdan da olabilmektedir. O yüzden yazılım üreten şirketlerin çok iyi bir problem raporlama yapısı kurmaları gerekmektedir.

Sina KUTLUAY

BİR YAZILIM ÜRÜNÜNÜN kalite, performans ve güvenilirliği belgelendirme ve tasarım ilkeleri ile yakından ilgili olduğu gibi kullanılan altyapı, geliştirme sistemi, metodolojiler ve insan kalitesine de doğrudan bağlıdır. Bu parametrelerin düzeyi yazılımın uygulama alanı ile doğrudan ilgilidir ve ürünün maliyetini, geliştirme süresini doğrudan etkiler.

1.4 UYUM SINAMASI, BİÇİMSEL BETİMLEME TEKNİKLERİ

**L. Güzide
ARABACKDĞLU**

DONANIM VE YAZILIM teknolojilerindeki gelişmelerin ve gereksinimlerin sürekli birbirini beslemesi sonucunda günümüz toplumundaki kullanılan her tür üründe yazılımın ağırlığının giderek arttığını görmekteyiz. Reklamaların "akıllı" ürünler ile dolup taşmasına neden olan bu eğilimin geleceğin bilgi toplumunun gereksinimleri paralelinde giderek hız kazanacağından şüphe yoktur. Bu nedenle özellikle yüksek teknoloji gerektiren ürünlerde ürün yazılımlarının tanımlanması ve üretiminde kullanılan yöntemlerin yeniden ele alınması gereği duyulmaktadır.

Geleneksel olarak ürün yazılımları ile ilgili gereksinimlerin tanımlanmasında doğal diller kullanılmaktadır. Tanımlamalardaki hata ve/veya eksikler ancak tümleştirme testinde saptanabilmekte, bu safhada gerekli düzeltmenin yapılması ise zaman kaybına ve maliyet artışına neden olmaktadır.

Yazılım üretimi için de aynı durum sözkonusudur. Üretim ile ilgili problemler tümleştirme testinde görülmekte, üretimin ve testin tekrarı gerekmektedir.

Tanımlama ile ilgili problemlerin minimuma indirilmesi için tanımlamalarda netlik, otomatik kontrol ve dönüşüm imkanı sağlayan formal betimleme tekniklerine dayalı dillerin kullanılması (Specification Description Language, Message Sequence Chart, vb.) yoluna gidilmelidir.

Yazılım üretimi ile ilgili problemlerin üretim safhasında saptanabilmesi için de formal betimleme tekniklerine dayalı dillerin gücünün "object-oriented" teknikler kullanılarak genişletilmesi ile elde edilen diller (Objective C, C++, vb.) seçilmelidir. Bu diller kavramların modellenmesi yolu ile prototiplerin üzerinde test yapma imkanı sağlamakta, böylece problemlerin saptanması ve çözümü tümleştirme testinden önce gerçekleştirilebilmektedir.

Geleceğin bilgi toplumunun gereksinimlerini karşılayabilmek, yazılım teknolojisinin ürün yazılımları ve bunlara ait yazılım destek sistemleri için geliştirilmesine bağlıdır.

Mehmet BARAY

YAZILIM ÜRÜNLERİNİN belirlenmesi, tasarım ilkeleri, testing, formal tanımlama gibi konularda yurdumuzda bilgi birikimi bulunmamaktadır. Bazı kuruluşlarda bu tür çalışmalar yapılıyorsa da bunu genellemek mümkün değildir. Bu konularda bazı standartların saptanması ve yeni geliştirilecek ürünlere uygulanması ileride yapılması beklenen yazılım ürünlerinin geliştirilmesinde yararlı olacaktır.

Semih BILGEN

TÜM TEKNOLOJİ ALANLARINDA olduğu gibi yazılım üretiminde de kuramsal gelişmeler, uygulama ve özellikle de ticari uygulamaların beş-on yıl önünde gitmektedir. Örneğin, yapısal programlama 1970'lerin başında irdelendi, bugün ticari uygulamalarda zorunlu; nesne yönelimi, 1980'lerde ortaya atıldı, pazarda egemenlik kurması yakındır...

Gerçekleştirilen yazılım ürününün, istenene ne ölçüde uyduğu, konuyla ilgilenenlerin hep

kaygıyla üzerinde durduğu ve çözümünü bir türlü kesinliğe kavuşturulamayan bir sorundur. Özünde yazılım kullanıcısıyla üreticisi arasındaki iletişimin niteliğine bağlı olan uyum sorunu, bu iletişimde kullanılan dil ne ölçüde anlaşılır, tekanlı ve yetenekli olursa o ölçüde çözülebilecektir. Başka bir deyişle kullanıcı, istemlerini ne denli açık, belirgin, eksiksiz, tutarlı ve yanlış anlaşılacak biçimde ortaya koyarsa, üreticinin doğru iş yapması o denli kolaylaşır. Ancak bir dilin zenginliği ve yetkinliği çoğu kez bağlam bağımlılığı, çok anlamlılık, vurgulama ya da bir kavramı karşılayan birden fazla sözcüğün varlığı gibi özelliklerle birlikte ortaya çıkmaktadır.

Günlük iletişimde kullanılan doğal diller (ör., Türkçe, İngilizce, vb.), çoğu kez yazılım gereklilerinin ortaya konmasında yinelenemeler, tutarsızlıklara ve kaçınılmayan çok anlamlılıklara yol açabilmektedirler. İşte bu nedenlerle, bilgisayar komutlarını anlaşılır biçimde ortaya koymaya yarayan programlama dilleri gibi, kullanıcının yazılım istemlerini belirtmekte yararlanması için ortaya atılan Biçimsel Betimleme Dilleri (Formal Description Languages) ya da teknikleri, bir yandan yazılım üretimin, diğer yandan ise uyum sınavının olmazsa olmaz ögesi konumuna adaydır. Bugün, "ciddi olarak" yazılım üretimi yapan bir kuruluşun, belirtim ve tasarım aşamalarında bu biçimsel yöntem ve araçlardan birini kullanmaması düşünülemez.

1970'li yıllarda genel yazılım uygulamaları, 1980'li yıllarda gerçek zamanda çalışan özel nitelikli uygulamalar ve iletişim sistemleri gibi birçok değişik alanı hedefleyen çok çeşitli Biçimsel Betimleme yöntemi ortaya atılmış ve belli ölçülerde kabul görmüşse de günümüzde bu alandaki öneri, araştırma ve geliştirme çalışmalarını yoğun biçimde sürmektedir.

Kemal İNAN

MİKROELEKTRONİĞE DAYALI İLETİŞİM
ve bilgisayar teknolojilerinde yer alan baş döndürücü gelişmeler şu sorunu ortaya çıkardı: sürekli değişen bir teknolojik ortamda ve çok sayıda üreticinin cihazlarını içeren bir bilgisayarlı sayısal iletişim sisteminde sözkonusu cihazların beraberce çalışabilmesi nasıl sağlanacaktır? Bu sorunun geçmiş benzerlerinden farkı bu cihazları birbirlerine bağlayan

arabirimlerin tanımının fiziksel tanımlardan (örneğin 0 volt-5 volt arasında gerilim çıktısı üreten bir cihaz gibi) çok daha karmaşık mantıksal tanımlara dönüşmüş olması, örneğin OSI (Open System Interconnection) modelinde session katmanının hangi servisi sağlayacağı üzerine geliştirilen standart, örneğin EStELLE betimleme dilinde yaklaşık 25 sayfalık bir biçimsel tanımlama (daha doğrusu tanım programını) gerektiriyor!

Biçimsel betimleme teknikleri özellikle karmaşık dağılımlı sistemlerin yazılım üretiminde şu işlevle» görebiliyor:

1) Arabirim standartlarının bir üst düzey yazılım olarak hazırlanabilmesi için sözkonusu biçimsel dilin kullanılması.

2) Üst düzey yazılımdan gerçek kullanılacak yazılımlara kadar inen belirtim ayrıntılandırması (spec'ificat'ion refinement) basamaklarının yani yazılım üretiminin otomatikleştirilmesi (en yoğun araştırma bu alanda).

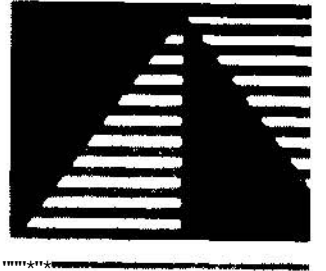
3) Üretilen yazılımlara varış metodolojisi her ayrıntılandırma basamağında bir önceki basamak ile mantıksal tutarlılığı sağlayan sınavın yapılabilmesi, ve böylece son yazılım ürününün güvenilirliğinin ve üretim veriminin artması.

4) Bugünkü varılmış tekniklerin gerçekçi bir sistem üzerinde ancak kısmi doğruluk sonuçlarını garanti edebilmesi nedeni ile üretilen yazılımın standartlara uyumunu sağlayacak test yöntemlerinin yarı-otomatik olarak üretilebilmesi, yani uyum sınavında biçimsel tekniklerin kullanılması.

5) Yazılım üzerinde yapılacak değişikliklerin (version change, maintenance) üst düzeydeki belirtimde yer alması ile beklenmeyen dolaylı hatalara yol açılmamasının sağlanması.

6) Bir yazılımın üretilmesinden önce üretici ile müşterinin yazılımdan istenilen işlevler konusunda üzerinde görüş birliğine varabilecekleri formal (biçimsel belirtim) ya da yan-formal bir dokümanın üretilebilmesi. Bu doküman, üretim öncesi anlaşmanın sağlanması ile yazılım üretiminin maliyetini azaltabiliyor.

Biçimsel betimleme teknikleri olarak bilinen alan, yazılım teknolojilerinin gerek kuramsal



YAZILIM ÜRETİMİ VE KULLANIMINDA ÇALIŞACAKLARIN TANIMLANMASI VE BU KİŞİLERİN EĞİTİLMESİ

2

Üst düzey tasarımcıyı yazılımın kendisinden biraz soyutlayıp, konuyu her açıdan elit bakış açısı ile ele alan bir insan olarak tanımlamak gerekir.

gerek uygulama yönünden en Heri sayılacak dallarından biridir. Bu nedenle de bu araştırma alanının ürünleri yakın bir gelecekte karmaşık yazılım üretiminin katma değerini katlayacak güce sahiptir. Türkiye'deki ileri teknoloji firmaları (iletişim ve askeri elektronik alanlarındaki firmalar) dünya piyasalarında rekabet etmedikleri için bu konu üzerine yeterince eğilmek gereğini şimdiki duymamaktadırlar.

1.5 BENZERİ BAŞKACA KONULAR

Oğuz UNCU

GÜNÜMÜZ YAZILIM TOPLULUĞUNUN çokça konuştuğu konulardan birisi de Nesnel Kökenli Yazılım geliştirmedir. Bu konuda henüz var olan bütün belirsizliklere karşın, akademik olarak yapılan çalışmalarla birlikte, endüstride ve yazılım geliştirme evlerinde birçok deneyimler gerçekleştirilmektedir. Nesnel Kökenli yazılım geliştirme konusuna üniversitelerin, yazılım geliştirme sektörünün ve endüstrinin değişik yaklaşımları vardır ve bu yaklaşımlar da ya daha Nesnel Kökenli yazılım geliştirme öncesi deneyimlerden ya da Nesnel Kökenli yazılım geliştirme deneyimlerinden kaynaklanmaktadır.

2.1 ÜST DÜZEY TASARIMCIUĞI, AYRINTILI TASARIMCIUK

Günay GÜNER

DÜNYADAKİ KÜLTÜREL VE SOSYAL gelişmelere paralel olarak ortaya çıkan birtakım ihtiyaçlar çeşitli alanlarda ama özellikle teknoloji alanında yeni fikirlerin ve bunları takip eden yeni uygulamaların doğmasına sebep olmaktadır. Yine günümüzde, teknolojinin ulaştığı ivme sayesinde ortaya atılan yeni bir fitlin uygulama alanı bulup pratik bir ürün olarak markete sunulması sadece birkaç yıl almaktadır. Ve bu ürünlerin hemen hepsinde yazılım, temel öğelerden biri olarak bulunmaktadır.

Bu fikrin ortaya çıkıp, bunun bir yazılım ürünü olarak meydana getirilmesine kadar olan süreç bir bütün olarak ele alındığında ortaya çıkan panoroma şu şekilde özetlenebilir:

- 1) Fikrin ortaya çıkması,
- 2) Bu fikrin değerlendirilip, ilgili alanda uygulama düşüncesinin olgunlaşması,
- 3) Değerlendirilen fikrin ayrıntılı olarak incelenip ilgili ürüne nasıl uygulanacağını ortaya konması,
- 4) Ayrıntılı olarak ortaya çıkmış olan yapıyı yazılıma dönüştürmek,
- 5) Ortaya çıkan yazılımın test edilmesi ve
- 6) Müşteri ve/veya üretici firma tarafından kullanımı.

Ortaya konan ham bir fikrin yazılım ürünü olarak ortaya çıkmasına kadar geçen sürecin en başında bu fikrin alınıp pratiğe uygunluğunun tartışıldığı süreç bulunmaktadır. Bu süreçte yer alan insanlar da üst düzey tasarımcı grubunu oluşturmaktadır. Görüşümce, üst düzey tasarımcıyı sadece tanımında yer alan "tasarımcı" kelimesi ile değerlendirmem gerekir. Üst düzey tasarımcıyı, bu yazının ilk paragrafında da vurgulamaya çalıştığım gibi, konunun bütününden, fikrin kendisinden ayrı tutamayız. Diğer bir deyişle üst düzey tasarımcıyı yazılımın kendisinden biraz soyutlayıp, konuyu her açıdan elit bakış açısı ile ele alan bir insan olarak tanımlamak gerekir. Genelde



üst düzey tasarımcıyı yazılımın içinde düşünmek istememe rağmen, tanımladığım eleman tipine karşılık gelen bir seçenek sunulmadığı için tanımlı böyle sunmak gerektiğini duyduğumu ve bunu bir eksiklik olarak değerlendirdiğimi bildirmeden geçemeyeceğim.

Bu tip insanın yetişmesine gelince. Bu alanda bir eğitim verilemeyeceği açıktır. Çünkü üst düzey tasarımcı bir noktada ilgili alandaki ilk insandır. Bu konuda bir eğitim, alt yapıda verilemeyeceğine göre üst yapıya yönelmek gerekir. Yani, bir şekilde o tanıma uygun vasıfları kazanmış insanlara yönelmek gerekir. Bu kişiler akademik kariyerlerini ilgili alanlarda sürdürüp, bu alanlarda çeşitli proje çalışmalarında bulunan veya yazılım sanatı ile ilgili konunun alt basamaklarından başlayarak alanlarına pratik olarak hakim olmuş kişilerdir. Bu tanım içinde yer alan insanların normal aktivitelerinin bir bölümü eğitim olarak algılanabilir. Buna örnek olarak da bu insanların, uluslararası standart koyucu kuruluşların çalışmalarında yer almaları veya aynı alanda çalışan diğer kuruluşlarla temaslarda bulunup, görüş alışverişinde bulunmaları verilebilir.

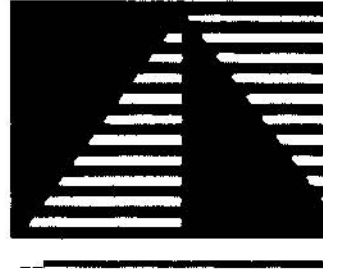
Bir fikrin üst düzeyde ele alınıp değerlendirilmesinden sonra bunun uygulama haline getirilmesine karar verirse, bunun ürüne uygulanabilirliği ana hatlarıyla belirtilmeli ve uygulamanın nasıl yapılması gerektiği kalın çizgilerle özetlenmelidir. Bundan sonra sıra ayrıntılı tasarımcıya gelir.

Elinizde bir ürün var. Bu ürünün geliştirilmesi iki yönde olabilir. Ya ürünün bütünü etkileyen bir iyileştirmeye gidilir, ya da ortaya çıkan yeni fikirlerin uygulamaya dökülmesi söz konusudur. Her iki durumda da yapılacak çalışmanın tüm hatları ile ortaya konması gerekir. Ayrıntılı tasarımcının tanımı da burada yatmaktadır. Birkaçyüz satırlık bir yazılım hazırlarken, belki ana hatları dışında fazla düşünmeden kodlamaya geçilebilir. Ama bunun adına yazılım demek neredeyse imkansızdır. Projenin boyutları yüzlerce kez büyütüldüğünde durumun vahameti daha da açık olarak ortaya çıkar. Kodlamaya geçilmeden önce yapılacak ayrıntılı tasarımın boyutları son derece geniş tutulacaktır. Bu noktada projenin gerektirdiği bütün yapı, tüm ilişkiler, kullanılacak kavramlar açık olarak, ayrıntısı ile ortaya konacaktır. Bütün bunları yapabilmesi için ayrıntılı tasarımı yapan kişilerin ürüne son derece

hakim olmaları gerekir. Bu insanlar da konuya son derece hakim, fakat konuya yaklaşımları özellikle spesifik bir ürün açısından olan insanlardır. Bunların eğitilmesi tabii olarak zaman içinde "on-job" training olarak gerçekleşecektir.

