

# Yazılım Mühendisliğinin Gelişimine Bir Bakış

## A Look at the Development of Software Engineering

**N. Kaya KILAN**  
Başkent Üniversitesi, Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü  
[kkilan@baskent.edu.tr](mailto:kkilan@baskent.edu.tr)

### Öz

*Bilimin tarihsel gelişim süreci, karşılaştığımız problemlere çözüm çıkışı bulmada yardımcı çıkarımlar içerebilir. Bilgisayar ve beraberinde oluşan bilgisayar bilimleri ve bilgisayar mühendisliği, yaşanan dünyayı tümü ile etkileyecek teknolojiler oluşturmuştur. Bu gelişimin önemli bir örneği yazılımdaki gelişmeden doğan “Yazılım Mühendisliği”dir.*

**Anahtar Sözcükler :** *Bilgisayar Bilimleri, Bilgisayar Mühendisliği, Bilgisayar Donanımı, Programlama Dilleri, Yazılım Bunalımı, Yazılım Mühendisliği*

### Abstract

*The historical development process of science and engineering may provide support and effective assistance to solve the various problems. Computers, computer science and engineering together create new technologies which have affected the life over the world. One important example of this development is the birth of “Software Engineering” from the developments in software*

**Keywords :** *Computer Science, Computer Engineering, Computer Hardware, Programming Languages, Software Crisis, Software Engineering.*

### 1. Giriş

Bilgisayar mühendisliği ile uğraşanlar çoğu kez konularının tarihsel süreci ile çok ilgilenmezler kanısındayım. Bu sonuç, birçok kavram ve olgunun yeniden, yeni gibi ortaya çıkması ile sonuçlanabilir. Ben inanıyorum ki, kavram ve olguların ilk doğuşu konusundaki nedeni ve ortaya konuşu, terim ve kavramlar kendisi kadar kıymetli ve yarar saçıcıdır.

Genç bir bilim ve mühendislik dalı olan *Bilgisayar Bilimlerinin* ve *Mühendisliğinin* kuram ve yöntemleri bir kaç on yıl denecek aralıkta, hiçbir alanda olmadığı kadar, yenilik ve yöntem üretimi ile genişlemiş ve kapsayıcı olmuştur. Bu genç bilim ve mühendislik dalı, bu çok hızlı değişimini bir öncekilerin üzerine yeni tuğlalar koyarak, bazen de çok farklı yöntemlerle yön değiştirerek, en önemlisi de *Elektronik Sayısal İşlemlere* ve *Von Neuman*'nın *Belleğe Yüklü Program (Stored Program)* mimarisine yaslanmasına borçludur. Bilgisayar bilimlerine koşut olarak uygulamadaki gelişmeler kuşkusuz programlama dillerindeki sürekli değişim ve gelişimle bağlantılı görülebilir [2][8]. Programlama dillerindeki değişimin; uygulama gereksinmesini karşılamak ve bazen de donanımsal gelişimin kullanımını sağlamak için önem kazandığını görmek yanlış olmaz sanırım.

---

Gönderim ve kabul tarihi : 09.12.2015 – 22.12.2015

Özellikle anabilgisayarların (*main frame*) doğmaya başladığı öncelikle, atmışlı ve yetmişli yıllarda, programlama dillerinin “*problemlerle*” etkileşimini *Bilimsel (Scientific)* ya da *İş alanı (Business)* gibi iki farklı alan konuları üzerinde geliştiği söylenebilir. İkinci dünya savaşının içindeki yıllarda savaş problemlerine yönelik çalışma ve özel uygulamalar bir kenara bırakılırsa, 1950’ler sonrası üniversitelerde ve araştırma merkezlerinde *anabilgisayarların* yer almaya başlamasının, bilgisayar bilimlerinin ve daha sonraları Bilgisayar Mühendisliğinin gelişmesine disiplinli bir yol açtığı söylenebilir. Başlangıçta matematiksel ve mühendislik uygulamaları konularında “*problem*” düzeyinde uygulamalar geliştirilirken ayrıca, problem iş alanlarına yönelirken *ana ve dış bellek büyüklüklerinin* ve işlemci hızının artması önemli bir yenilik getirmiştir [6][7].

