



**Ahmet Hamdi
ATALAY**

Akgün Bilgisayar Program ve Hizmetleri Sanayii Ticaret A.Ş.
Yönetim Kurulu Başkan Vekili • aatalay@akgun.com.tr • www.akgun.com.tr

YAZILIM VE OYUN DEĞİŞTİREN YÖNELİMLER

2020'ler itibariyle artık mevcut yöntem ve iş modelleri pazarın ihtiyaç ve taleplerine cevap vermekte zorlanıyor. Yazılım geliştirme süreçlerinin daha etkin ve verimli olması, hızlı ve hatasız üretim, maliyet etkin çözümler, daha akıllı ve güvenli sistemler ile uygulamaların hizmet olarak sunulması (as a service) gibi beklentiler artıyor. Bu dönemde Hizmet Odaklı Mimari, Microservice gibi yaklaşımlar ve Yapay Zekâ teknolojileri öne çıkmaya başlamış durumda. Yapay Zekanın hayatın her alanına girmesi ve sektörleri dönüştürmeye başlaması yeni bir dönemin başlangıcı anlamına geliyor.

20. yüzyılın ilk yarısında insanlığın hayatına girmeye başlayan Bilgisayarlar (Computer), insan emeği ve dikkati ile istenen hız ve kalitede işlenmesi zor olan çok sayıda verilerin/bilgilerin sayılmasına (işlenmesine) yarayan makineler olarak ortaya çıkmıştır.

1945'de Pensilvanya Üniversitesinde Mauchly ve Eckert tarafından geliştirilen ENIAC (Electronic Numeric Integrator and Computer) makinası ilk örnek sayılabilir. 1949 yılında Princeton Üniversitesi John von Neuman tarafından geliştirilen EDVAC (Electronic Discrete Variable Computer) makinası ise gerçek anlamda ilk bilgisayar olarak kabul edilir.

Günümüz bilgisayarında, program ve veriler belleğe ikili kodlar (binary digit) olarak yüklenir. 1 veya 0'dan oluşan bu kodların her birine "bit", 8 bitin bir arada olduğu dizine ise "byte" denir. Bilgisayar işlemcisinin (CPU) anlayabileceği tek dil bu 1 ve 0'larda oluşan makine dili (machine code) dir. İnsan tarafından yazılan her programlama dilinin kaynak kodu bir derleyici tarafından makine diline çevrilme zorundadır ki bilgisayar tarafından anlaşılın.

İnsanlar tarafından Bilgisayara yapacağı işi tarif eden komut dizisine ve talimatlara Programlama denir. Makine dilinde programlama zor bir iş olduğundan günümüzde yazılım, kodlamak ya da kod yazmak şeklinde de tanımlanan bu faaliyet için bugüne kadar çok sayıda programlama dili geliştirilmiştir. 1957 yılında IBM tarafından makine dili bilmeden de program yazabilmesini sağlamak üzere ilk programlama dili FORTRAN (FOR-

mula TRANslating) geliştirildi. O yıllarda Bilgisayarlar mühendislik işlemleri için ve mühendisler tarafından kullanılacak bir araç olarak kabul edildiğinden FORTRAN bir mühendislik diliydi. Daha sonra gelen ALGOL, PL1, PASCAL, C, ADA vb. diller FORTRAN soyundan gelirler.