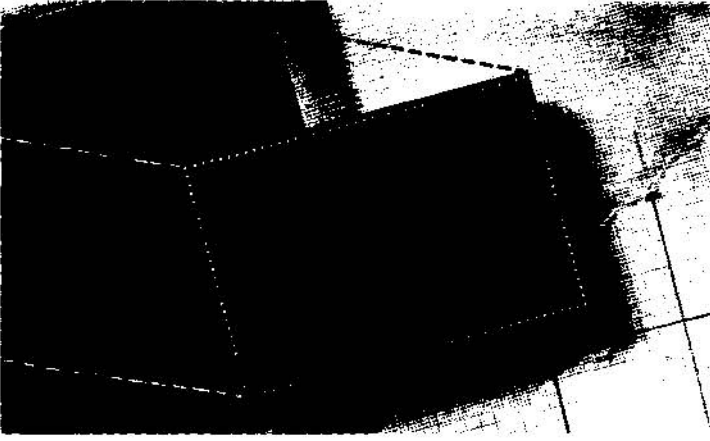
Artık sıra kodlamaya gelmiştir. Bu aşamada çalışacak insanlardan beklenecek en önemli şey ürünlerini tanımlarıdır. Zaten bu noktadan itibaren çalışan tüm grupların özel olarak çalıştıkları ürüne veya ürünün boyutları çok büyükse, özel olarak ilgilendikleri ürün parçasına hakim olmaları beklenmelidir. Programcı denilen grup ayrıntılı olarak ortaya konan istekleri o yazılım projesi açısından ürüne en iyi şekilde uygulayabilmek için ürünü tanımak yanında yazılım felsefesine hakim olması beklenmelidir. Bu tip elemanın çıkış noktası bugün için üniversitelerimizin neredeyse müspet bilimlerle ilgili tüm branşlarıdır. Elbetteki özellikle yazılım alanında üniversite eğitimi görmüş elemanların varlığı, yazılıma teknik olarak yakın olmaları açısından pozitif bir puandır, ancak sonuçta yazılım bir matematiksel zeka oyunudur. Ve herkese açık olmalıdır.

Yazılım projesinin kodlama aşamasını da geçtiniz. Şimdi sıra bir yazılım projesinin en önemli aşamalarından biri olan ama nedense genellikle pek önemsenmeyen kısmına, test etmeye gelmiştir. Bir yazılım projesinin test etmek sadece istenenin yapıp yapılmadığını test etmek değildir. Beklenmeyen durumların değerlendirilmesi ve test edilmesi gerekir. Okul yıllarında yazılan birkaç satırlık programlarda bile son derece doğruymuş gibi gözükken mantıkların koşturulduğunda nasıl büyük hatalar verdiği unutulmamalıdır. Yazılım ile ilgili tüm kitapların ilk sayfalarında yazılımın birinci kuralı gibi ortaya konan "hatasız program yazılamaz" düşüncesi özellikle hacim açısından çok büyük olup zamanı kısıtlı projelerde daha da büyük bir önem taşımaktadır. (Piyasa koşullarında da her zaman zaman kısıtlaması vardır.) Dolayısıyla, bu düşünceden yola çıkılarak teste büyük önem verilmesi ve bu iş için ürüne olduğu kadar test araçlarına hakim elemanların kullanılması gerekir. Bu eleman küçük ölçekli bir çalışmada programcının kendisi de olabilir. Ancak çok büyük ölçekli bir projede özel olarak yetiştirilmiş elemanlar bulundurulmalıdır. Bu konuda üniversitelerin eğitim programlarına bir veya



iki dönemlik dersler koymak şu an için olmasa bile yakın gelecekte Türkiye'de yazılım alanında büyük gelişmeler olduğu takdirde gerekli olacaktır. Şimdilik ihtiyacı olan kuruluşlar bu tip elemanları kendi bünyelerinde eğitmek durumundadır.

Test, bir yazılım ürününün gerçek anlamda markete sunulmasından önceki en önemli süreçtir. Burada saptanan hatalar programlama sürecine in put olarak geri dönecektir. Sonunda, ürünün piyasaya çıkabilmesi için gerekli olan şartları sağladığına inanıldığında, kullanıcıların hizmetine sunulacaktır. Bu aşamada kullanıcı grubu kavramı ortaya çıkmaktadır. Genellikle ürünü satın alan kuruluş dahilinde bu ürünün günlük kullanımını gerçekleştiren insanlar olarak tanımlayabileceğimiz bu grup elemanların kaynağı yine üniversitelerin ilgili alanlarından mezun olanlar olacaktır. Bu insanların belli bir ürünün kullanımında uzmanlaşması gerektiğinden o ürünün kullanımı ile ilgili olarak ürünü gerçekleştiren kuruluş tarafından eğitime tabi tutulmalıdır. Bu hususta ayrıca yine ürünü gerçekleştiren firma tarafında gerekli dokümanlar, kullanıcı kılavuzlarının sağlanması gerekir.



Sina KUTLUAY

YAZILIM GELİŞTİRME yaşam döngüsünde en temel kısmı üst düzey tasarımcılığı oluşturmaktadır. Bu aşama genellikle hızlı geçilmekte, yapı ve veri akış analizleri ile vakit geçirilmeden kaba bir tanımlama ile diğer aşamalara geçilmektedir. Halbuki sonuçta oluşacak ürün için bu önemli aşamada yeterli çalışma yapılmalı, tasarımı yönlendirecek belgeler hazırlanmalıdır.

Ayrıntılı tasarımcılık aşaması ise ilkinde göre daha değişik disiplinler gerektiren ve kodlama süresini ve ürünün kalitesini belirleyen bir süreci oluşturur.

Her iki aşamada çalışacak kişilerin metodolojiler, analiz ve tasarım yöntemleri gibi konularda konulmuş kurallar ve kriterleri de öğrenecek ve özümseyecek şekilde eğitilmeleri başarılı ürünlerin kısa sürede ortaya konulmasında önemli bir faktör olacaktır.

Selim SARPER

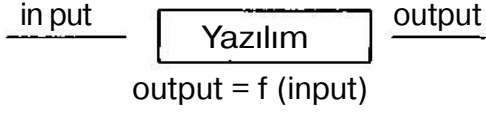
ÖZELLİKLE ÜST DÜZEY TASARIMCILIĞI kavramı üzerinde durmak istiyorum. Sizin konunuzun ana başlığında da görülmekte olduğu gibi, siz kendinizi tamamen yazılımla sınırlamakta mısınız? Yani bir "System Design" diyebileceğimiz bir faaliyette rol almak durumunda mısınız? Yoksa "System Design" çalışması sonucunda, tamamen (100%) yazılım ağırlıklı çalışmalar tanımlandıktan sonra mı devreye giriyorsunuz?

"System Design" yapan ekip -mesela Telekomünikasyonda- şebeke mimarisi, işaretlemesi ve CCITT, ETSI gibi örgütlerin çalışmalarından haberdar olmak, onları yorumlamak ve ne yapılması gerektiğine karar vermek durumundadır.

Elbette bu çalışmaların sonucunda, yazılımın da üst düzey ve ayrıntılı tasarımı gerekmektedir. Herhalde en akılcı çözüm, yazılımdaki üst düzey tasarım ekibini "System Design" ekibi ile birlikte çalıştırmak olsa gerek.

M.Danis UNVERDİ

1) YAZILIMLA UĞRAŞANLAR, hangi seviyede olurlarsa olsunlar, şunu hissetmelidirler: Dünyanın en karmaşık yazılımı gerçekte çok gelişmiş bir OYUNCAKtan başka bir şey değildir. Bu sözümle yazılımın DETERMİNİST tabiatının altını çizmek istiyorum. Ne kadar karmaşık olursa olsun, her bilgisayar programı, çıktısı bütünüyle girdisi tarafından kontrol edilen bir kara kutu (veya bir FONKSİYON) olarak görülebilir.



Bütün mesele, hiç atlamadan bütün girdileri saptamak ve yazılımın gerçekleştirdiği fonksiyonunun ne olduğunu açığa çıkarmaktır.

Basit görünen bu konunun üzerinde duruyorum. Çünkü yazılımla uğraşan birçok kişinin başkaları (özellikle de yabancı memleketlerdeki yabancı şirketler) tarafından yazılan programlara esrarlı, içine girilemez konular olarak baktığını, birçoklarının da aksine, ne olduğunu tam anlamadan, noktasal bir incelemeden sonra (belki de daha derin bir incelemeyi imkansız gördüklerinden) bir programı değiştirebilecekleri kanısında olduklarını düşünüyorum.

2) İyi bir yazılım üretmek için, ille de üniversite mezunu olmak gerekmediğine, düşünce yapısının (METOT VE DİSİPLİN) en önemli faktör olduğuna inanıyorum. Bu söylediğim bilhassa küçük ve orta boy yazılımlar için geçerli. Bu tip yazılımlarda, sözünü ettiğim özelliklere sahip bir tek kişi (bu çok küçük bir grup da olabilir) üst düzey tasarım aşamasından kodlamaya kadar, yazılımın her adımını gerçekleştirebilir.

Büyük çaplı yazılımlarda da, çalışmayı gerçekleştirecek grubun mümkün olduğu kadar KÜÇÜK tutulması gerektiği kanısındayım.



Grup ne kadar büyürse, grup içindeki koordinasyonu sağlamak o kadar güçleşecektir. Bir örnek vermek gerekirse, 5 kişinin 20 ayda biteceği bir yazılımı, 10 kişi 10 ayda değil, 15 ayda (belki daha bile uzun bir zamanda) bitecektir. Yazılımın kalitesinin daha kötü olması da rahatlıkla beklenebilir.

3) Program yazılımcılığının üstünde ayrıntılı tasarımcılık, üst düzey tasarımcılığı gibi konuların varlığını, ancak çok büyük şirketlerde, başka çare kalmadığı için başvurulan bir çözüm olarak görüyorum. Bu durumda bile, ayrıntılı tasarımcılar ve üst düzey tasarımcılar mutlaka çekirdekten yetişmiş olmalı ve yapılması için önlerine iş koydukları alt seviye çalışanların (program yazarlar) pratikte karşılaştıkları zorlukları mutlaka yaşamış olmalı. Unutulmamalıdır ki, sonuçta çalışacak olan program üst düzey tasarımcının tasarımı değil, alt seviye programcı tarafından yazılan kodun bizzat kendisidir. Her durumda, üst seviye tasarımcılar ürettikleri tasarımı bizzat önlerine alıp, en alt seviyeye kadar inebilme yeteneğine sahip olmak zorundadırlar.

Yazılım, her seviyede çalışan kişiler tarafından biraz mühendislik bilgisi, ama — daha çok sağlam bir düşünce yapısı gerektiren bir SANAT olarak algılanmalıdır.



2.2 PROGRAM YAZILIMCI LIĞI, PROGRAMCILIK

Tülay ALUÇ

YAZILIM FİZİKSEL ÖZELLİKLER içermeyen, insan fikirlerinin ifade edildiği bir üründür. Yazılım mühendisliği ise yazılım karakteristiklerini yansıtan kavram ve teorilere dayalı, yeni bir mühendislik disiplini. Bu disiplinde esas alınan, insanın akılla ilgili faaliyetleri (intelligent human activities) ve bunların bilgisayar mimarisi ile olan bağlantı ilkeleridir. Geleneksel yazılımcılıkta kalite ve verimlilik, yazılım mühendisinin kapasitesi ile sınırlı kalmaktadır. Günümüzde ise yazılımın kalite ve verimliliğini insan kapasitesinin sınırlarının ötesinde arttırmak amacıyla, tıpkı malzeme üretiminde olduğu gibi, yazılım üretim otomasyonu için kurallar geliştirilmiştir.

Yazılım artık bireyin değil, bir ekibin ürünü

**Yazılım
üretimini
endüstriyel bir
üretim süreci
olarak
değerlendiril-
mesi ve
devletin de bu
üretim
kârlılığı için
gerekli yasal
düzenlemelere
gitmesi
gerekir.**

olmakta ve çoğu yazılım projesi değişik yetenek ve becerilere gereksinim duymaktadır. Bu da giderek yazılımın çeşitli alanlarında uzmanlaşmayı gerektirmektedir. Yazılım teknolojisindeki ilerlemeye paralel olarak eğitimin de güncelleştirilmesi, temel eğitim ve uzmanlık eğitimlerinin kapsamlarının yenilenmesi zorunlu olmaktadır.

Bu kapsamda değişik bilgisayar sistemlerinin özellikleri tanıtılmalı, çeşitli düzeylerdeki bilgisayar dilleri kullanım alanları açısından kıyaslanmalı, hata arındırma teknikleri ile yazılım kalite, performans ve güvenilirlik unsurları ele alınmalıdır. Değişik yazılım alanlarının gerektirdiği teorik ve uygulamalı eğitim verilmelidir.

Selim SARPER

PROGRAM YAZICILIĞI, kimileri tarafından çok önemli, kimileri tarafından ise daha az önemli bulunur. Sonuçta hedeflenen üründe çalışacak olan "şey" programcıların yazmış olduklarıdır. Bu nedenle de çok önemlidir.

Ancak tasarım etabında verilen kararlar yanlışsa, programcının yapmış olduğu bizi doğruya götüremez. Bu nedenle de programcılık tasarımcılığa göre ikinci derece önemli gözükmemektedir.



2.3 TESTÇJÜK

Selim SARPER

YAZILIM TESTİ GENELDE, üzerinde çalışacağı fiziksel sistemle beraber yapılmaktadır (eğer PC'de çalışan bir muhasebe programı değilse). Bu nedenle test kadrosunun daha önce tanımlandığı gibi Sistem seviyesinde bilgili olması gereklidir.

2.4 KULLANICIUK

F. Mehmet ÖZEL

PİYASAYA VERİLEN YAZILIMLARIN paketi üzerinde, kullanıcının hangi düzeyde bilgiye ihtiyacı olduğunu belirten işaretler konulması yararlıdır. Bu uygulama sadece Türkiye için değil dünya için de geçerlidir. (Otelin yıldız sayıları gibi bir derecelendirme olabilir) örnek olarak, oyunlar hemen hemen hiçbir bilgi gerektirmez. Buna mukabil bazı paketler hiç olmazsa işletim sistemini bilmeyi gerektirir, vb.

Erdal YALT

YAYGIN OLARAK BİLGİSAYAR kullanımını geçerken, önce kullanıcıların; eğitim seviyeleri, bilgisayara olan alışkanlıkları ve kullanım amaçları belirlenerek gereken eğitimin verilmesi, çeşitli yazılımlardan alınan verimi daha üst seviyelere çıkaracaktır.

Ticari amaçlarla hazırlanmış olan paket programlarda sürekli yapılan versbn değişiklikleri, temel eğitiminde noksanları bulunan kullanıcılar için sorunlar yaratmakta ve kullanılan paket programdan tam kapasite ile yararlanılmamasına neden olmaktadır.

3 YAZILIM SANAYİNİN DURUMU VE SORUNLARI

3.1 GÜNÜMÜZ BÜYÜK SANAYİ ŞİRKETLERİNİN YAPISI İÇİNDE YAZILIM ÜRETİMİ

O.Aydın AÇIKGÖZ

ÜLKEMİZDE YAZILIM SANAYİNİN kurulması ve geliştirilmesi için büyük sanayi şirketlerinin konuya ilgi göstermeleri ve gerekli yatırımı yapmaları, gelecekte sıkıntıya düşmemekte ve çağı yakalayabilmek için gerekli görünmektedir. Çünkü, bu şirketlerin üretimlerinin çeşitli aşamalarında bilgisayara, dolayısıyla yazılıma duyacakları gereksinimin gittikçe artacağını kestirmek hiç de güç değildir. Belirli standartlarda üretilen donanım dışalım yoluyla sağlanabilir. Buna karşın donanımı kullanıcının hizmetine sunacak yazılım araçlarının dışalım yoluyla sağlanabilmesi her zaman olanaklı değildir. Dışardan sağlanan yazılımın Türkiye'de etkin bir şekilde kullanılmamasının nedenlerinden biri dH farklılığıdır. Ayrıca, çeşitli üretim, otomasyon, kontrol, eğitim işlemlerinde uygulanan yol ve yöntem farklılıkları da dış ülkelerde üretilen yazılımları bizim kendi yapımıza uydurmamızı zorlaştırmaktadır.