## 2. Programlama ve Programlama Dilleri

Uygulama ağırlıklı çalışmalar artarken “*programlama*” adını alan bilgisayar uygulama yöntemi önemini giderek artırıyor. Tasarımı güçleşen yapılandırmayı, karmaşık bilgisayar işlem kodları ile bilmece çözer gibi, bir oyun izlemi becerisi ile başarabilenler “*programcı*” olarak kendini yetiştirme çabasına giriyordu. Özellikle ABD’de programcı “*ayrıcalıklı kişi*” nitelemesi kazanmıştı. Uygulama başarısının kuralları olarak bulgularan programlama yöntemi, belki de henüz bir kuramsal içeriğe ulaşmadığından yetersiz kalması nedeni ile 1960’ların temel sorunlarından biri olarak ortaya çıkıyordu. Kuşkusuz sorun, yalnızca yöntem değil, *programlama dillerindeki* yetersizliğin de, programlamayı güçleştiren temel nedenlerden biri olduğu görülüyordu. Bilgisayar üreticileri ve kimi akademisyenler; “*programcının*” çabasını ve

tasarımını kolaylaştırmak için program geliştirmeye kuramsal çözümleme yolunu açan, programlama sürecinin bilgisayarla özleştirmeyi kolaylaştıran, hem problem hem de donanımsal işlevin yürütümünü tanımlayan, *yeni programlama dilleri* geliştirmeye önem vermek zorunda kalıyordu.

İlk “*probleme yönelik*” programlama dili *FORTRAN*<sup>5</sup> (*Formula Translation*), IBM adına 1954 de *John Backus* tarafından geliştirildi. Bilimsel ve mühendislik uygulamalar için geliştirilen bu dilin, konusunda önemli bir boşluğu doldurduğu söylenebilir. Bu çalışmanın bir özelliği, dili *Makine Dili*’ne dönüştüren derleyicisinin (*compiler*) de birlikte hazırlanmasıdır. Diğer önemli bir aşama, 1962’de ABD Deniz Kuvvetlerindeki çalışmanın ürünü olan ve Bayan *General Grace Hopper*’ın çalışmaları ile anılan *COBOL* (*Common Business Oriented Language*) dilinin geliştirilmesidir. Bu dil ve beraberindeki uygulama tasarımı yaklaşımı, bilgisayarları iş-alanı ve ticari uygulama alanlarında başarılı uygulamalara götürdüğü söylenebilir [2].

Bu iki dev yazılım dilinin geliştirildiği yıllar aralığında geliştirilen *ALGOL* (*ALGOrithmic Language*) dilinin kendinden çok, bilgisayar disiplininin iki ayrı yakada (ABD ve Avrupa) uğraş veren çalışma gruplarını bir araya getirip, bilgisayar program geliştirme yöntem ve dillerinin artan sorunlarını birlikte çözmeye zorlaması ayrı ve büyük önem taşır. Örneğin, 1958 de önce; bilimsel programlama ve sonunda iş-alanı uygulamalarında kullanılması amaçlanan *ALGOL-58* için 1960 da Paris’te; Avrupa’dan *Friedrich L. Bauer, Peter Naur*, ile ABD’den *John W. Backus, Julien Green, Charles Katz, John Mc Carthy, Alan J. Perlis*, ve diğerlerinin katılımı ile yapılan *ALGOL 60* toplantısı, birlikte çözüm bulma yolunu açan ve

<sup>5</sup> 1954-57 de genel amaçlı sayısal- bilimsel ve mühendislik problemleri için geliştirilen dil, başlangıçtan bu güne gelişen programlama yöntemlerine paralel olarak sürekli geliştirilmektedir. Örneğin; 1977 de Karakter İşlem ve Yapısal Programlama özellikleri eklenerek *FORTRAN-77*, genetik programlama özellikleri

eklenerek *FORTRAN-90*, işleme hızı ve ondalık sayı boyu genişletilerek *FORTRAN-95*, Nesne Programlama özellikleri eklenerek *FORTRAN-2003* ve Eşzamanlı Programlama özelliği eklenerek *FORTRAN-2008* olarak yenilenmekte ve Süper Bilgisayarların vazgeçilmez uygulama dili olarak kullanıla gelmektedir [1][2].

bilgisayar bilimleri ve uygulama alanını geliştiren atılımdır. ABD Savunma Bakanlığı için geliştirilen ADA<sup>6</sup> ve ALGOL Dilleri üzerinde farklı grupların çalışmaları ilerleyen yıllarda PASCAL, C++, Java Dillerinin kuluçkası olmuştur [4].