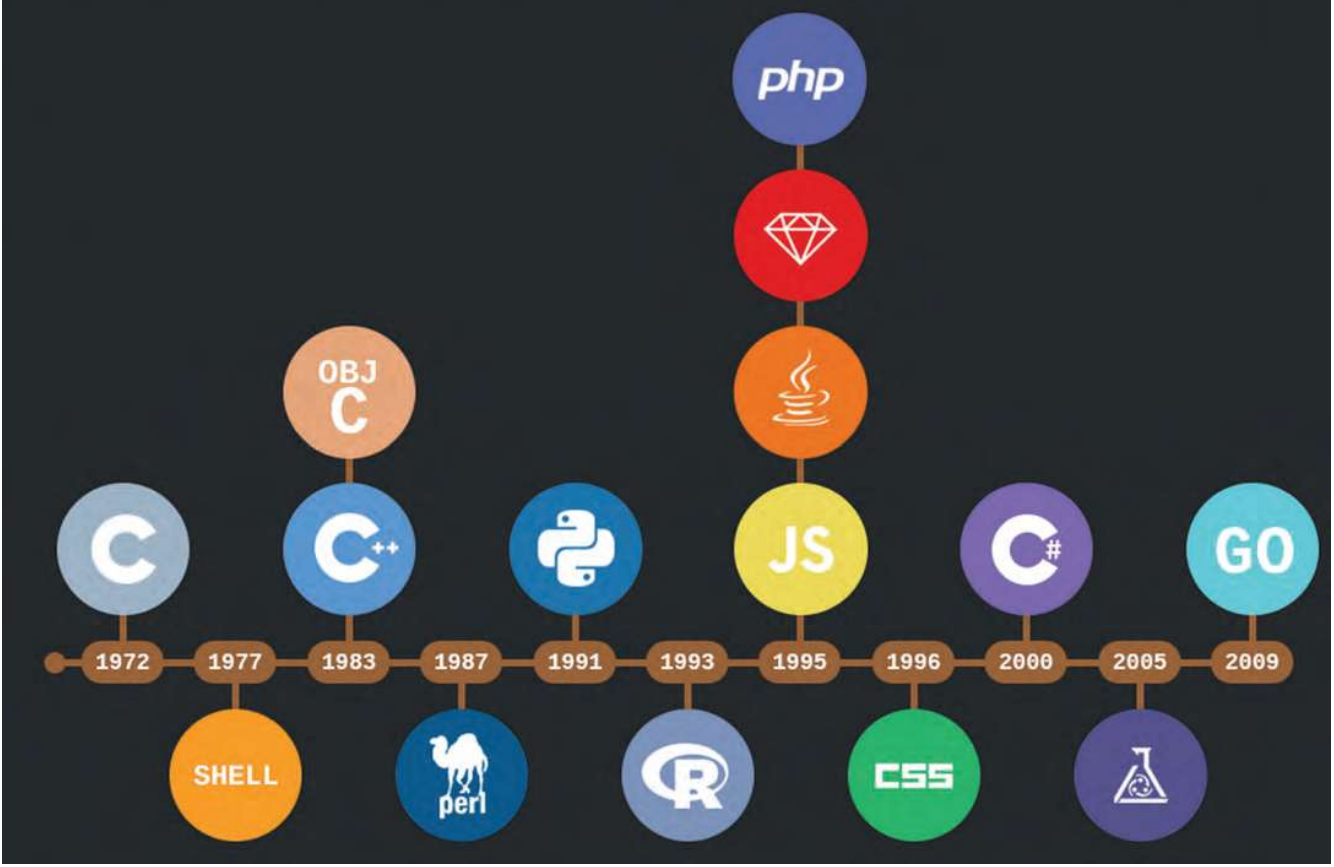
1990'larda Bilişim Teknolojilerinin ve İnternetin yaygınlaşmasıyla birlikte köklü değişimler meydana geldi. Bilgisayarlar iş, eğlence, ticaret, bankacılık, eğitim vb. pek çok alanda günlük hayatın vazgeçilmez aracı haline geldi. İş yerlerinde ve evlerde herkesin bir kişisel bilgisayarı (Personal Computer) oldu. Bu durum yazılım ihtiyacını olağan üstü arttırıp çeşitlendirdi.

Yazılımın Altın Dönemi

2000'lerde ise, Mobil haberleşme teknolojilerinin yaygınlaşması ve akıllı telefonların ortaya çıkması ile Bilgisayarlar herkesin cebine girmiş oldu. Yeryüzündeki insanların üçte ikisi Bilgisayar veya Cep telefonu sahibi olurken her iki insandan biri İnternete bağlanabilir hale geldi. Kişisel bilgisayarlardan yapılan tüm iş ve işlemlerin mobil cihazlardan da yapılabilir kılınması talebi mobil uygulamalar pazarını oluşturdu. Mobil uygulama geliştirmeyi mümkün kılan Python, PHP ve Node.js gibi yeni nesil yazılım dilleri popüler hale geldi. Farklı uygulamalar için farklı özelliklere sahip çok sayıda programlama dili ortaya çıktı.

2010'lerden itibaren, sürekli değişen ve gelişen teknolojiler ve ona uygun yazılımlar, akıllı uygulamalar, mobil uygulamalar, oyunlar, iş uygulamaları, endüstri





uygulamaları vb. ile yazılım dünyasında adeta patlama yaşıyor. Artık internet-siz bir hayat, bilgisayarsız bir dünya düşünülüyor. Milyarlarca insanın cebindeki akıllı mobil cihazlar otuz yıl öncesinin süper bilgisayarlarından daha güçlü ve daha yüksek kapasiteli. Teknolojik gelişmeler, ihtiyaç ve talepleri sürekli artırıyor ancak buna cevap verebilecek insan kaynağı kapasitesi oluşturulamıyor.

Bir yandan Siber Güvenlik riskleri diğer yandan Bulut Bilişim ve Açık Kaynak yazılımların yaygınlaşması gibi gelişmeler pazarı evrilmeye zorluyor.