Yazılım üretiminin endüstriyel bir üretim süreci olarak değerlendirilmesi ve devletin de bu üretimin kârlılığı için gerekli yasal düzenlemelere gitmesi gerekir. Bir başka deyişle, yazılımın da endüstriyel bir ürün olduğunun kabul edilmesi, herhangi bir endüstri ürünü gibi pazarlanabilmesi ve üreticilerin ürünleri üzerindeki yasal haklarının korunması gerekir. Küçük yazılım şirketlerinin devletin gerekli düzenlemeleri yapabilmesi için bir baskı grubu oluşturamadıkları görülmektedir. Bu konularda büyük sanayi şirketlerimizin öncülük etmesi devletin de bu alandaki sorunlara daha fazla ilgi göstermesini sağlayacaktır.

Sanayi kuruluşlarımızın oluşturacağı yazılım bölümlerinin hem kendi kullanımları hem de iç ve dış piyasalar için üretim yapabilmesi olanaklıdır. Bunun için çok büyük yatırımlar gerekmemektedir. Gittikçe ucuzlayan donanım ve yetişmiş insan gücü iyi kullanıldığında başarılı olunamaması için hiçbir neden yoktur.

Selami ATLI

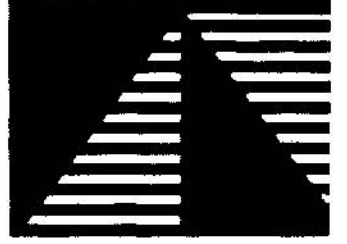
ASKERİ SİSTEMLERDE Yazılım Geliştirme ve Uygulamadaki Problemler

Yazılım geliştirme sürecinde yazılıma gereken önemin verilmemesi, yetersiz dokümantasyon, ihtiyaçların tanımlanamaması, standartların belirlenmemiş olması gibi problemler gözlenmektedir. Ancak hemen belirtilmesi gereken bir husus bu problemlerin sadece askeri yazılımlara özgü olmadığı, genelde yaşandığıdır.

Standartlara uygun tariflenmiş, tasarımılanmış, geliştirilmiş, test edilmiş, kalite güvencesi sağlanmış ve bütün bu süreçlerde dokümantasyonun tam olarak yapılmış olduğu bir sistemin maliyeti, idame masraflarının geliştirme masraflarından çok daha yüksek olduğu göz önüne alındığında, daha ucuza gelecektir. Ancak ihtiyaç sahibi makamların sadece geliştirme maliyetlerine göre karar vermesi sonucu idamesi mümkün olmayan ve sonuçta kullanılmayan sistemlere sahip olması ve bunun yol açabileceği kayıplar çok büyük miktarlara ulaşmaktadır.

ÖNERİLER

Türkiye'de askeri yazılım teknolojisi son yıllarda gelişmekte olmasına rağmen henüz olması gereken düzeye ulaşamamıştır. Silahlı Kuvvetlerin ihtiyaçlarının teknolojik gelişmelerin en büyük itici gücü olduğu daha önce de belirtilmiştir. Ancak bu ihtiyaçların, Silahlı Kuvvetler-Üniversite-Endüstri işbirliğiyle sağlanabileceği, batıdaki örneklerle kanıtlanmıştır. Nitekim ABD'de özellikle Ünh/erske-Endüstri işbirliğinin bu ülkeyi teknolojik olarak bugünkü konumuna getirdiği herkes tarafından kabul edilen bir gerçektir. Arzulanan düzeye gelebilmek için Devlet, TSK, Üniversite ve endüstri kuruluşlarınca alınması gereken önlemlerin en kısa zamanda gerçekleştirilmesi bu sürecin kısaltılması ve ulusal savunma sistemlerimizde bağımsız olmamız açısından gereklidir. Alınması gereken önlemleri sadece yazılıma yönelik olarak düşünmemek, askeri yazılımın askeri sistemlerin bir parçası olduğunu göz önüne alarak, bir bütünün içinde görmek gerekir.



Üniversitelerin güçlendirilmesi

Araştırma-Geliştirme programlarının belirlenmesi

Eğitim programlarında düzenlemeler

Üniversite-Endüstri işbirliğinin sağlanması

Ortak geliştirme programları

Kaynak israfını önlemek

Standartların belirlenmesi

İhtiyaçların tanımlanması

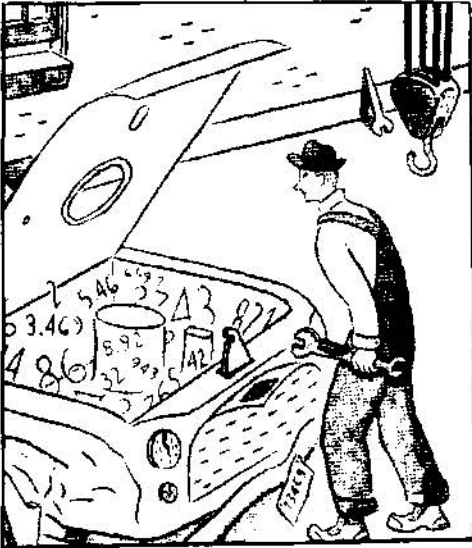
Teşvik tedbirlerinin en kısa zamanda belirlenip uygulanması

gibi önlemlerin bu çerçevede alınması gerekli görülmektedir.

Kemal İNAN

TÜRKİYE BAZI ÇEVRELERCE sık tekrar edildiği gibi yazılım sanayinde önemli bir kaynak yahut, avantaja sahip değildir. Bunun nedeni, konuyla ilgili eğitim kurumlarının (bilgisayar bölümlerinin) görece azlığı ve olan kararındaki pek çoğunun gerek öğretim kadroları gerek teçhizat açısından yetersizliğidir. Türkiye'nin eğer varsa, bir avantajı gelişkin sayılabilecek bir sayısal iletişim şebekesine sahip olması ve bu ve benzeri alanlarda üre-

tim yapan birkaç büyük şirketin birikimleridir. Bu avantajın uluslararası düzeyde kullanılabilmesi için ilgili eğitim kurumlarına yatırım yapılması ve sözkonusu şirketleri dış piyasalara itecek siyasi perspektifin ve kararlılığın gerçekleşmesidir. İç piyasaya yönelik yazılım sanayi de ancak böyle dış perspektifli bir altyapı üzerinde sıçrama yaparak yeşerebilir.



Fulya TURAN

GÜNÜMÜZDE "YAZIUM" ya da "Yazılım ile çalışan sistemler" üreten firmaların temel problemleri yazılımın yatırım ve bakım maliyetlerinin saptanması, kontrol altında tutulması ve optimize edilmesidir. Bu nedenle amaca uygun bir tasarım ortamı ve kontrol süreçleri gereklidir. Ancak bunlar tek başlarına yeterli değildir. Geliştirilen yazılımın mimarisi de kullanım amacı ile uygun olmalıdır.

Oğuz UNCU

ASELSAN İÇİNDE YAZILIM üretimi

ASELSAN, Banka Alarm Sistemi, Oto yol Ücret Toplama Sistemi, Oto yol Acil Yardım Sistemi gibi bilgisayar denetimli sistemler geliştirmiştir. Bu tür sistemlerin ortak özelliği sadece yazılımdan oluşmayıp (örneğin bir muhasabe yazılımı gibi), birçok donanım ile birlikte iç içe çalışmaları ve genellikle de birtakım uzak akıllı birimleri bir merkezden denetleme işlevine sahip olmalarıdır. Bu nedenle, geliştirilen yazılımın birçok dış donanım ya da yazılım elemanı ile arayüzü, yazılım geliştirme planlaması, yazılım entegrasyon ve testi gibi konularda sorunlar yaratmaktadır. Sözkonusu sorun yönetim yapısı matris olan ASELSAN içerisinde değişik boyutlar kazanmaktadır.

3.2 TEMSİLCİLİK YOLUYLA BİLGİSAYAR PAZARLAMASI YAPAN ŞİRKETLERDE YAZIUM ÜRETİMİ

M.Danis UNVERDİ

TEMSİLCİLİK YOLU İLE bilgisayar pazarlaması yapan şirketlerde yazılım üretiminin sözkonusu olması doğrusu beni şaşırttı. Ben bilgisayar ve yazılım pazarlaması ile yazılım üretiminin birbirinden çok farklı konular olduğunu, bu tip şirketlerin yazılım üretimi yapmalarının uzak bir ihtimal olduğunu düşünüyorum. Temsilciliğini yaptıkları yabancı şirketler, bunların gerçek anlamda ve kendilerine, Türkiye pazarlarıyla kısıtlı bile olsa, rakip olacak biçimde yazılım üretimi yapmalarına hiçbir zaman sıcak bakmayacaklar ve kendi açılarından haklı olarak engelleyeceklerdir, (ihtiyaç duyulan yazılımın hazır verilmesi gibi).

3.3 SALT YAZIUM ÜRETİMİ YAPAN KÜÇÜK VE ORTA ÖLÇEKÜ ŞİRKETLERİN OLUŞTURDUĞU YERÜ YAZIUM SANAYİNİN NEDEN BUGÜNE DEĞİN YETERLİ DÜZEYDE OLUŞAMADIĞI VE ÇÖZÜM ÖNERİLERİ

Şebnem ARIK

YAZIUM SANAYİNİN gelişebilmesi için konusunda uzmanlaşmış yazılımcılardan ya da araştırmacılardan oluşacak yazılım gruplarına ihtiyaç vardır. Bu gruplar değişik yazılım konularında araştırma yapılması, piyasayı ve bilimsel kuruluşlara destek verilmesi/ va%>ii yazılımlar geliştirilmesi üzerinde Mşarak yazılım sektörünün gelişmesine katkı bula-nabilirler. Özel sektörün ve üniversitelerin yardımıyla bu grupları bir çatı altında toplayan bir kurum (enstitü, dernek,...) kurulabilir

Mehmet BARAY

YAZILIM ÜRETİMİNDE veya geliştirilmesinde en önemli unsur yetişmiş insan gücüdür. Şu anda üniversite ve yüksekokullarımızdan yeterli sayıda eleman yetişmemektedir. Eğitimin daha yaygın olması, üniversite ve yüksek okullardan daha nitelikli elemanların yetişmesi, yazılım üretimi için en önemli konudur.

F. Mehmet ÖZEL

YERLİ YAZIUM SANAYİ bugününden çok daha ileri durumda olmalıydı. Çünkü genel şartlar böyle bir sonucu gerektiriyor. Eğitilmiş insan sayısı yeterli düzeyde, bilgisayar parkı da yeterli düzeyde. Şu halde neden başarı bize gelmemiş?

A) Üretilen yazılımlarda standart, norm ve kalite denetiminin genellikle olmayışı. Bu durum Türkiye'de tereddütlere yolaçmış ve güvensizlik yaratmıştır. Şimdi bunu düzeltmek çok zor olmaktadır. Oysa dünya ile birlikte hareket edilerek. "USER FRIENDLY" "ICONS" "BENCH TESTS" ve benzeri kavramlarla daha önceden tanışıp, uygulamaya geçilmeliydi. Bu konuda faz kaybedilmişdir.

B) Pazarın sanayicileri FEASIBLE bir düzeyde desteklememesi. Başka bir deyişle, pazar hacminin batıya göre daha yavaş gelişme-

si. Bu durum sanayicilerin ürünlerine daha fazla yatırım yapmalarını önlemiştir. Ve hatta global düzeye çıkmayı da engellemiştir. Bu konuda şunlar söylenebilir:

Potansiyel müşteriler, uzun süre yazılım alımında tereddüt göstermişlerdir. Bunda ciddi ürünlerin daha başlangıçta ortaya çıkmamasının payı vardır.

Büyük organizasyonlar, küçük yazılım kuruluşlarını desteklememişlerdir.

C) özellikle küçük yazılım üreticileri, pazarlamaya profesyonel önemi vermemişlerdir.

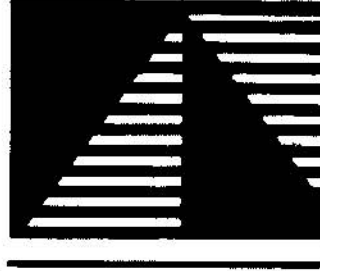
D) Dünyadaki diğer yazılım şirketleri ile teknik anlaşmalara girilmesi çok gecikmiştir.

E) Türkiye'deki kanunlar bu sanayicileri teşvik etmeye yetmemiştir. Kanunsuz çoğaltma konusu çözüme kavuştuğu takdirde, Türkiye'deki bu sanayi kolu çok daha hızlı gelişecektir.

M.Danis UNVERDI

BENİM GÖRÜŞÜM, yazılımın, tabiatı itibariyle küçük gruplar tarafından yapılması gerektiğidir; dolayısıyla ağırlık küçük ve orta çaplı şirketlerde olmalıdır (en çok birkaç yüz mühendisin çalıştığı şirketler). Başka bir deyişle, Türkiye'de büyük çaplı yazılım şirketlerinin olmaması bence endişe yaratacak bir durum değildir. Bu şirketlerde dinamizm önde gelen unsur olarak görülmelidir. Çalışanlar kendilerini günün belirli saatleri arasında iş gören insanlar olarak değil, hareket serbestisi ve bağımsız karar verebilme olanağı yüksek, sürekli çalışan kişiler olarak görmelidir. Yönetici kademelerinde bulunanlar, çalışanların net sonuçlarına göre değerlendirme yapmalı, yazılım konusunda, modası geçmiş tutumlar bırakılmalıdır. Çalışanlara tanınan bu kolaylıklar dolayısıyla artan dinamizmin bir sonucu olarak, görevini hakkıyla yapamayanların işten uzaklaştırılmaları, yerlerine yeni elemanlar alınması, işten uzaklaştırılanların başka şirketlerde kendilerine daha uygun işler bulmaları bir rutin olarak ortaya çıkacaktır.

Bu tutum, elette tek başına yerli yazılım sanayinin problemlerine çözüm olmaya yetmeyecektir. Bu konuda devletin de bazı katkıları beklenebilir ki, bunu 6. soruya verdiğim cevapta belirteceğim.



Oysa, yazılım
üretebilmek
için gerekli
olan en önemli
unsurlar
Türkiye'de
zaten
bulunmaktadır.
Bunlar, varolan
donanım
yatırımı,
yetiştirilmiş
eleman ve
beyin gücüdür.
Yani altyapı
hazır
durumdadır.

3.4 YAZILIM SANAYİNİN YURTDIŞINA
AÇILMASI
M. zafer DOKUZOĞLU

YAZILIM SEKTÖRÜ fazla bir yatırım yapmayı gerektirmeden ürün üretebilecek ve dışa açılabilir bir sektör olmasına rağmen Türk yazılım sektörü bu imkandan yeteri kadar faydalanmamaktadır.

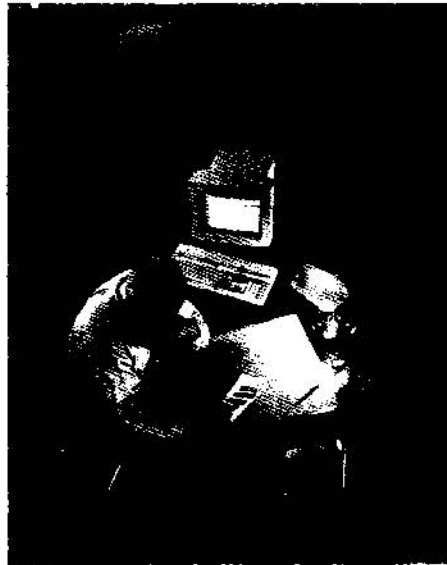
Hannover'deki CeBIT92 fuarında Türk firması olarak sadece iki firma katılmışken, Yunanistan'dan 13, Bağımsız Devletler Topluluğu'ndan 20, Bulgaristan'dan 5, Macaristan'dan 7 firmanın katılmış olduğu gözlenmiştir. Eski Ooğu bloğu ülke firmalarının da ürünlerini sergilemekte olduğu gözlenmiştir. Doğu bbku ülkelerinin örneğin "Optical Character Recognition" konusunda ilerlemiş oldukları gözlenmiştir.

Türk firmalarının kendilerine hedef platform seçmeleri (PC, Macintosh, VAX, IBM vb.) ve bu platformlarda geliştirebilecekleri ve yurtdışında rekabet edebilecekleri uygulamaları seçmeleri gerekmektedir.

Firmalar yazılım sektöründe başarılı olabileceklerini kanıtladıkları takdirde devletten de teşvik alacakları kesindir.

Haluk GÖKÇE

YAZILIM ÜRÜNÜNÜ diğer ticari ürünlerden ayırmak doğru olmaz. Bu nedenle yerli üreticisi korunan ürünler gibi yazılımın da yerli üreticisinin ithal yazılımlara karşı korunması gerekir. Nasıl otomobil ithalatı için koruyucu fonlar mevcut, aynı fonlar yazılım ürünleri için de sözkonusu olabilir. Yazılım endüstrimizin dış dünyayla rekabet edebilecek düzeye gelebilmesi için, geliştirilmesi planlanan tüm endüstri sektörlerimize uygulanan yöntemlerin aynısı bu günümüz ve geleceğin sektörüne de uygulanmalıdır. Dış açılma bu prosesin sonlarında ancak gündeme gelebilir.