Konuya yukardan baktığımızda; Bilgisayar Bilimleri ve Mühendisliği disiplini Avrupa'da kuramsal bir özel alana doğru sürüklenme tehlikesinden kurtulurken, *Bilişim (Informatics)* ve Amerika'da *Bilgisayar Bilimleri (Computer Science)* isimleri ile büyüyen bir anabilim dalları olma kazanımını görüyoruz.

1965-1972 Yılları, programlama yöntemine kıta Avrupası, kuramsal ve bilimsel bir içerik kazandıran çalışmaların odağı olmuştur. Örneğin; *Donald Ervin Knuth* 1963 de yayınlamaya başladığı 4. Cildin 2001 de yayımlanan *Bilgisayar Programlama Sanatı (The Art of Computer Programming)*, 1965 de *C.A.R. Hoare Veri Yapıları (Data Structures)* başlıklı çalışmaları konusunda ağırlık taşır. *E.W.Dijkstra* programlamaya disiplin ve boyut getiren *Yapısal Programlama (Structured Programming)* görüşünü açıklaması ardından, yapısal programlama dilleri ve örneğin PASCAL Dilinin geliştirilmesi, programlamayı hem bilimsel hem de ticari alanda geniş ve yeni ufuklara götüren öncü çalışmalar olmuştur [4].

Programlama dillerindeki gelişmeler sürerken örneğin, ABD'de Savunma Bakanlığındaki bilgisayar uygulamalarında ortaya çıkan problemlerine ivedi çözüm için üst düzey yapısal programlama dili ADA Programlama Dili geliştirildiğini görüyoruz. Honeywell'den *Jean Ichbiah* tarafından geliştirilen bu dilin *Yapısal programlamayı (Structured Programming)* genişleten ve *Nesne programlamaya* götüren dil

---

<sup>6</sup>Ada Programlama Dili; Nesne Yönelimli Programlamaya kapı açan dil, Avrupa'da modern bilgisayarların başlangıcı sayılan programlanabilir "Analitik Makine"yi tasarlayan İngiliz matematikçisi [Charles Babbage](#)'in asistanı ve ilk

çığı açtığı söylenebilir [3][10]. 1963'lerde, *Zaman Paylaşımı (Times Sharing)* ana-bilgisayarların ortaya çıkması, iş alanlarına yönelik uygulamaları bu sistemlere taşımayı zorunlu kılarken, uygulama problemlerinin *Çok Büyük Programlama (Programming too big)* zorluklarına sürüklenme sorunu doğurmuştur. Bu gerçek sonuçta, "*Proje kapsamında Yazılım Geliştirme*" konusuna yönelimi yarattığı söylenebilir [5][7].

### 3. Büyük Yazılım

Yazılım alanında beklenmedik bir sıçramanın, ikinci dünya savaşının çok yönlü etkileşim alanı içinde; *Çok Büyük Yazılım (Too Big Software)* ya da *Büyük Programlama (Programming in the Large)* sorunu ile karşımıza çıktığını görüyoruz. Özellikle ABD 'de askeri projelerin bilgisayar uygulaması desteğine alınma isteği, *büyük programlama* sorunlarının giderilmesini zorunlu kılmıştır. Bu alanda önemli bir olay; 1950-1958 Yıllarında soğuk savaşın ABD'deki etkisiyle üniversitelerin ve büyük bilgisayar şirketlerinin (*IBM, Burroughs, Rand Cooperation*) görev aldığı *Askeri Hava Savunma Projesi SAGE (The Semi Automatic Ground Environment)* yeni yaklaşımı zorlayan bir geliştirilme süreci olarak tanımlanabilir.

Beş yıldan çok süren bu proje içinde; donanım, iletişim ve yazılım geliştirme çalışmaları ve *Gerçek Zamanlı (Real Time)* işletim tabanlı geliştirilen: iki "*en büyük*" bilgisayar (ağırlığı 270 ton ve kullanılan vakum tüp sayısı 60.000 ve mühendis ve yazılımcı desteği ise 3.000) sisteminin geliştirilme çabalarının, dünya bilgisayar sistemlerinin ve yazılım geliştirme ve kuramsal yönlendirmeye katkısı büyüktür [1][5][7].

bilgisayar programcısı sayılan bayan Lady Ada Lovelace'in (1815-1852) ismini taşır.