M. Daniş UNVERDI

YAZILIM SANAYİNİN yurt dışına açılması konusuna gelince, bu çok arzulanmış gelişmenin ancak iç pazara hakim olunduktan sonra, gerçek boyutlarıyla gündeme gelebileceğini düşünüyorum.

3.5 BENZERİ BAŞKA KONULAR

M. Ounmil AZIZOĞLU

TÜRKİYE'DE YAZILIM ÜRETEN küçük ve orta ölçekli işletmelerin en büyük sorunu, devlet desteğinin, diğer sanayii dallarında olduğu kadar bu alanda görülmemesi, hatta öneminin ve var olan beyin gücünün tam olarak kavranmamış olmasıdır. Böylece donanım alanında olduğu gibi yazılım alanında da dışa bağımlılık sürmektedir.

Oysa, yazılım üretebilmek için gerekli olan en önemli unsurlar Türkiye'de zaten bulunmaktadır. Bunlar, varolan donanım yatırımı, yetiştirilmiş eleman ve beyin gücüdür. Yani altyapı hazır durumdadır. Sadece yapılacak bir miktar yazılım yatırımı ile (yazılım üretmek için gerekli tool'lar) ve KOSGEB gibi bir devlet kuruluşunun organizasyonu doğrultusunda Türkiye'de yazılım üretimi başlatılabilecektir.

Ayrıca, DEC, SUN, HP, IBM, MICROSOFT vb. gibi, dünya bilgisayar piyasasını elinde tutan büyük şirketlerden de destek alınabilir.

Kamil
ÖZÇIKMAK

YAZILIM ÜRETİMİ klasik anlamda üretilen mal, hizmet üretimi kavramlarından biraz değişiklik göstermekle birlikte, yazılım üretimi diğer klasik mal ve hizmet üretiminde olduğu gibi üreten ve tüketen olarak iki önemli ilgi grubunu içerir. Bu yazım-

da yazılım tüketicisi olarak yazılım sanayinin bugünkü durumunu irdeleyeceğim.

Yazılım tüketicileri iki ana grupta ele alınabilir:

1- Bir işi yapmak için ne istediğini iyi bilen yazılım şartnamesi yazabilecek bu şartnameye uygunluğunu kontrol edebilecek bilgi birikimi olan tüketiciler.

2- Gereksinimi olduğu yazılımı tam olarak tanımlayamayan, yeterli bilgi birikimi olmayan bu nedenle benzeri sorunları olan yakınındaki tüketicilerin izlediği yolu veya yazılımı satın alan tüketici grubu.

2. tip tüketicilerin aldıkları yazılımları genel amaçlı olduğu için bu yazılımları özel amaçları için geliştirmek, ayarlamalar yapmak durumunda kalmaktadır. Bu tür tüketiciler için yazılım hakkında yeterli doküman ve eğitim desteği sağlanmalıdır. Bunlar ilk başta maliyeti artıran unsurlar olmakla birlikte, orta ve uzun vadede pazarı büyüteceği için yazılım üreticilerinin yararına olacaktır.

Yazılım tüketicileri açısından bir başka önemli husus yazılımların taşınabilirliği ve diğer yazılımlarla iletişim kurabilme özelliğinin bulunmasıdır. Bu durum yazılım üreticilerinin açık sistem OSI ilkelerini benimsemesi ile olacaktır.

Sonuç olarak yazılım üreticileri;

1) Yazılım tüketicilerinin yaptığı yatırımın efektif olabilmesi için tüketiciyi koruma, bilgilendirme yükümlülükleri vardır. Ürettikleri ve/veya sattıkları yazılım ürününe yeterli teknik, bilgi, doküman ve eğitim desteğini sürdürmelidirler. Bu nokta en az mükemmelliği kadar önemlidir.

2) Yazılım üreticileri ürettikleri yazılımı güvenilirlik, kullanılabilirlik ve açık sistemlere uygunluğu bakımından test yöntemleri geliştirip bunları yazılım tüketicilerine bildirmelidirler.

3) Üretilen yazılımların işletim sistemlerinden olabildiğince bağımsız taşınabilir olmasına özen göstermelidirler.

F. Mehmet ÖZEL

YURTDIŞINDA YAZILIM PAKETİ üreten firmaların Türkiye'de temsilciliğini yapan pazarlama şirketleri çok garip bir durumdadır. Türkiye'de kopya yapılması yeterince

cezalandırılmamaktadır. Böyle şirketler, fedakarca ürünleri Türkiye'ye ithal etmekte, tanıtmakta, fakat kanunsuz çoğaltma yüzünden çabalarının sonuçlarını alamamaktadırlar. Bu düzenlemelerin çabuk ve etkili bir şekilde yeniden ele alınması gerekmektedir. Bu, bizim medeni dünyaya kabulümüz için ayrıca bir gerekliliktir. Bu alan sadece fikir ve sanat eserleri başlığı altında ele alınmamalıdır. Yazılım başlı başına bir olgudur. Ve kendine özgü düzenlemeler gerektirmektedir.

Selim SARPER

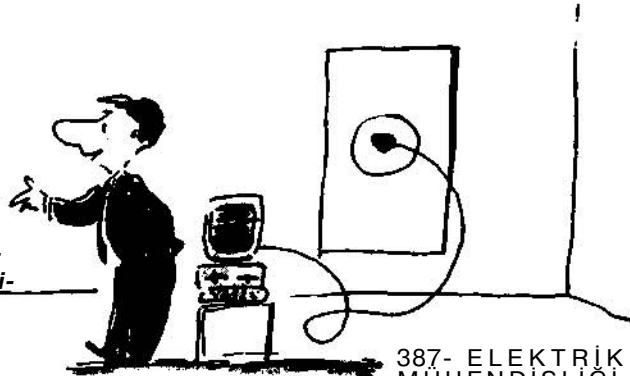
TÜRK YAZILIM SANAYİNİN (yazılımı Sistem'den ayrı tutmak biraz güç) arzu edilen yere gelebilmesi için aşağıdaki çözümleri önereceğim. Bu çözümlerin hepsinin aynı anda (paralel) uygulanmasında yarar vardır.

a) Üniversite-Lise-Ortaokul (?) aşamalarında itibaren, öğrencilere "System" kavramının ve yazılımın gerektirdiği mantık kurallarının ve özgüvenin aşılanması,

b) Türkiye'nin yapmak zorunda olduğu atılımın devletin resmi bir politikası haline getirilmesi, maddi ve manevi açıdan desteklenmesi,

c) Şirketlerimizin özgün tasarımlarda bulunmaya özendirilmeleri. Ancak burada sınırın nerede olacağına çok iyi karar vermek gereklidir. Tekerleği yeniden keşfetmek yerine mevcudu iyi anlayıp daha iyisini hedeflemeliyiz.

d) "Amaca uygun" hazırlanmış lisans anlaşmaları ile , ilerlemiş ülkelerin "know-how"larının Türkiye'ye kazandırılması.



387- ELEKTRİK MÜHENDİSLİĞİ

155

Ticari

uygulama

yazılım

sektöründe yer

almaya çalışan

bir firmanın

yaşadığı ana

darboğazlardan

birisi, bilgi

biriktirme

mühendisliği

çalışmasının altı

ay ile iki yıl

arasında

değişen bir

zaman süresi

gerektirmesidir.

Osman SEVAIOĞLU

YERLİ YAZIUM endüstrisinin bugünkü durumu ve sorunları.

Türkiye'de yazılım geliştirmeli miyiz?

Bu sorunun yanıtı büyük oranda ticaridir. Yazılım nerede kullanılacaktır, neler beklenmektedir, kullanılması için ne kadar acele edilmektedir, kaç defa kullanılacaktır, yazılımın kapsamı nedir, prototipinin hazırlanması için kaç adam-aylık bir çalışma gerekmektedir, yurtdışında geliştirilmiş rakipleri var mıdır, bu rakipler amaca ne kadar uymaktadır ve fiyatları nedir? Bu sorulara verilecek cevaplar, yazılım geliştirmenin oluruğunu saptayacaktır. Belli bir alan için olumsuz çıkan bu olurluk analizi bir başka alanda olumlu çıkabilir. Bu anlamda, konu bir arz ve talep meselesidir.

Ticari Uygulama Yazılımları

Ticari uygulama yazılım sektöründe yazılım üreten ve pazarlayabilen firmaların sayısı oldukça azdır. Bu firmalardan sektöre hakim durumda olan bellibaşlı olanlarının bile yaklaşık olarak beş veya yedi yıllık bir geçmişleri vardır. Personel kadroları genellikle on ile otuz arasında değişmektedir. Nadir durumlarda yüze kadar ulaşabilmektedir. Aralarında yoğun bir rekabet ortamı vardır. Hemen hepsi korsanlığa karşı, anahtar diokot, cari çıkışlara takılan lan dn-kilit benzeri birtakım tedbirlerle kendilerini korumaktadır.

Ticari uygulama yazılım sektörünün bugünkü yapısı, teknik düzeyi oldukça yüksek personele gereksinim duyması nedeniyle, istihdam kapasitesini arttırmaya pek uygun görünmemektedir. İçinde bulunulan rekabet ortamı, fiyatların belli bir tavanın üstüne çıkamaması, müşterinin genellikle bir makinasının olmaması ve yazılımı kullanabilmek için önce

bu makinaryı satın almak durumunda olması, müşterinin sürekli, anında, ve mutlaka kaliteli servis hizmeti beklemesi istihdam sorunlarının kaynağını oluşturmaktadır. Müşteri firma ilişkilerinde genellikle donanım servis personeli eksikliği yoğun olarak hissedilmektedir, fakat bu gruptaki servis personeli donanım firmaları tarafından istihdam edilmektedir.

Bu noktada donanım ve yazılım firmaları arasındaki ilişkilerden de kısaca bahsedilebilir. Donanım satışı ile uğraşan firmaların hemen hepsi bayilik sistemi ile belli başlı ticari yazılım firmalarının ürünlerini pazarlamaktadırlar, yani bu anlamda, müşteri ile satış anında ilk ilişkiyi kuran bir yazılım firması değil, bir donanım firmasıdır. Buna karşılık, en belli başlı donanım firmalarının bile sattıkları yazılımın değil kullanılması, fakat kurulması, yani,

kullanıma hazır hale getirilmesi donanım firması elindeki kopyalanabilir türdeki kelime işlem, veri tabanı gibi birtakım paket yazılımları satılan makinarya kurarak teslim etmektedir. Böyle bir ortamda yeni kurulan küçük boyutlu bir yazılım firmasının birkaç kişilik personel kadrosuyla bile olsa ayakta durabilmesi ve ürünlerini pazarlayabilmesi oldukça zor görünmektedir.

Mühendislik Uygulama Yazılımları

Özelliklerinden dolayı bu sektörü ticari uygulama yazılım sektöründen ayrı olarak düşünmek belki daha yararlı olacaktır.

Mühendisim uygulama yazılımlarında genellikle inşaat sektöründe ümit verici gelişmeler görülmektedir. Bu sektörde, çok detaylı ve geniş kapasitede dış kaynaklı yazılım potansiyeli vardır. Bu yazılımlar genellikle belli donanım firmaları tarafından ithal edilerek pazarlanmaktadır. Bu konuda da satış sonrası teknik hizmetlerin yeterli olarak verilip verilmediği bir sorudur. Yazılımın ithal kaynaklı olması, yerli yazılım sektörüne bir katkı mı yoksa engelleme mi getirdiği sorusunu gündeme getirmektedir.

Mühendislik uygulama yazılımlarında bilgi



biriktirme mühendisliği için harcanan zaman süresi, ticari uygulama yazılımlarına göre çok daha fazla olabilmektedir. On yıllık bir çalışma sonunda hala tam olarak geliştirilememiş mühendistik yazılım örnekleri az değildir. Bu noktada yazılım geliştirme durumunda olan firmanın ne kadar süre ile ve kim tarafından ve neler beklenilerek destekleneceği cevaplanması gereken önemli bir sorudur. Bugünkü durumda, başarılı olan birkaç firma geçmişte ya ana uğraşı alanı dışında elde ettiği ticari bir rant ile bu işe girmiş, ya da üniversite devlet kuruluşları gibi birtakım dış kuruluşlarda yazılımın ana çekirdeğini oluşturduktan sonra ticari piyasaya açılabilmişlerdir. Sonuç olarak, bu firmaların çoğunun bilgi biriktirme mühendisliği süresini müşterilerine fatura edebildikleri söylenemez.

Mühendistik uygulama yazılımlarında gerektirdiği bilgi birikimi adam-ay hacminden dolayı maliyetler oldukça yüksektir. Kullanılması gereken makina genellikle, ticari uygulama yazılımlarındaki kullanılanlara göre daha iyi teknik özelliklere sahiptir ve daha pahalıdır. Bu sektörde de yazılım fiyatları konusunda, özellikle yazılım ithalatı nedeniyle bir üst limit vardır.

Donanım firmalarının mühendisi* uygulama yazılımları konusunda ticari uygulama yazılımlarına göre çok daha fazla duyarsız davranırdıktan söylenebilir. Bu duyarsızlık geniş ölçüde, sektörde kullanılan yazılımların genellikle yurt dışı kaynaklı olmasından, buna karşılık yurt içindeki talebin de oldukça dar olmasından kaynaklanmaktadır. Konunun mühendislik bilgi biricimi gerektirmesi, donanım firmalarının konudan uzak olmaları, bu ölçülerde detaylı teknik bir yazılım için müşteriye karşı sorumlu olmaktan kaçınmak istemeleri bu sektörde çatışmak isteyen yazılım firmalarının ana darboğazını oluşturmaktadır.

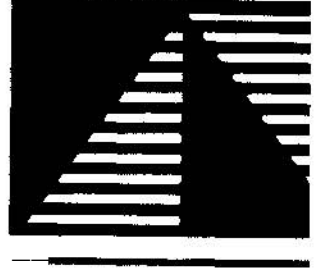
Mühendislik uygulama yazılımı sektöründe ayakta durabilen firmaların istihdam ettikleri personel sayısının ticari uygulama yazılım firmalarına göre çok daha az olduğu görülmektedir. Böyle firmaların en konusunda bile yeterli yazılım personeli ve bilgi birikimine sahip oldukları söylenemez. Bir problem çıkması durumunda, bu konuda sunulabilen servis, müşteri ile bayilik yapılan yazılım firması arasında bir ilişki kurulmasının sağlanmasından ibarettir. Donanım firmasının ana çalışma alanının, temelde donanım ürünleri satmak ve

bunların servisini sağlamak olduğu düşünülürse, yazılım hizmetleri konusundaki zayıflıklarını bir noktada beki anlayışla karşılamak mümkündür, fakat sonuçta müşterinin yazılım donanım birlikteliğinden, yani ortak çalışmasından, donanımının yazılma uyumundan genellikle pek mutlu olmadığı, istek ve şikayetlerine yeteri etkinlikte ve hızda bir yanıt alamadığı, satış sonrasında sahipsiz bir şekilde ortada kaldığı da bir gerçektir.

Ticari uygulama yazılım sektöründe yaşanan bir diğer sorun da konunun uygulama alanında mesleki deneyim gerektirmesi, buna karşılık yazılım geliştirme durumunda olan personelin bu deneyimden genellikle yoksun olmasıdır. Başarılı bir uygulama, muhasebe, bankacılık, belediyedir, finansman, vergi mevzuatı, stok ve personel yönetimi, hukuk gibi alt sektörlerden birinde uzmanlaşılmasını gerektirmektedir. Meslekten bir uzman ile bir yazılım geliştirme personeli arasındaki bilgi iletişimi kullanılan yöntem genellikle, "Knowledge Engineering" olarak tanımlanan bilgi biriktirme mühendisliğidir. Bu nedenle, sektörde hem uygulama alanında hem de bu uygulamayı profesyonel düzeyde bir yazılıma aktarabilecek düzeyde kadrolara gerek sinim vardır. Bu tür bir yerli işgücü kaynağının ise zengin olduğunu söylemek zordur.

Bu aşamada yaşanan temel zoriuk, uygulama alanında deneyim sahibi olan uzmanın genellikle donanım veya yazılım sektöründen uzak olmasından kaynaklanmaktadır. Yani, ikinci tür personelin yazılım geliştirme çalışmalarına katılabilmeleri genetikle mümkün olmamaktadır. Bu nedenle, uygulamalardaki ana itici güç, genetikle konuya girmek için ticari gayret ve irade gösteren yazılım firmalarından oluşmaktadır.

Ticari uygulama yazılım sektöründe yer almaya çalışan bir firmanın yaşadığı ana darboğazlardan birisi, bilgi biriktirme mühendisliği çalışmasının akı ay ile ki yıl arasında değişen bir zaman süresi gerektirmesidir. Bu süre, yeni kurulan bir yazılım firması için oldukça uzun bir süredir ve başlangıç yatırımını yapmak, herhangi bir karşılık almadan çalışmak ve personel istihdam etmek anlamındadır. Bu darboğazı günümüze kadar sadece bellibaşlı birkaç firma aşabilmiştir. Bu firmaların şu andaki iş



hacimleri, personel sayıları, yazılımın güvenirliliği, kullanım el-kitaplarının düzeyi, satış sonrası sunulan servis hizmetlerinin düzeyi giderek yükselmektedir. Fakat tekrarlamakta fayda vardır ki böyle firmalar sayılıdır ve yeni kurulan küçük çaplı yazılım geliştirme firmalarının bu firmalar karşısındaki rekabet şansları oldukça azdır. Bu durumda küçük çaplı firmalar da daha çok danışmanlık statüsü içinde, daha ufak ve kısa süreli bilgi birikimi mühendisliği gerektiren küçük ve lokal uygulamalara girmektedirler.

Makina Hatlarının uygulanan ticari yazılım fiyatının çok üstünde olması da sektörün bir başka darboğazını oluşturmaktadır. Bir müşteri, satın aldığı bir makinanın değerinin genellikle dörtte biri ile onda biri arasında değişen bir bedel karşılığında istediği bir yazılımı alabilmektedir. Çoğu durumda, bu yazılım donanım firması tarafından donanımın teslimi anında makina ile birlikte verilmektedir. Yurtdışı kaynaklı bir yazılım kullanılması durumunda ise bu bedelin bile ödenmesine gerek kalmamakta, bellibaşlılarda bile direkt olarak yazılım geliştirme ile uğraşan personel sayısı bir veya ikiyi geçmemektedir. Bu sektörde gereksinim duyulan yazılım geliştirme personeli genellikle sonradan yazılım konusuna girmiş uzman mühendislerdir. Ticari uygulama yazılım sektöründe pek görülmeyen bu durum, bu sektörde bilgi biriktirme mühendisliğinin çok daha zor olmasından, fazla zaman ve işgücü gerektirmesinden kaynaklanmaktadır. Bu sektördeki firmalar, gerek varlıklarını devam ettirebilmek ve gerekse uyguladıkları yazılımların daha üst düzeyde makinalar gerektirmesi nedeniyle, makina ve kart düzeyinde donanım satışına da girmek zorunda kalmışlardır.

Neler Yapılabilir?

Yerli yazılım sektörünün gelişmesinin sağlanabilmesi için neler yapılabilir? Bu soruya devlet kuruluşları ve firmalar açısından iki yönden bakılabilir.

Devlet desteği olarak yazılım geliştirilen kuruluşlara kredi verilmesinden çok, destek yazılımlar sağlanması daha gerçekçi bir yaklaşım olarak görülmektedir. Burada destek yazılımlar, firmanın yazılım geliştirme çalışmalarında gereksinim duyduğu,

derleyici, editör, bellek yönetici, bellek artırıcı, veri tabanı yönetimi, işletim sistemi, gibi temel yazılımlardır ve hemen hemen tamamı yurtdışı kaynaklıdır. Destek verecek devlet kuruluşu bu konuda bir destek yazılım arşivi oluşturmaktadır.

Yerli yazılım teriminden ne anlaşılması gerektiği açıklığa kavuşturulmalıdır. Bu konuda ana çizgi yerli yazılım kullanıcılarından alınacak geribesleme, yani yansımadır. Müşteri kimdir? Ne kadar bedel ödemeye hazırdır gibi sorular sektör bazında ve anketlerle incelenmelidir.

Yazılım ithalatından korkulmamalıdır. Özellikle devlet tarafından yapılacak tüm yazılım yatırımlarında mutlaka bir olurluk analizi yapılmalı, yatırımın hangi uygulama sektörü için yapıldığı, finansmanın ne kadar bir süre içinde ve ne şekilde geri döneceği saptanmaya çalışılmalıdır.

Uzun zaman gerektiren, özellikle mühendislik uygulama yazılımı geliştirme projeleri için öncelikle bir pazar araştırması yapılmalı ve aynı konudaki yurtdışı kaynaklı yazılımların fiyatları, yurtiçinde uyarlanabilene özellikleri incelenmelidir.

Yazılım telif hakları yasası mutlaka çıkarılmalıdır. Yazılım geliştiren bir kimse, bir anlamda bir sanat eseri ile aynı emek yoğunlukta eser üreten bir çalışandır. Bu insanın emeği mutlaka koruma altına alınmalıdır. Bir bilgisayardan bir yazılımı, başka bir kuruluştaki bir başka bilgisayara kopyalamanın bir arabayı çalmak kadar suç olduğu bilinci yerleştirilmelidir.

Mukaddes ŞEKERCİ

İHTİYAÇ DUYULAN BİR yazılımın nasıl bulunacağı, hangi şirketlerin bu konuyla ilgili çalıştıkları, ihtiyaçların nasıl belirleneceği konularında danışmanlık yapacak bağımsız kuruluşların olmaması kullanıcılar açısından çok büyük zorluklar oluşturmaktadır.

Yazılım şirketlerinin sayısı her geçen gün artmaktadır. Hangi konularda faaliyet gösterdiklerini takip etmek mümkün değildir. Bu konuda yazılım firmalarının katılımıyla yapılacak bir çalışma ile kullanıcılara yol gösterecek bir yapının oluşturulması gerekmektedir.



4 YAZILIM SANAYİNİN ULUSAL EKONOMİDEKİ YERİ

4.4 STANDARTLAR

Suat BAYSAN

TÜRKİYE BİLİŞİM ÇAĞINI yakalamak zordur. Bu nedenle yazılımda sanayileşme ve standardizasyon kaçınılmazdır. Gerek ithal gerek Türkiye'de üretilen ürünlerde dünya standartlarıyla uyumlu olma koşulu ve standart ürün ve arabirimlerle ulaşılan dünyaya pencereleri açılan geniş amaçlı veri bankalarının oluşumu ürünlere dünya pazarında yarışabilecek bir kalite getirecektir. Ancak bu şekilde Türkiye'de dünya pazarıyla uyumlu bir yazılım sanayi oluşur.

Mukaddes ŞEKERCİ

BU KONUDAKİ en önemli eksikliklerden biri, yazılım ürünleri standartlarının ortaya konulmaması, beklenmemesidir. Bu durum yazılım ürünlerinin kalite kontrolü ile yakından ilgilidir. Standartlar olmadığı için, yazılım kalitesini ölçmek mümkün olmamaktadır.

Türkiye'de yazılım endüstrisini geliştirmek, hatasız yazılımlar üretmek ve dünya yazılım sektöründe rekabet edebilmek için en kısa zamanda bu konu ele alınmalıdır.

4.5 TELİF HAKLARI

M. Zafer
DOKUZOĞLU

TELİF HAKLARINA uyulması konusunda yurdumuzda gerekli özen gösterilmemektedir. Örneğin senelerce müzisyenler bu konudan yakınmışlar ve kısmen de olsa bandrol uygulaması ile bu konuya çözüm bulmuşlardır. Yazılımda bandrol uygulanması gibi konular bürokrasiyi artıracak verimi düşürecek uygulamalardır. Fakat anlaşılacak odur ki insanlarımızı telif haklarına saygılı olacak şekilde eğitinceye dek bu veya benzer uygulamaların yazılım sektöründe de uygulanması gerekebilecektir.

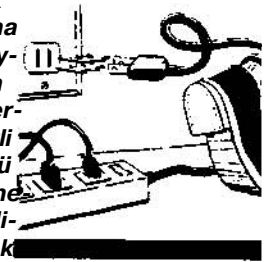
Yazılımın kopyalanması sorunu ülkemizde yazılım sektörünün gelişmesine engel olan en büyük sorundur. Bu sorun çözüme kavuşturulmadıkça yazılım sektörü gelişmeyecektir.

Eyüp ERGEÇEN

BİLGİSA YAR PROGRAMLARI, Fikir ve Sanat Eserleri Kanunu çerçevesinde, net ve spesifik olarak korunmamaktadır. Diğer fikir ve sanat eserlerine göre, Bilgisayar programları esas niteliği olarak bazı farklılıkları içermektedir. Bunlara örnek olarak; bir bilgisayar programını edinen kimsenin o programı kullanmak için çoğaltmak zorunda olması veya bilgisayar programlarının kullanıcının ihtiyaçlarına göre özelleştirilmesinin dolayısıyla eser üzerinde değişiklik yapmaya müsadde edilmesi gibi aslında diğer fikir ve sanat eserlerinde olmayan kimi farklı nitelikler söylenebilir. Edindiğimiz bir kitabı kullanmak için çoğaltmamız veya üzerinde değişiklik yapmamız gerekmez. Bu türde özelliklerin kanunda yer alması ile bilgisayar programlarının telif hakları korunmuş olabilir.

Ülkemizde yazılım korsanlığının düzeyi yabancı yazılımlarda %90, oyun yazılımlarında %100, yerli üretilen yazılımlarda ise %30 civarındadır. Korsan kullanım düzeyinin yüksekliği yazılım sektörüncü yapılacak yatırımları ve ülkemizde daha çok yazılım üretilmesini engellemektedir. Ülkede yazılım sektörünün sağlıklı bir şekilde varolabilmesi, gelişmesi temel olarak bilgisayar programlarının telif hakları ile hem kanunda hem de uygulamada sıkı koruma altına alınması ile başlar. Daha sonraki tedbirlerden biri ise sektörün devlet teşvikleri ile geliştirilmesidir. Diğer ülkelerde görüldüğü üzere, kanunla korunuyor olsa bile korsanlık düzeyi sifira indirilemez, genellikle %10 düzeyinde korsanlık kanuna rağmen var olacaktır. Bu ise yazılım kullanıcılarının eğitim ve bilinç düzeyi ile ilgilidir. Kullanıcıların çeşitli tekniklerle ve kampanyalar ile eğitilmesi, bu %10'u daha aşağılara çeker.

Yabancı orijinli yazılımların kopyalarının ülkemizde bir bedel ödemeden yaygın olarak kullanılıyor olması beki ilk bakışta, yurt dışına döviz kaybı olarak görülebilir, ancak daha ayrıntılı düşünüldüğünde yabancı yazılımların bedelsiz olarak piyasada var oluşu, yerli alternatiflerine karşı bir üstünlük sağlayacak yerli yazılım geliştirilmesini dolayısıyla bu sektörü baltalar. Bu fikri güçlendirmek için şu söylenebilir, yazılım kullanıcıları para ödemeden edinebilecekleri yabancı yazılımı, para ödeyerek



Bilgisayar
olayını
henüz
sindireme-
diğimizi,
hala gelişme
İçerisinde
olduğumu-
zu da
görüyoruz
ve
yaşadığımız
gerçekler de
ortada.
Kişiler
kullanacak-
ları her
yazılım
ürünü için
satın alma
yoluna
gidecek
olsalar
bunun
altından
kalkamazlar,
dolayısıyla
bilgisayar ve
getirdikleri
de bu ölçüde
geniş bir
kesime
yayılmaz.

edinecekleri yerli yazılıma tercih ederler, bu yazılım kullanıcısı açısından çok ekonomik bir gerçektir. Yazılım korsanlığının diğer bir etkisi ise, kullanıcıları etkileyen olumsuzluklardır. Korsanlık nedeniyle yeterli adetlerde yazılım satamayan firmalar mali nedenlerle yeterli düzeyde teknik destek ve eğitim verecek örgütlenmeye gidememektedir. Bir yazılımı yasal yollardan edinmiş olsanız, yeterli teknik destek ve eğitim veya diğer satış sonrası hizmetleri alamama riskiniz bulunmaktadır. Son analizde korsanlığın önlenmesi kullanıcıların da işine yarayacaktır.

Yazılım işinin çekici ve kârlı bir iş haline gelmesi, kullanıma sunulan yazılımlarda ürün çeşitliliğini artıracak bu ise kullanıcıların edinebilecekleri yazılım opsiyonlarını çoğaltacaktır.

Yazılım ürünlerinin telif hakları ile uluslararası standartta korunması umudu ile...

Haluk GÖKÇE

DIŞARIYA YILDA MİLYONLARCA dolar yazılım ücreti ödediğimiz düşünülürse, gerekli insan kaynağımız olan yurdumuzda daha aktif politikalarla bu akışı azaltıp tersine bile çevirebiliriz. Bunun için atılacak adımlardan ilki, belki de en önemlisi insanlarımızın yazılıma para vermeyi öğrenmeleri. Bunun içinde daha sağlıklı ve kontrollü telif hakları yasa ve uygulamaları gerekli. Yazacağı yazılımın satın alınmayacağını, kopyalanıp dağılıcağını düşünen hiçbir yazılımcı yurdumuzda program yazmayacaktır. Yurdumuzda daha emekleme aşamasında olan bu sektör, geç olmadan standartlaşması sağlanmalı, belki de bakanlık seviyesinde olmasada daha alt seviyelerde devlet kademesinde üzerine eğilmelidir.

4.7. BENZERİ BAŞKACA KONULAR

Ferhat ARIKAN

YAZILIMIN EKONOMİMİZ için gerekliliği

Bilgisayarlar daha güçlü duruma gelip fiyatları da tersine daha da düştükçe, ürün tasarımından silah sistemlerini kontrole kadar bir çok alanda kullanılmaya başlanışlardır. Fa-



kat, bu işleri asıl mümkün kılan yazılımdır ve önemi de gittikçe artmaktadır, örneğin, 1950'lerde Amerikan Savunma Bölümü (US DoD), yeni sistemler edinmek için hazırladığı bütçenin %80'ini donanıma ayırırken, bugün bu oranı yazılım için kullanmaktadır. Artık DoD senede 10 milyar doları yazılım için harcıyor duruma gelmiştir (1). Öte yandan, eskiden Japonya'da yazılım kodu yazmak düşük prestijli bir işken, bugün çoğu Japon şirketleri (örneğin, Sony) 21'inci yüzyılı hedef alan önemli yazılım geliştirme programları başlatmışlardır (2).

örneklerden anlaşılacağı gibi, yazılım gelişmiş Ökelerde önemli bir sektör konumuna gelmiştir. Bu gelişmeye ve var olan rekabete bağlı olarak şirketler, yabancı ülkelerden eleman transfer etmeye ya da yazılım işini düşük ücretle yine bu dış üyelerdeki (örneğin Rusya ve Çin'deki) insanlara yaptırmaktadırlar. Bu iş için fazla bir yatırım yapmaları da gerekmemektedir. Böylece, ürünlerini daha ucuza mal edip, pazar şanslarını da artırmaktadırlar. Aynı zamanda, işi yapanlar da ülkelerinde eskiye oranla daha çok kazanmaktadırlar.

Eğer ekonomik olarak güçlenmek, gelişmiş ülkelerle hiç olmazsa bazı alanlarda rekabet edebilmek ve bunları da bir an önce gerçekleştirmek istiyorsak, yazılım bu amaçlar için iyi bir seçimdir. Çünkü sanayinin çoğu dalı dışa bağımlı büyük yatırımlar gerektirmez. Birkaç bilgisayar ve insanın olduğu bir büroda dahi önemli yazılımlar gerçekleştirilebilir. Burada, bizim asıl üzerinde durmamız gereken faktör, insandır. İşin donanım kısmıyla da iklilenilebilir. Fakat, bugün dünyanın batısından doğusuna çoğu ülkesinde, birçok firma tarafından, hem de çok ucuza bilgisayar donanımı üretilmektedir. Eğer gelecekte donanım konusunda bir standart, uyum sağlanırsa (ki bugün bu bazı türlerde vardır), benzer ürünü daha ucuza mal edemedikten sonra üretmenin pek bir anlamı yoktur. Yine de her olasılığı düşünerek bu konuda da bir bilgi birikimimiz olmalıdır.

En önemli faktör olan insan konusunda ise hiç de azımsanmayacak bir potansiyele sahibiz. Yapılması gereken yazılım konusunda bu insanların üniversiteden de önce yönlendirilmesi ve eğitilmesidir. Ayrıca, gereken araçlar asıl değerlerinden çok daha ucuza bu insanlara verilmelidir. Nasıl turizmle uğraşanlara arazi tahsisi (ya da kiralınması), ucuz kredi verilmesi gibi işler yapılıyorsa, benzer kolaylıkların yazılım sektöründe çalışanlara da gösterilmesi gerekir. Bu sayede, iç piyasaya yönelik yazılım üretirken, dış piyasaya da ilk önce buradaki isim yapmış şirketler adına yazılım üretmek açılabiliriz (tabii direkt olarak da kendi ürünlerimizi satabiliriz). Yapılan bilgi alışverişi ve kazanılan tecrübeden sonra, biz de dünya yazılım piyasasında söz sahibi olabiliriz.

Bu arada, ulusal savunmamız için gerekli askeri sistemin etkinliği artık yazılıma da çok bağlıdır. Bunu da yabancılar yerine bizim yazmamız zorunludur. Bu da yazılıma neden önem vermemiz gerektiğini gösteren bir başka konudur.