#### 4. Yazılım Krizi

1965-1980 yılları üniversite ve iş alanlarında güçlü *Bilgisayar Merkezleri (Computer Centers)*'in kurulduğu, bilimsel ve iş alanı uygulamalarının ayrıldığı ve özellikle kapsamlı iş alanı uygulamalarının bilgisayar sistemlerine uyarlanmaya başladığı, *Yazılım Evlerinin (Software Houses)* yavaşta olsa geliştiği yıllar olarak tanımlanabilir. Başta IBM olmak üzere, yazılımın kiralanabilmesi ya da satın alınması döneminin başladığı bu yılların bir özelliği; uygulamaların ve özel koşullarına yönelik uygulama problemlerine ve özellikle "*Büyük Uygulama Projelerine*" bütüncü ortak çözüm arama girişimi görülmeye başlamış önemli bir dönemdir [5].

1970'li yılların ikinci yarısından başlayan, yazılım geliştirme sayısında ve boyutunda sıçrama döneminden söz edilebilirse de; büyük-proje konusunda gerek yazılım evlerinin, gerekse kurulu Bilgisayar Merkezleri uzmanlarının çözümlenme, tasarım, programlama gibi temel aşamalarda karşılaştığı sorunlar nedeni ile uygulamayı öngörülen sürede ve maliyetle tamamlayamamaktan doğan, maddi ve manevi zararlar başta iş sahiplerini olduğu kadar, çalışanları da güç durumda bırakmıştır. Bu süreç, kaçınılmaz olarak; "*Yazılım Bunalımı (Software Crisis)*" adı verilen bir bunalım oluşumu yaratmıştır. Olumsuzluğun en önemli yanı; bilgisayar uygulamasına binlerce hatta milyonlarca ABD doları yatıran yöneticilerin yatırımlarının karşılığında hayal kırıklığı olmuştur. Sorun, donanımdan çok, yazılım tasarımının başarısızlığından ya da denenilen çözümlerin ya da deneyimin yetersiz olmasından kaynaklandığı çok açıktı.

Üniversite ve iş alanlarında güçlü *Bilgisayar Merkezleri (Computer Centers)*'in kurulduğu, bilimsel ve iş alanı uygulamalarının ortamlarının ayrıldığı ve özellikle kapsamlı iş alanı uygulamalarının bilgisayar sistemlerine uyarlanmaya başladığı yıllarda, "*Yazılım Bunalımı*"nın kendini gösterdiği en önemli örnek, özellikle değindiğimiz "*büyük*" Askeri projelerdi.

Sorun "*Büyük*" olmaktan mı? Kaynaklanıyordu. Diğer taraftan, üçüncü kuşak bilgisayarların hızla ortaya çıkması da, programlama güçlükleri yanında yeterli sayı ve nitelikte uzman programcı bulamama sorunlarını yaratmıştır [3][7][11].

#### 5. Yazılım Bunalımına Çözüm

"*Bunalımın*" gerek bilgisayar bilimleri ve mühendisliği, gerekse bilgisayar teknolojisi ve uygulaması alanında "*yöntemsel*" bir kırılma yaratacak çözüm aramaya yol açtığı söylenebilir. Tarihsel sürece baktığımızda, kişisel çabalar yanında, bunalıma çözüm arama konusunda bütüncül çabanın Avrupa'dan geldiğini görüyoruz İlk ve ses getirici uluslararası girişim ve çalışma; 1968 Yılında Almanya'nın Garmish-Partenkirchen kentinde, NATO' Bilim Kurulunun desteklediği *Münih Teknik Üniversitesi (München Technischen Universität)*'nin ev-sahipliğinde; *İnformatik Enstitüsü Başkanı Bilgisayar Bilimcisi Prof. Dr. Friedrich L. Bauer* yönetimindeki konferanstır. Konferansa, 50 matematikçi, mühendis ve bilgisayar bilimcisi katılmış ve "*Büyük Yazılım Projesi*" geliştirme ve yöntemleri konusu ele alıp, çözümler tartışılmıştır [3]. Bu bağlamda, araştırma ve uygulamacıların çözüm önerileri ışığında, büyük yazılım projelerinin yazılım çözümlenme, tasarım ve yönetimi için, kuramsal farklı yaklaşımlara gerek olduğunu ortaya konmuş, tartışılmış ve sonucunda, özetle konunun *Yazılım Mühendisliği (Software Engineering)* adı altında özel ve yeni bir disiplin olarak geliştirilmesi kararı alınmıştır [2][3][9]. Böylece, yazılım bunalımına çözüm getirecek Yazılım Mühendisliğinin doğumu bu konferans olarak tarihe geçmiştir<sup>3</sup>.