Herkes üzerine düşeni yaparsa, kısa sürede yatırımı az, getirişi yüksek bir başka bacasız sanayiye daha sahip olmamamız için hiçbir neden yoktur. Böylece ekonomimize büyük bir katkı da gerçekleşecektir.

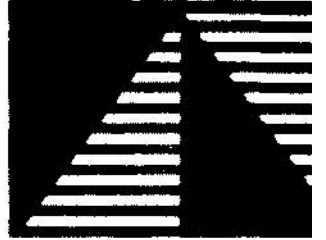
Kaynaklar:

- 1)EDN, 17 Ağustos 1989.
- 2) Fortune, 17 Haziran 1991

HER ÜRÜNDE OLDUĞU GİBİ, oluşturulan yazılımlarda da verilen emek ve harcanan çabanın karşılığının olması ve yerine ulaşması gerektiği inancındayım. Ancak bu konuya gereken önemi vermediğimiz de bir gerçek.

Diğer yandan, bıkışiyar olayını henüz sindiremediğimizi, hala gelişme içerisinde olduğumuzu da görüyoruz ve yaşadığımız gerçekler de ortada. Kişiler kullanacakları her yazılım ürünü için satın alma yoluna gidecek olsalar bunun altından kalkamazlar, dolayısıyla bilgisayar ve getirdikleri de bu ölçüde geniş bir kesime yayılamaz. Bir de gelişimin bir parçası olan üniversiteleri düşünelim. Yazılım sanayiinde yer alacak insanlar temel eğitimlerini bu kurumlarda alıyor. Mal yetersizlik içinde olan üniversiteler ist bu eğitim için gerekli ortamı yaratmakta şu anda bile kaynak sıkıntısı çekiyorlar.

Çok fazla sermaye istemeyen ve gerektirdiği sermayenin de karşılığını kat kat çıkaran bu sektörde bedellendirme daha gerçekçi olarak yapılmalı, özellikle derleyiciler gibi yazılım geliştirmeyi destekleyecek türdeki ürünlerde bazı kolaylıklar sağlanmalı, üniversiteler için özel çözümler düşünülmeli ve pek tabii ki bunların yanında yazılım sahibinin de hakları mutlaka korunmalı.



5 YAZILIMIN YARINKİ GÖRÜNÜMÜ

5.1. BDYM ARAÇLARI

(BİLGİSAYAR DESTEKLİ YAZILIM MÜHENDİSLİĞİ)

Gül ERGÜL

CASE (Computer Aided Software Engineering) araçları bir bilgi işlem merkezinin var oluş nedeni olan uygulamaların geliştirilmesi ve bakımının yapılmasında üretkenliği arttırmaktadır. Bu nedenle ekonomik önemleri büyüktür. Bir uygulama geliştirme projesinin tasarım aşamasından test aşamasına kadar geçirdiği tüm evrelerde kullanılabilen çeşitli CASE yazılımları bulunmaktadır. Bunun yanı sıra CASE araçları, var olan uygulamaların iyileştirilmesinde de kullanılabilirler. Genel olarak bakıldığında CASE araçları uygulama geliştirme sürecinde geleneksel programcı işlevlerinin bir kısmını yüklenmekte ve giderek özel eğitim/deneyim gerektirmeksizin kullanılabilir hale gelmektedir. Ancak, çok çeşitli yazılım firmaları tarafından geliştirilen CASE araçlarının oldukça fazla sayıda ve bu aşamada henüz entegre olmadığını da gözlemleyiz. Oysa CASE araçlarından esas yarar, bu araçların bireysel değil entegre (bütünleşik) kullanılması sonucu sağlanabilir. Bu nedenle Tool IntegratJon" özenle üzerinde durulması gereken bir diğer konudur.

Doğan GÜVEN

YAZILIM GELİŞTİRMENİN geleceği, dizge çözümleyicisinin bilgisayar ekranında geliştirildiği uygulamanın otomatik olarak işletilebilir koda dönüştürülmesini sağlayan CASE teknolojisine doğru yol almaktadır. CASE araçları, yazılım geliştirme yaşam döngüsünün çeşitli aşamalarında birçok işlemi otomatikleştirerek ve dizge geliştiricileri arasında ideal iletişim ortamı kurarak, yazılımın daha kısa sürede, daha dikkatli ve daha az maliyetle geliştirilmesini sağlar. CASE araçlarının geliştirilmesinde, kodlama aşamasının daha az sürede gerçekleştirilmesi

asıl hedef değildir. Hedef, kodun tasarım bilgisine doğrudan bağlı olmasını sağlamaktır. Böylece, ileride yapılacak bakım işlemleri kaynak kod üzerinde değil doğrudan tasarım tanımlamaları üzerinde yapılabilecektir.

Piyasadaki CASE araçlarının, metodoloji sınırlamaları, yönetim zorlukları ve belgeleme yetersizliği gibi nedenlerle verimliliklerinin tamamını göstermemeleri, günümüzde kullanıcılarının daha hızlı artmasını engellemiştir. Yeni tasarlanan CASE araçları, özellikle son yıllarda popülerliği ve kullanımı oldukça yaygınlaşan nesneye dayalı yazılım geliştirme metodolojisi temel almaktadır. Nesneye dayalı yaklaşımda, bütün paylaşılan veri alanlarının nesnelere arası ileti gönderme ile ortadan kaldırılması, nesnenin nitelik ve metodlarının ilgili nesne dışındaki bütün program elemanlarından saklanması, veri ve işlemi çok iyi bir arabirim düzeneği ile gruplayarak modülerliği sağlaması, onu diğer yazılım geliştirme metodolojilerine karşı üstün kılar. Nesne mekanizması, gerçek dünya nesnelere ve ilişkilerine doğrudan bilgisayar gösterimlerine dönüştürülmesi için doğal bir yol sağlar.

Sonuç olarak, yazılım geliştirmedeki sorunlar, tasarımın yaratılmasını hızlandıracak bir nesneye dayalı dizge tasarım dili etrafında birçok araçları bütünleştirerek, statik hataları bulan, eniyilemeyi sağlamak için dinamik olarak tasarımın benzetimini yapan, tasarımı belgeleyen, yazılım kodunu otomatik olarak üreten ve klasik veri türleri yanında görüntü, ses, metin gibi yeni kuşak uygulamaların gereksinimi veri türlerini de destekleyen iş istasyonları için geliştirilmiş CASE ortamları ile ortadan kaldırılabilecektir.

Hamdi YAVUZ

GÜNÜMÜZ YAZILIM geliştirme ortamından beklenen özellikler gitgide artmaktadır. Bu ortamlarda her kullanıcıya ayrı işlemci gücü verebilmek ve gerektiği anda ekleme yapabilmek, aynı anda aynı kullanıcıdan kodlama, belgeleme, ürün izleme ve set işlevlerini yürütebilmek, kaynak dosya ve versiyon yönetimi yapabilmek ve kullanıcıları dinamik olarak projelere dağıtabilmek en önemli konulardır. Bu isteklerin karşılanması için entegre ve çok amaçlı yazılım geliştirme ortamlarından yararlanmak gerekir.

5.2 GÜVENLİK

Aydın KOLAT

BİLGİSAYAR DÜNYASINDA güvenlik gereksinimi ve buna uygun hizmetler çeşitlenerek artmaktadır. *Authentication (Kullanıcı Kimliği Denetimi), Acces Control (Erişim Denetimi), Confidential'ity (Gizlilik) ve Data Integrity (Bilgi Bütünlüğü)* alanlarında merkezi işlemci ve çevre birimleri güvenliğini sağlama amaçlı güvenlik hizmetleri artarak kullanılmaktadır. *Tap, disk, vb. çeşitli korunmuş kaynaklara erişim yetkisi denetimi, kullanıcı kimliği belirleme ve doğrulama, şifreleme, vb. bu hizmetlere bazı örneklerdir.*

Gelecek dönemlerde, bilgisayar güvenliği alanındaki olası gelişmelerin, merkezi işlemciler ve çevre birimlerine ek olarak, özellikle uzak kullanıcı gruplarının erişimi ile değişik mimarilerdeki sistemlerin birbirleriyle iletişimi boyunca ağ güvenliğinin sağlanması alanlarında ortaya çıkması beklenmektedir.

OSI Ağ Güvenliği konusunda çalışma yürüten, IBM'in de üyesi olduğu *Stantartlar Organizasyonları (ISO, CCITT- ECMA, ANSI, NSA - SDNS, MAP/TOP...)* şu ana OSI (Öpen System Interconnect) tabakalarının bazılarını kapsayan standartları geliştirmiş durumdadırlar. Hatta, daha ileri seviyede, tabakalara özgü olmayan güvenlik mekanizmalarını oluşturmayı amaçlayan bir güvenlik mimarisi tanımlanmıştır. Geleceğe yönelik olarak yürütülen çalışmaların kapsamı içinde şu güvenlik mekanizmalarını saymak mümkündür:

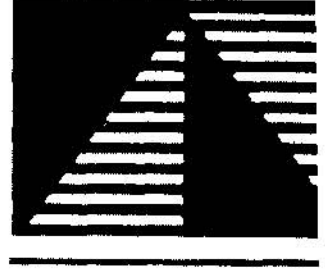
- Encipherment (Şifreleme)
- Digital Signature (Sayısal İmza)
- Access Control (Erişim Denetimi)
- Data Integrhy (Bilgi Bütünlüğü)
- Authentication Exchange (Kimlik Denetim bilgileri Değişimi)
- Routing Control (Yönlendirme Denetimi)
- Notarization (Onaylama)

5.3. YAPAY US VE UZMAN SİSTEMLER

Aydın KOLAT

KARMAŞIK PROBLEM çözme, mantık yürütme, planlama, algılama, öğrenme, gibi insan düşünüş ve davranışına ilişkin fonksiyonların incelenmesinde kullanılan tekniklerden artık, bilgisayarlara benzer özellikler kazandırabilmek için de yararlanılmaktadır. Tümüyle insan beyninin özellikleri olan bu fonksiyonların bilgisayar programlarına yansıtılmasına yönelik çalışmalar, yapay us başlığı altında geniş bir yelpazede bilgisayar projelerine yayılmış bulunmaktadır. Yapay us yelpazesi, doğal dil işleminden görüntü işleme, öğrenilebilir program yazımından robot üretimine kadar pek çok araştırma sahasını kapsamaktadır.

Belli konularda uzmanlaşmış kişilerin bilgi birikimini ve karar verme mekanizmalarını bilgisayar ortamında yaşatmayı amaçlayan uzman sistemle, günümüzde kullanım alanı en yaygın olan yapay us uygulamalarıdır. Uzman sistemler sayesinde belirli uzmanlık alanlarında karşılaşılan günlük problemlerin çözümlerine artık bilgisayarlar yön vermektedir.



Yapay us çalışmaları, dünya çapında birçok üniversite ve araştırma laboratuvarında artan bir hızla gerçekleştirilmektedir. Tüm bu çalışmaların amacı, çevresindekileri algılayan, algıladıklarına doğru tepkiler vererek hareket eden, insanlarla kendi dillerinde iletişim kuran, başarılı ve başarısız deneyimlerinden ders alan, kısacası "akıllı" olabilen bilgisayarı yaratmaktır. Geleceğin düşünen robotlarına her geçen gün biraz daha yaklaştığımızı düşünürsek, ileride okul sıralarında çocuklarımıza "akıllı" bilgisayarların ders vereceklerini düşünmek bir fantazi olmaktan çıkacaktır.

5.5. BENZERİ BAŞKACA KONULAR

Ender AKINCI

HİZMETTEN BAĞIMSIZ YAPI

(Bir yazılım "Bilim - Kurgu" Denemesi)

Bugün yazılımın "gerçek zaman" uygulamalı en yoğun kullanım alanı şüphesiz iletişimdir.



Bunda, gelişimin canlı varlıkların en temel gereksinimlerinden biri olması yanında (solunum bUe bir Bet'tşhn olayıdır), özeSkle insanlığın gelişiminde oynadığı sosyo-ekonomk toplumsal rol de büyüktür. Buradan hareketle, son yirmi yılda fetişimin ne dent yazılım bağımlı olduğu ve yazılımın temeBnde de keSmenin sözlük anlamıyla iletişimin bulunduğu dkkate alınarak BİLİŞİM-İLETİŞİM ikilisinin ayrılmaz bir bütün olması kaçınılmazdır.

Burada hemen bir çağnşım ve buna bağılı bir benzeşim üzerinde durmakta yarar görüyorum: Evrenin bXnen en akıllı varlığı olan insanoğlu bugüne kadar gerçekleştirdiği her şeyi, bütün çabalara rağmen henüz sırrı tam olarak çözülmemiş beynin denetiminde diğer organlarını da kuManarak gerçekleştirmiştir. Beyin, tüm bu karmaşık işlemleri yaparken kapasitesinin maksimum %25lni kuşanmakta ve başlıkta geçen "Hizmetten Bağımsız Yapı" çerçevesinde yukarıda bahsettiğimiz "BİLİŞİM - İLETİŞİM" işlevinin en mükemmel örneğini vermektedir.

İşte bugün, yazılım alanında ulaşılmaya çalışılan da süperbUgisayar alanından iletişim alanına kadar, adını "yapay-us" koyduğumuz beyni taklit etme çabasıdır. Zira beynin herkes tarafından kabul gören en önemli özeliği, öğrenme işlevi ve öğrendiğini kuşanmadaki esnekliğidir. "Hizmetten bağımsız yapı" dan söz ederken söylemek istetBğkn de budur. Yani, fizyolojik yapısı değişmeden her türü etkiye kendi kapasitesi çerçevesinde en "akıla" tepkiyi verme, bunu yaparken de tüm birikimini kullanabilme özeMiği Bu özeUği sağlayacak yazılımların geSştiriknesi ile yakın gelecekte her alanda akıKı yapılar gündeme gelecek (İletişimde "Akıllı Şebeke" gtoi) ve bu bağlamda yapay us çalışmaları çerçevesinde tüm işlevsel yapıların "Hizmetten Bağımsız" ve sadece yazılım ile "BİLİŞİM" yapısı değişen "Evrensel Yapıya (Evrensel Yapay Us) doğru geBşimi söz konusu olacaktır.

Yücel BAĞRIAÇIK

TARİHÇE:

SON YİRMİ YIL İÇİNDE telekomünikasyon sistemleri elektro-mekanfc yapıdan bilgisayar kontrollü yapıya doğru evrim geçirmişlerdir. Bu zaman dilimi içerisinde ortaya çıkan telekomünikasyon sistemlerinin fonksiyonel davranışlarının tanımlanması ihtiyacı CCÜTnin 1968 yılındaki toplantısında ortaya atılmıştır.

Bu görüş doğrultusunda 1969-1975 yılları arasında çalışmalarını sürdüren CCITT çalışma grupları, 1976 yılında SDL (Specification and Description Language) adı altında bir standart ortaya çıkarmıştır. SDL'in 1980 ve 1984 yıllarındaki uyarlamalarını takben 1988 yılında zamanın ortaya çıkardığı ihtiyaçlara göre değişen uyarlaması SDL88, CCITT tarafından* CCITT Recommendation Z 100: SpecificaHon and Description Language SDL, Annexes A-F to Z. 100,1988* olarak resmen yayınlanmıştır.

SDL, gelişme süreci içerisinde araştırma kurumlarının, telekom idarelerinin ve üretici kuruluşların katkılarıyla uluslararası bir işbirliğinin sonucu ortaya çıkmıştır. Başlıca telekomünikasyon sistemlerinin tanımlanmasında kullanılan SDL aynı zamanda sonlu durum makinaları He modellenebilen herhangi bir gerçek zaman sisteminin tanımlanmasında da kullanılabilir.

SOL:

özellikleriyle tanımlama manasına gelen 'specification* deyimi, bir bilginin iki kişi arasında zihinlerde soru işareti bırakmayacak şekilde aktarılması için kullanılmaktadır. Telekomünikasyon sistemlerinin gün geçtikçe karmaşık hale gelmesi bu teknoloji üzerine kurulu sistemlerin tanımlanmasında yeni dillere olan ihtiyacı ortaya çıkarmıştır. Çünkü doğal dillerle yapılan sistem tanımlamaları ki uç arasında karışıklıklara neden olmaktadır; doğal dillerde bir cümleye bazı durumlarda birden fazla yorum getirilebilir. Tanımlama dillerinde amaç taraflar arasında bir fikrin, bilginin net, açık, seçik ve kesin bir şekilde aktarılmasını sağlamaktır. Telekomünikasyon sistemleri gibi büyük, karmaşık ve gündün güne geliştirilen sistemlerde ise böyle bir dilin kullanılması artık kaçınılmaz hale gelmiştir..