Bu konuda, Münih Teknik Üniversitesi "*İnformatik-Bilişim*" Enstitüsünün başındaki bilgisayar bilimlerinde birçok başlangıca imza atmış, değerli *Bilgisayar Bilimcisi Prof. Dr. Friedrich L. Bauer* 'ın katkısına ve önderliğine özel bir yer vermek zorunluğu kaçınılmaz görünmektedir.

Bauer, 1967/1968 çalışmaları; "...Yazılımın tasarım, geliştirme ve uygulamasında eksikliklerin ve darboğazların ortaya konmasında ve çözüm yolları aranmasına zorunluluk getiren yeni bir bilgisayar bilimleri alt disiplininin, konuyu öne çıkaran bir isimle ortaya koyuyor. Bir yönü ile, Yazılım Mühendisliğinin isim babası "Fiedrich. L. Bauer" in sürükleyici ve birleştirici katkısı olmasaydı, çözüm bulma hızı o gün gerçekleştiği kadar büyük olamazdı [3].

Bauer'in Yazılım Mühendisliğinin doğuşundan başlayan görüşleri Bilgisayar Bilimlerine ve özellikle Bilgisayar Mühendisliğine çok önemli bir alt-dal ve "proje düzeyindeki" uygulamalarının başarısını sağlayan yeni bir ufuk ve kuramsal bir yol kaçmakla kalmamış, bu gelişimden sonra bilgisayarların donanım yapılanmasını değiştirip, yeni gelişmelere götüren boyutlar kazanmasını da zorunlu kılmıştır. Temel etmen ise, yazılım geliştirmeyi, yeni bir disiplin (New Methodology) olarak ortaya koymuş olmasıdır [2][3].

Birinci konferanstan sonra, 1969'da yapılan Roma Konferansı ilki kadar etki yaratacak boyutta olmamasına karşın, her iki yakada da çalışmalar süre gelmektedir<sup>7</sup>. Diğer taraftan, Prof. Bauer ve yazılımcı arkadaşları Yazılım Mühendisliğini geliştirmek gereksinimlerine yönetsel yanıt bulmak ve yaymak için gerek NATO desteğinde, gerekse, Üniversite desteğinde davetli konuşmacıların yönetim ve katkıları ile çeşitli ülkelerden seçilmiş bilim adamı ve uzmanlara *Yaz Okulları (Summer Schools)* düzenlemeyi sürdürmüşlerdir.

Örneğin: 1972 Yılında, NATO Bursu ile katıldığım yine, Münih Teknik Üniversitesinden Prof. Dr. Baure'in yönetiminde düzenlenen "*İleri Yazılım Mühendisliği*" (*Advanced Software Engineering*)

<sup>7</sup> 2000 li yıllarsa Yazılım Mühendisliği disiplinin kuramsal gelişimi Bilgisayar Mühendisliği içine sığmamaya başlamış ve kimi inivermelerde ayrı bir mühendislik dalı olarak bölümlerin açılmasına neden olmuştur. Bu bağlamda, kimi akademisyenler;

yaz okulunda verilen derslerin bütünleştiği kitabın ön sözünde: Yazılım Mühendisliğinin geçirdiği evreyi özetlerken, Yazılım Mühendisliğinin geleceği konusundaki görüşlerini Bauer, [1] kısaca şöyle aktarıyor:

"... 1967 ve 1968'lerde, "Yazılım Mühendisliği" çarpıcı sözcüğü; yazılımın tasarım, geliştirme ve kullanılmasında "kışkırtıcı" bir anlam yüklenerek ortaya atıldı ve kullanıldı... O tarihlerden bugüne, durum o denli değişti ki, yazılım üreticileri Yazılım Mühendisliği temel kurallarına boyun eğdiler, ve kısa zamanda Yazılım Mühendisliği özendirilir olmaya dönüştü.....