SDL ile bir sistemin fonksiyonel davranışı uygulamadan bağımsız olarak tanımlanabilir. Yani verilen bir SDL tanımlamasından donanım ve yazılım yönünden iki farklı sistem gerçekleştirilebilir

SDL bir sistemi çeşitli ayrıntı seviyelerinde tanımlamayı mümkün kılmaktadır. Bunun en önemli uygulaması PTT idareleriyle üretici kuruluşlar arasında ortaya çıkmaktadır. Günümüzde PTT idareleri kullanıcılara çok çeşitli hizmetler sunabilmektedir. İdareler kullanıcılara sundukları bu hizmetleri üretici kuruluşlardan aldıkları sistemler ile gerçekleştirmektedirler.

Sonuç olarak PTTler şebekelerini kurarken veya genişletirken ne istediklerini tam olarak üretici firmalara iletmek durumunda, üretici firmalar ise geliştirdikleri ürünlerin fonksiyonel özelliklerini karşı tarafa net bir şekilde anlatmak durumundadırlar. Bu şekilde sistemler PTT idarelerince daha iyi analiz edilebilirler. Tarafların anlaşılmasını sorunu ancak kuralı bir dilin kullanılması ile giderilebilir. Böylece PTT idareleri ile üretici firmalar arasında istek ve tekliflerin sağlıklı olarak iletimi için SDL gibi kurallı bir dilin kullanılması zorunluluğu ortaya çıkmaktadır.

Tipik uygulama alanlarına örnek olarak telefon, telex, veri anahtarlama, sinyalleşme sistemlerinin veri protokolleri ile karşılıklı etkileşimleri verilebilir, özellikle Yüklü Program Kontrollü (SPC, Stored Programme Controlled) anahtarlama sistemleri göz önüne alındığında çağrı işleme (çağrı ele alma, yönlendirme v.b.), bakım ve hata giderimi (alarm, otomatik hata giderimi, sistem konfigürasyonu v.b.) sistem kontrolü gibi fonksiyonlar SDL ile kağıda dökülebilmektedir. Ayrıca CCITT'nin yapmış olduğu önerilerde protokollerin tanımlanması SDL ile yapılmaktadır.

Yukarıda bahsedilenler ışığında SDL başlıca iki gruba seslenmektedir. Birinci grup, bir sistemi tanımlayan aktif kullanıcılar, idareler, diğeri ise SDL ile yapılmış dokümanları okuyan sistem tasarımcıları, üretici firmalardır. Sonuç olarak tüm PTT idareleri ve üretici kuruluşlar uluslararası tanımlama ve anlatım dilini kullanarak birbirleri arasında standart bir haberleşme ortamı yaratmaktadır.

Bir sistemin gerçekleştirilmesinde toplam zamanın çok büyük bir kısmı tasarım aşamasında ve bunu takip eden süreçte olası tasarım hatalarının düzeltilmesi için harcanmaktadır. SDL gibi standart tanımlama dilleriyle yapılan tasarımlarda çalışma gruplarının birbirleri arasındaki haberleşme sorunları minimuma indiği için bu zamandan büyük ölçüde tasarruf edilmektedir.

Yazılım Formları:

SDL'in aynı model üzerine kurulmuş iki farklı şekli vardır. Bunlardan ilki standart birtakim semboller üzerine kurulan SDL/GR (Grafiksel gösterim), diğeri ise SDL/PR¹ (Cümlelerle gösterimdir).

Birinciyi oluşturan grafiksel anlatımın daha kolay anlaşılabilir olma avantajı vardır. Başlangıçta SDL/PR SDL dokümanlarının bilgisayara aktarımı esnasında kullanılmak üzere düşünül-

müştü. Çünkü o zamanlar bilgisayara grafiksel olarak veri girişi yapabilmek oldukça zordu. Ancak daha sonraları grafit terminallerinin gelişmesi ve yaygın olarak kullanılmaya başlanmasıyla bu sorun ortadan kalmıştır. Günümüzde PCler ve grafik iş istasyonlarında çalışabilen SDL*yi (CCITT SDL 88) destekleyen programlar bulunmaktadır. Bunlardan bazıları kuruluşlarca kendi kullanımları için geliştirilmiş (AT & T, Siemens) bazıları da ticari amaçla üretilmiştir (Telesoft SDT). Bu programlar yardımıyla dokümanlar hazırlanabilmekte, tasarlanan sistemlerin simülasyonları yapılabilmekte ve hatta kaynak programlar üretilebilmektedir.

Sonuç:

Tüm bu anlatılanlar ışığında, SDL tanımlama ve anlatım diline duyulan ihtiyacın nedenleri aşağıdaki şekilde özetlenebilir.

1. SDL, CCITT tarafından tasarlanmış uluslararası standartlara sahip bir dildir.
2. SDL, özelliğide birçok paralel işlemin birbiriyle iletişimde bulunduğu karmaşık gerçek zaman sistemlerinin tanımlanması için tasarlanmıştır.
3. SDL, grafiksel gösterim özelliğiyle kullanıcılar tarafından kolayca anlaşılabilir.
4. SDL yapısal bir dildir.
5. Bir sistem SDL ile çok genel bir şekilden çok ayrıntılı bir şekle doğru seviye seviye anlatılabilir.
6. SDL, PTT idarelerinin telekomünikasyon sistemleri siparişinde ve üreticilerin bu sistemleri sağlamasında eksiksiz bir anlatım getirmektedir.
7. Kuralsız doğal dillerde ortaya çıkan karmaşık yapı, problemin SDL ile herkes tarafından kolayca anlaşılabilir bir şekle dönüştürülmesiyle ortadan kalmaktadır.

Murat ÖNLÜ

YAZILIMIN YARINKİ GÖRÜNÜMÜNDE NESNEYE DAYALI PROGRAMLAMA

5. KUŞAK DİLLERİ DE denen Nesneye Dayalı Programlama dilleri uzun yıllardır bilinmelerine rağmen Dünya'da ve Türkiye'de henüz yaygın olarak kullanılmamaktadır. Nesneye Dayalı Programlama'nın yaygınlaşması için kullanıcıların bu farklı yaklaşımı benimsemeleri başta gelen koşuldur. Ancak yazılım platformlarının da bu geçişi kolaylaştıracak şekilde daha entegre ve kullanıma yönelik tasarlanmaları gerekmektedir.



6 DEVLET BÜROKRASİNDE BİR YENİDEN YAPILANMA GEREKLİ Mİ?

6.1. BİLİŞİM ETKİNLİKLERİNDE KURUMLAŞMA

Semih BİLGEN

GÜNÜMÜZDE BİLGİSAYAR ve iletişim teknolojileri, başka bir deyişle elektronik ve tüm yönleriyle kullanımı, ekonomik etkinliklerin en önemlilerindedir. Dünyanın en büyük kuruluşları arasında, enerji sektörünün yanısıra bilişim sektörü kuruluşlarını görmekteyiz. Ülkemizde de, dünyada da bilişim etkinlikleri, ekonomik etkinlikler içinde önemli bir pay oluşturmaktadır.

Bilişimin önemi, birçok ülkede kurulan ilgili bakanlıklarla, hükümet etkinliklerine yansıtılmış bulunmaktadır. Japonya'da Bilişim bakan-

artık "bilişim" çerçevesinde oluşturulmalı ve yürütülmelidir. Bu çerçevede elektronik sanayi, bilgisayar donanımı ve yazılım projelerinin desteklenmesine, kullanımın yaygınlaşmasına ve özellikle kamu kesimindeki kullanıma eşgüdüm sağlanmasına ilişkin hükümet politikalarının oluşturulması ve yürütülmesi için Bilişim bakanlığı'nın kurulması Türkiye'den beklendiği sık sık vurgulanan "yazılım hamlesinin" gerçekleşmesine de katkıda bulunacaktır. Bu bakanlığın temel işlevleri arasında uluslararası uyumluluğun gözetilmesi, bu amaçla teknolojik standartların izlenmesi ve en önemlisi oluşumlarına katkıda bulunulması olmalıdır. Bakanlık bu bakımdan TSE ile yoğun işbirliğini gerçekleştirmelidir.

Teknolojiye, özellikle de bilişim kadar hızlı gelişen bir teknolojiye katkının kişisel ve kopuk kopuk çalışmalarla gerçekleştirilmesi düşünülemez. Bu katkı örgütlü, destekli, ulusal ve uluslararası düzeyde eşgüdümlenmiş, araştırma, geliştirme, üretim, pazarlama ve özellikle kamusal kullanım yönlerini birlikte planlayıp gerçekleştiren çalışmalarla ortaya çıkabilecektir. Bilişim Bakanlığı bu çalışmaların çekirdeğini oluşturmalıdır.

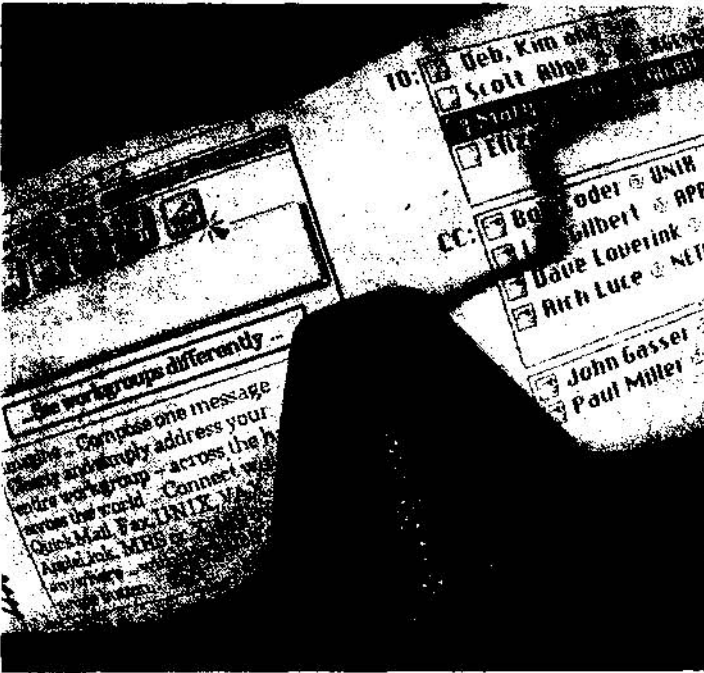
Selim BOZOK

TARİHÇİLER, İLERİDE 20. Yüzyılın incelenmesini yaparken; Bu Yüzyılın son çeyreğini bilgi çağı'nın başlangıcı, olarak tanımlayacaklardır. 60'lı yıllardan başlayarak bilgisayarın oluşumunu ve 30 senede yaşanan devrimi düşünürsek, bu yargıya katılmamak elde değildir.

Ülkemizde de benzer bir gelişme yaşanmıştır. Orta ve büyük çaplı işletmelerde vazgeçilmez bir unsur olan enformasyon sistemlerinin bir kaç sene içerisinde küçük işletmelerde de kullanılmaya başlanılacağı muhakkaktır.

Ne yazık ki bu kadar hızla benimsediğimiz bu yeni teknoloji ve onu oluşturan kavramların gerektirdiği yeniden yapılanma da, her zaman olduğu gibi gene geç kalmaktayız. Halbuki daha işin başında iken gelecekte büyük sorunlar yaratabilecek konularda gerekli altyapının kuruluşuna başlanılmalıdır.

Bir düzene kavuşturulması gereken ve bilgisayar kullanımı ile doğrudan bağlantılı konulardan bazıları şunlardır.



lığı, Hindistan'da Elektronik Bakanlığı, Çin Halk Cumhuriyeti'nde Elektronik Sanayi Bakanlığı, birçok Avrupa ülkesinde ise bilim ve teknoloji bakanlıkları yada genel müdürlükleri bu konuyla ilgili politikaların belirlenmesi ve uygulanması sorumluluğunu taşımaktadır. Bugüne dek ülkemizde ulaştırmanın bir parçası olarak kabul edilmiş olan iletişime ve hiçbir hükümet etkinliğinde gereken yeri alamamış, yakın geçmişe kadar gümrüklerde yalnızca "muhasabe makinası" olarak tanımlanabilen bilgisayarlara ilişkin politikalar

1. Hukuksal Düzenlemeler:

Yasalarımızın ve Yönetmeliklerimiz, bu yeni bilgiyi kullanma, saklama, dağıtma sistemlerini düzenleyecek biçimde gözden geçirilmelidir. Karşılaşılabilecek sorunlara bir kaç örnek verebiliriz;

a- Çalıştığı şirketin bilgisayar kayıtlarından hayati bilgileri silen bir programcı nasıl yargılanacaktır?

b- İş değiştiren bir programcı beraberinde eski şirketinin bilgisayar programlarını kendisinin geliştirmiş olması nedeni ile götürüp başka bir şirkette kullanabilir mi?

c- Bilgisayarlar arası veri iletişimde elektronik faturalar ve irsaliyeler ne zaman geçerlilik kazanacaktır?

2. Bilgi Alışverişini Düzenleyecek Bilişim Standartlarının Saptanması

Bankalarımız, büyük şirketler ile çek/senet bordrolarının elektronik ortamda alışverişine başlamışlardır. Benzeri biçimde maaş ödemeleri de bilgisayar kayıtlarının hesaplara transfer yolu ile yapılmaktadır. Ne yazık ki bu konuda da bir standartlaşma gereği düşünülmediği için banka sayısı kadar kayıt tipi oluşmuştur ve gün geçtikçe de düzensizlik yaygınlaşacaktır.

Örneklerle kısaca değinmeye çalıştığımız konunun devlette bir sahibinin olması gerektiği kanısındayız. Gerekli yetkilerle donatılmış bir kuruluş ancak bir bilişim bakanlığı olabilir.

Baki DEMİREL

TARIM, SANAYİ, TİCARET vb. gibi bakanlıkların yanı sıra bir Bilişim Bakanlığı ya da benzeri bir yapılanma oluşturulmalı.

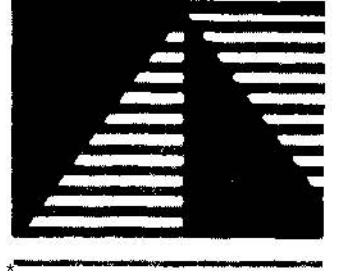
Bilgisayar ve iletişim teknolojilerinin geliştiği günümüz dönemi enformasyon çağı olarak nitelendirilebilir. Enformatikteki gelişmelerin zorlaması ile iş dünyası, devlet, eğitim ve araştırma kurumlarında yeniden yapılanma gündeme gelmektedir. Bilgi patlamasından gerekli ve işe yarayan bilgiyi ayıklayabilen ve bununla elde tutulur yararlar sağlayan bilişim sistemlerine sahip ülkeler, yiyecek, üretim, sağlık, eğitim ve doğal kaynakların kullanımı gibi konularda sağladıkları avantaj ile refah toplumuna daha da çok yaklaşmaktadırlar.

Zaman ve kaynak kullanımında çığır açan bu avantajlardan yararlanmak için gerekli olan yapılanmanın bir ölçüde oluşturulmasının ve denetiminin sorumluluğunu taşıması gerekli olan bir oluşuma mutlaka ihtiyaç vardır. Bu amaçla kendi alanlarının gelişiminden sorumlu olan Tarım, Sanayi, Ticaret, v.b. gibi bakanlıkların yanı sıra bir bilişim bakanlığı yada eşdeğeri bir yapılanmaya mutlaka gidilmelidir.

Çığdem ERTEM

BİLGİSAYAR UYGULAMALARININ son derece yaygın ve kaçınılmaz olduğu bugün, bilgisayarsız bir ortama göre yapılandırılmış devlet bürokrasisinin yeniden bir yapılanmaya gitme gerekliliği açık bir şekilde ortadadır. Çünkü, şu anda bir çok devlet kuruluşunda bürokrasi, devlet dairelerinde iş yürütme peşinde olan vatandaşları bezdiren manuel form akışları ile yürütülmektedir. Bu tip devlet dairelerinde bilgisayar kullanılsa bile, işler çoğu kez mükerrer olarak yapılmakta, bilgiler öncelikle yıllar önce düzenlenmiş ve genellikle hiç bir işlerliği kalmamış formlara kaydedilmekte ve daha sonra bilgisayara geçirilmektedir. Bilgi sistemi yapısına geçilirken, mevcut yapı, bilgisayarın etkin kullanımına imkan verecek şekilde yeniden tasarlanacağı yerde, bilgi sistemi mevcut yapıya uydurulmaktadır.

Bu konuya çözüm olarak, hükümette bu konuyla ilgili bir bakanlık açmak soruna tam bir çözüm oluşturmayacaktır. Bilişim bakanlığı veya benzer bir ad altında oluşturulan bu bakanlık aslında bilişim işlevini içlerinde bulunduran diğer bakanlıkların yeniden yapılanmasına etkin bir katkıda bulunmayacaktır. Buna ek olarak, böyle bir bakanlık hükümet politikası değişikliklerinden de fazlası ile etkilenenecektir. Bu nedenle, konunun en duyarlı çözümü, olabildiğince özerk ve yaptırım gücü olan bir enstitü kurularak, bilişim işlevleri kontrolünün bu kuruma devredilmesi olacaktır. Türk standartları Enstitüsü veya Türk Dil kurumu gibi bir statüde çalışacak bu enstitü, Türkiye'de ihtiyaç duyulan bilişim sektörü standartlarını belirlemede ve Devlet bürokrasisinin yürütülme biçimini bilgisayar kullanarak etkin, veri tekrarsız ve daha verimli iş gücü kullanımına imkan sağlayacak bir hale getirmede yararlı olabilir.