Bir bakıma, Yazılım Mühendisliğinin zaman içinde kendini kısmen savunma izlemine yöneldiği de söylenebilir. Ama öyle umuyorum ki, bu yaklaşım bir gün tersine dönecek ve öyle umuyorum ki, bir gün Yazılım Mühendisliği öngörülerini bilgisayarların nasıl tasarlanacağı ve kullanılacağına belirleyicisi olacaktır.

Gerçekte yapmak istediğimiz, geleceğimizin ortamını oluşturmaktır. Diğer taraftan, yazılım krizini çözmedeki başarısızlık, günümüzün bilgisayarları ile bağımlı çalışan bilim insanlarını çıkmaza sürüklediği söylenebilir ve hatta uluslararası büyük projelerle, zengin ülkelerin bilim ve ekonomilerini de "Büyük Bilim" (Big Science) gibi eksi yönde etkileyebilir..." [1][2].

Bu gelişme belki de, Bilgisayar Mühendisliğinin "Yazılım Mühendisliği" bağlamındaki kapsam, içerik ağırlığının her yıl güncellenmesini getirdiği de söylenebilir.

Bilgisayar ve Bilgisayar Bilimlerinin genişleyen yöntem ve bulgularının, dün olduğu gibi bugün de, uygulamaya taşan boyutları ile sürekli olarak endüstri, iş alanları ve kısaca toplumsal yaşam ve

"Yazılım geliştirme ve tasarım bir parça sanat ve bir parça bilim gibi görünüyor fakat hangi tarafının daha ağırlıklı olduğunu hangisi kazanacak? Gibi tartışmaları Twitter, Facebook sayfalarında bile görüyoruz. @sdtimes: on Twitter|on Facebook.

uygulama biçimlerini değiştiren oluşumlar yaratmayı sürükler olduğu kesindir. Yaşamı yeniden yapılanmaya götüren kuramsal süreçler, özelleşerek yeni mühendislik dallarının doğmasını zorunlu kılmaktadır. Bu gün bu kaynaktan doğan 10'a yakın mühendislik dalı doğduğu sayılabilir. İşte bunlardan ilkinin, her gün yeni yazılım mimarileri ile büyüyen *Yazılım Mühendisliği* olgusu olduğunu söylemek yanlış olmayacaktır...

## Kaynakça

- [1] Bauer F.L., "Advanced Course on Software Engineering", Springer-Verlag Heidelberg, Newyork, 1973.
- [2] Bauer,F.L., "Software Engineering", Information Processing, 71, 1972, NATO Science Committee, Oct. 7-11,Brussels, Scientific Affairs, 1968.
- [3] Bauer F. L.,Editors: Naur P.,Randell B., "Software Engineering" Report on Conferance, NATO Sciece Committee, Garmisch, Germany,7th to 11th October, 1968.
- [4] Bergin,T.,J. and R., G. Gibson, "History of Programming Languages-II.",New York, ACM Press, 1996.
- [5] Brennecke A.,Keil,R., (Editors), "History of Software Engineering", Position Papers, University West Lafeyetei,USA, 1997.
- [6] Capron H.L. and B.,K.,Williams, "Computers and Data Processing",The Benjamin/Cumming Pub.Co. Inc. California,1982.
- [7] Grier,D.,A.,"Software Engineering: History", Center for International Science and Technology Policy, George Washington University, Washington, District of Columbia, U.S.A.,2010.
- [8] Glass R.,L., "Foundations of Software Engineering: An Early History of Software Engineering", Ken Anderson, 1999.
- [9] Naur F., and Randell B., "Software Engineering",Report on a Conferance, Germish, October 1968.
- [10] Randell B.,"The 1968/69 NATO Software Engineering Reports", Schloss Dagstuhl August 26 - 30, 1996.
- [11] Wirth N., "A Brief History of Software Engineering", July–September, ETH, Zurich, 2008.Published by the IEEE Computer Society, 2008.