Selim SARPER

DAHA ÖNCE DE değinildiği gibi, yazılım tamamen zihinsel bir çalışmadır. Bu nedenle ilgililerin zihin güçlerinin tamamını işlerine ayırbilmeleri için onların kafasını meşgul edecek (gümrük, işçi - işveren ilişkisi, çalışma koşulları, vergilendirme, telif hakları v.b.) her türlü konu, devletin oluşturması gereken en üst düzey bir komite (Bakanlık?) tarafından düzenlenmelidir.

Bu konunun devlet tarafından en üst düzeyde önem taşınması gerekliliği, devlet için ne demekse, o yapılmalıdır.

M.Danis UNVERDI

OLDUKÇA UZUN BİR SÜRE yurt dışında okumuş ve çalışmış olmam dolayısıyla, Batı memleketlerinde genelde yüksek teknoloji (Hitec), özelde de yazılım konusunda çalışmakta olan şirketlere, yaptıkları işin boyutuyla orantılı olarak devletin doğrudan bir yardım yaptığını yakından biliyorum. Doğal olarak, bu yardımın kaynağı vatandaşlardan toplanan vergilerden bir başka deyişle vatandaşların yüksek teknoloji ile uğraşan şirketlere parasal

destek sağlaması bir hükümet politikasıdır. Hükümetler bu konuda, pazar ekonomisinin kurallarına aykırı düşmek gibi bir komplekse veya kaygıya kapılmamakta, sonucu olarak bu paranın bir şekilde ülke refahının yükselmesine katkıda bulunacağına hesabını yapmaktadırlar. Bu konuda, hükümetin bir başka devlete şartlı yardım olarak verdiği paranın, bu devlet tarafından yapılan bir satın alma ile çalışmış olduğum şirkete geri transfer edildiğini bizzat gördüm.

Bilhassa, yazılımın henüz canlanma aşamasında olduğu Türkiye örneği ülkelerde devletin gerçek anlamda yazılımla uğraşan şirketlere doğrudan destek vermesini kaçınılmaz görüyorum. Bu konuyla ilgili bir bakanlık kurulması düşüncesine olumlu bakıyorum.

Cavit ÜNSAL

YAZILIM ÜRÜNLERİNİN her tür yaşam akt/itelerini, içine alıp düzenleme, süreç ve sonuçlandırma bakımından etkileyebileceği çok açık yaşanmaktadır. Yazılım ürünleri ve sanayi, yaşamı etkileyen bir yeni kültür olgusudur. Bu kültür olgusu sanayide ve tarımda verimi artırmak, ticarete muhasebenin daha sıhhatli yapılmasını sağlamak ve sanatta güzeli ve iyiyi yakalamada yaratıcılığı geliştirmeyi sağlayan bir toplumsal olaydır. Düzenli işleyebilmesi için koruyucu ve bilgi almayı sağlayıcı güvenilir kurumlara ihtiyaç vardır. Bu yüzden bu düzeni sağlayacak bir "BİLİŞİM BAKANLIĞI'nın kurulması çok yararlı olur.

Bilişim Bakanlığı kurulabileceği gibi, bu konuda dernekleşme de yararlı olabilir. Bakanlık ve derneklerden oluşturulan bu kuruluşlar, kullanıcılara yazılım olgusunun geliştirilmesi ve belki de bu güne kadar ihmal edilmiş olan "Bilgisayar Virüs" üne karşı tedbirli olma ve savaşma konusunda yön verebilirler. Ayrıca bu yolla bilgisayar sistemlerinin maliyeti, gümrük sorunları, standartları ve yazılımların telif hakları meselelerine de çözüm getirmek mümkün olabilecektir.

Bu tür kuruluşlar programcılık veya tasarımcılık konusunda insan hayatını kolaylaştırıcı işlemlere ve amaca yönelik çalışmalara ve araştırmalara açık olmalıdır.



Ö Z G E Ç M İ Ş L E R

BAKIRGÖZ Oktay Aydın

1967 yılı CÜMÜRTED BİLGİSAYAR MÜHENDİSLİĞİ MEZUNU.
1989 BİFCENT ÜNİVERSİTESİ. BİLGİSAYAR MÜHENDİSLİĞİ
VE ENFORMATİK BİNİLERİ, YÜKSEK LİSANS.
ASELSAN MST GRUBU BHYM

AKINCI Ender (Dr.)

Telekomünikasyon Yüksek Mühendisi mezunu,
Doktora çalışmaları (Fransa) TELETAŞ Santral
Ürün Müdürü

ALUÇ, Tolay

1972 İTÜ Elektrik Fakültesi mezunu.
TELETAŞ

ALTAY Müjdat

İTÜ Elektrik Mühendisliği Bölümü mezunu. BÜ
Biyomedikal Müh. Yüksek lisans.
TELETAŞ Ürün Araştırma Geliştirme Direktörü.

ARIK Şebnem

1969 yılı İTÜ Fen Bilimleri Enstitüsü Kontrol -
Bilgisayar Müh. Master Programı mezunu.
Big Bayar Yüksek Mühendisi.
TELETAŞ AR-GE Direktörlüğü Bilgisayar Destekli
Tasarım Bölümü.

ARIKAN Ferhat

1989 yılı OOTÜ Elektrik ve Elektronik Mühendisliği
Bölümü mezunu. 1982 aynı bölümden Yüksek
Lisans. ASELSAN MST Grubu. Big Bayar
Ürün Araştırma ve Yazılım Müdürlüğü, Yazılım
Mühendisi.

ARABACIOĞLU LGüzide

1985 Boğaziçi Üniversitesi Elektrik ve Elektronik
Mühendisliği Bölümü mezunu.
TELETAŞ Santral Müdürlüğü. Ürün Müdürlüğü
Santral Bölüm Şefi.

ATLJ Selami

1969 Yılı OOTÜ Elektrik Mühendisliği Bölümü
Mezunu. 1972 Yüksek Lisans. ASELSAN.
Mikrodajga ve Sistem Teknolojileri Grubu,
Bilgisayar Hizmetleri ve Yazılım Müdürü

AZİZOĞLU M.Cemil

1990 ODTÜ Bilgisayar Mühendisliği Bölümü
mezunu. ASELSAN Bilgisayar Hizmetleri ve
Yazılım Müdürlüğü, Yazılım Mühendisi.

BAĞRIAÇIK Yücel

ODTÜ Elektrik ve Elektronik Mühendisliği Bölümü
Mezunu. ODTÜ Elektrik-Elektronik Bölümü Yüksek
Lisans Çalışması. PTT AR-GE

BARAY Mehmet (Prof. Dr.)

Bient Üniversitesi, Bilgisayar Mühendisliği
Bölümü.

BAYSAN Suat

Manchester Technkal University Mezunu.
TELETAŞ Teknoloji, AR-GE ve Ürün Yönetimi
Genel Müdür Yardımcısı.

BELGÜL Ahmet

İTÜ Elektrik Mühendisliği Bölümü Mezunu.
İTÜ Elektrik Mühendisliği Bölümü Yüksek Lisans.
TELETAŞ Yeni Ürün Takdimi ve Yazılım Kalite
Temini Müdürü.

BOZOK Selim

1971 Yılı Robert Kolej Makina Mühendisliği, 1972
Yılı Virginia Polytechnic Institute Endüstriyel Mühendisliği
M.S. Bölümleri mezunu.
PEGA.Ş. Bigi İşlem Müdürü.

CİVELEK Meltem

BM Türk Ümited Şirketi. Uygulan Yazılım
Geliştirme Merkezi

ÇMAR Bahadır

BÜ Elektrik, Mühendisliği Bölümü Mezunu, BÜ
Bilgisayar Mühendisliği Bölümü Yüksek Lisans.
DMS AR-GE 1. Uluslararası Tasarım Müdürü.
TELETAŞ

DEMİREL Baki

OOTÜ Bilgisayar Mühendisliği Bölümü Mezunu.
ASELSAN MST Grubu BHYM

DOKUZOĞLU M.affer

1982 yılı OOTÜ Elektrik Mühendisliği Bölümü
Mezunu. 1985 Aynı Bölüm Yüksek Lisans.
ASELSAN Mikrodajga ve Sistem Teknolojileri
Grubu. Bilgisayar Hizmetleri ve Yazılım
Müdürlüğü, Şef Mühendisi.

ENFİYECİ Kemal

1989 yılı OOTÜ Bilgisayar Mühendisliği Bölümü
Mezunu. ASELSAN Bilgisayar Hizmetleri ve
Yazılım Müdürlüğü. Yazılım Mühendisi.

ERGEÇEN Eyüp

Türkiye Yazılımlar Birliği
Yürütme Kurulu Başkanı

ERGÜL Güllü

BM Türk Ümited Şirketi. Çözüm ve Yazılım
Bölümü

ERTEM Çiğdem

1989 OOTÜ Bilgisayar Mühendisliği Bölümü
mezunu. ASELSAN

GÜVEN Doğan (Dr.)

1986 yılı Hacettepe Üniversitesi Bilgisayar
Mühendisliği Bölümü Mezunu. Ege Üniversitesi
Bilgisayar Mühendisliği Bölümü Yüksek Lisans ve
Doktora çalışmaları.
EGE Üniversitesi, Bilgisayar Mühendisliği Bölümü

GÖKÇE Haluk

Digital Equipment Türkiye A.Ş.

GÜLER Günay

1967 OOTÜ Elektrik ve Elektronik Mühendisliği
Bölümü Mezunu. TELETAŞ. AR-GE Direktörlüğü.

İNAN Kemal (Prof. Dr.)

Orta Doğu Teknik Üniversitesi
Elektrik - Elektronik Mühendisliği Bölümü

KOLAT Aydın

BM Türk Ümited Şirketi. Çözüm ve Hizmetler
Bölümü

KUTLUAY Sina

1963 yılı OOTÜ Elektrik ve Elektronik Mühendisliği
Bölümü Mezunu. 1986 yılı Yüksek Lisans.
ASELSAN Mikrodajga ve Sistem Teknolojileri
Grubu, Bilgisayar Hizmetleri ve Yazılım
Müdürlüğü, Şef Mühendisi.

ÖZÇIKMAK Kamil

1983 Marmara Üniversitesi Teknik Eğitim Fakültesi
Elektronik-Bakım Anabilim dalı mezunu. Marmara
Üniversitesi Fen Bilimleri Enstitüsü
Elektronik-Elektronik Ana Bilim Dalı Yüksek Lisans
Çalışması. TELETAŞ Elektronik Bakım Mühendisi

ÖZEL Fatih Mehmet

SPEKTRUM Elektrik* sahibi
PAN ELEKTRONİK Ltd. Şti. Genel Müdürü

SARPER Selim

1984 yılı Elektrik* Yüksek Mühendisi Mezunu
(Fransa) TELETAŞ, S 12 Yazılım Departmanı
Müdürü.

SİVİKNİHAN

1991 İTÜ Elektrik - Elektronik Fak. Kontrol ve
Bilgisayar bölümü Mezunu. TELETAŞ AR-GE
Bilgisayar Destekli Tasarım Bölümü, Sistem
Destek Mühendisi

ŞEKERCİ Mukaddes

1986 yılı İTÜ Kontrol ve Bilgisayar Bölümü
Mezunu. TELETAŞ AR-GE Bilgisayar Destekli
Tasarım Sistem Sorumlusu.

TURAN Fulya

OOTÜ Elektrik Mühendisliği Mezunu. Ürün AR-GE
Yazılım Geliştirme Müdürü TELETAŞ

UNCU Oğuz

1987 yılı OOTÜ Bilgisayar Mühendisliği Bölümü
mezunu. Aynı bölümde Yüksek Lisans. ASELSAN

UNVERDİ M.Danis

Mathematiques Supérieures. Mathematiques
Speciales (Lyon-Fransa) Ecole Polytechnique
(Paris-Fransa) Ecole Nationale Supérieure
des Télécommunications (Paris-Fransa) TELETAŞ
Sistem-12 Sayısal Telefon Santral Yazılım
Bakım Şefi.

ÜNLÜ Murat

Boğaziçi Üniversitesi Elektrik Mühendisliği Bölümü
mezunu. TELETAŞ AR-GE Sistem Planlama.

UNSAK Cavrak

1983 yılı Anadolu Üniversitesi, Elektronik*
Teknikliği Bölümü Mezunu.
TELETAŞ Santral Müdürlüğü

YALTI Erdal

1976 OOTÜ Elektronik Mühendisliği mezunu.
TELETAŞ Yeni Ölçü Düzenleri ve Yazılımlar
Bölüm Şefi.

YAVUZ Hamdi

İTÜ Elektrik Mühendisliği Bölümü mezunu.
İTÜ Bilgisayar Yüksek Lisans. TELETAŞ Ürün
Geliştirme Ortamı Müdürü

YAZILIM SAYISINDA YER ALAN BAZI TEKNİK TERİMLER

Teknik Terimler bölümümüzde Yazılım özel sayılarımızın ilki olarak hazırladığımız bu sayıdaki kimi yazılarda geçen sözcüklerden seçtiğimiz bir minik derlemeye yer veriyoruz.

S / W : yazılım (software)
H / W : donanım (hardware)
IC : tümleşik devre
SDL : belirtim betimleme dilleri (specification description languages)
FDL : biçimsel betimleme teknikleri (formal description techniques)
OBJECT - ORIENTED : nesneye dayalı, nesnel kökenli

OSI : açık dizge bağlantısı (öpen system interconnection)
SYSTEM DESIGN : dizge tasarımı
PC : kişisel bilgisayar (personel computer)
USER FRIENDLY PROGRAM : Kullanımı rahat ve kolay olan program.
OPTICAL CHARACTER RECOGNITION : görsel harf tanıma

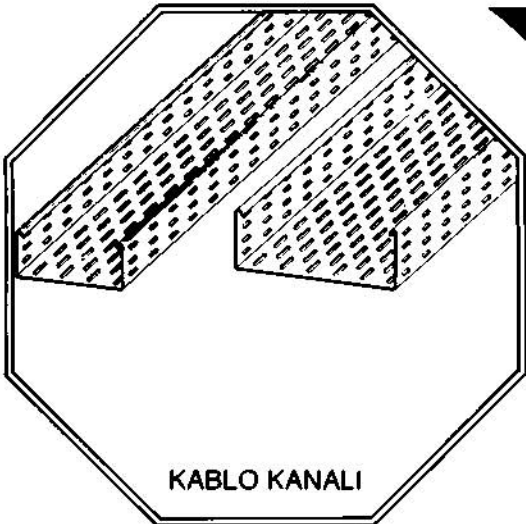
ELEKTRİK MÜHENDİSLİĞİ DERGİSİ
Adresinize Düzenli Olarak Gönderilmektedir.
LÜTFEN ADRESLERİNİZİ GÜNCELLEŞTİRİNİZ.

EKİP

ELEKTROMEKANİK KONSTRÜKSİYON
İMALAT PAZARLAMA SANAYİ VE
TİCARET LIMITED ŞİRKETİ

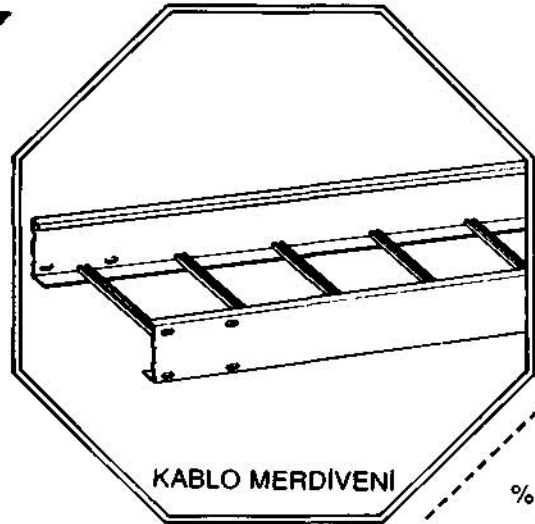
Yeni Ostim Sanayi Sitesi
41. Sokak No. 38 06370 ANKARA
Tel : (4) 385 11 27 PBX Fax: (4) 385 11 29

KABLO TAŞIMA SİSTEMLERİ



KABLO KANALI

* Kablo Kanalları,
* Kablo Merdivenleri,
* Döşeme Kanalları,...



KABLO MERDİVENİ

- MODERN «GÜVENLİ • UZUN ÖMÜRLÜ «EKONOMİK
- Sizler için en yetkin EKİP



% 5
EXTRA
İNDİRİM
KUPONU

Ayrıntılı bilgi için Okuyucu Servisi Kuponu'ndan (12) numrunu iğareileyiniz.