Yapay Zekâ Çağı

2020'ler itibarıyla artık mevcut yöntem ve iş modelleri pazarın ihtiyaç ve taleplerine cevap vermekte zorlanıyor. Yazılım geliştirme süreçlerinin daha etkin ve verimli olması, hızlı ve hatasız üretim, maliyet etkin çözümler, daha akıllı ve güvenli sistemler ile uygulamaların hizmet olarak sunulması (as a service) gibi beklentiler artıyor.

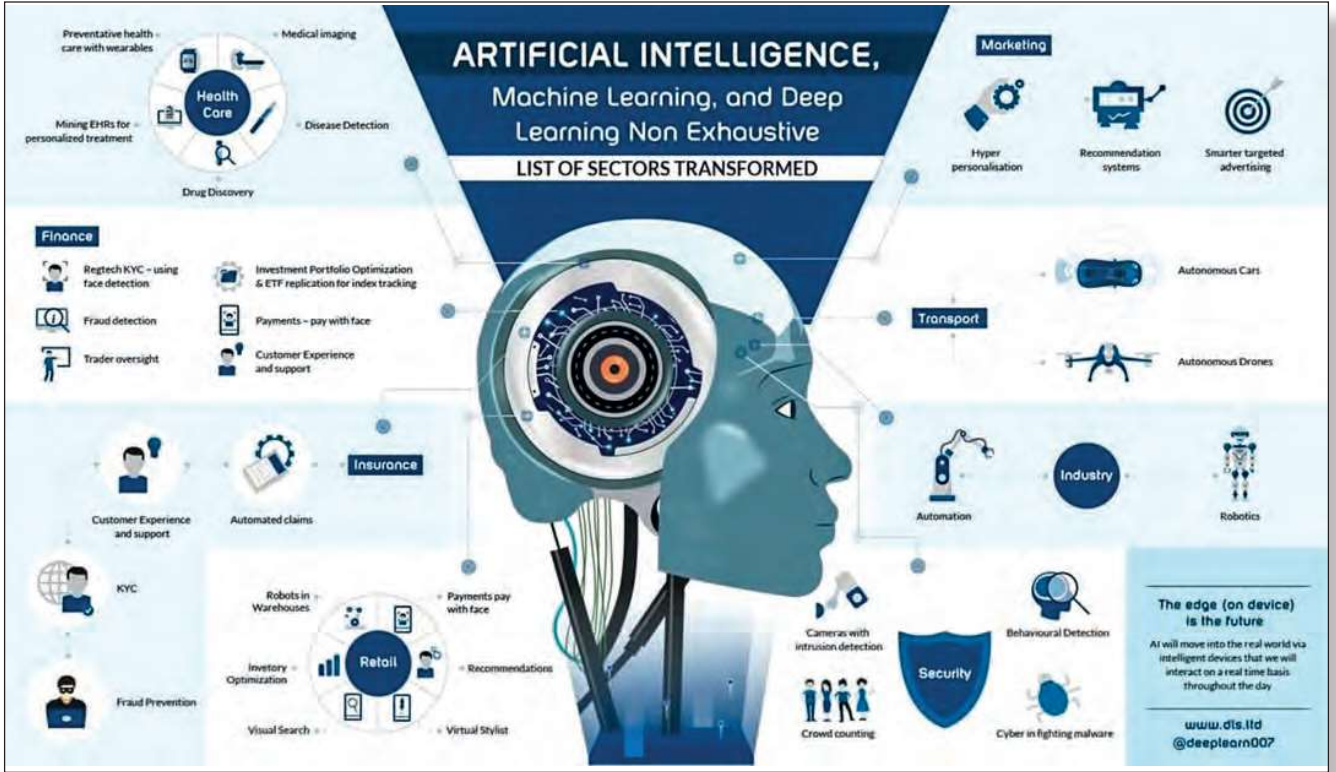
Bu dönemde Hizmet Odaklı Mimari (SOA), **Microservice** gibi yaklaşımlar ve **Yapay Zekâ** teknolojileri öne çıkmaya

Size of programming language communities in Q3 2022

Active software developers, globally, in millions

		Most popular in	Least popular in
JavaScript*	19.6 M	Apps for 3rd-party ecosystems, Cloud	DS/ML/AI, Embedded
Python	16.9 M	DS/ML/AI, IoT apps	Web, Mobile
Java	16.5 M	Cloud, Desktop	Web, DS/ML/AI
C/C++	12.3 M	Embedded, IoT apps	Cloud, Web
C#	10.6 M	Desktop, Games	DS/ML/AI, IoT devices
PHP	8.9 M	Web, Cloud	Mobile, DS/ML/AI
Kotlin	6.1 M	Mobile, AR/VR	Games, DS/ML/AI
Visual development tools	4.9 M	AR/VR, Games	Embedded, Cloud
Swift	4.2 M	Mobile, AR/VR	Embedded, Cloud
Go	3.8 M	Apps for 3rd-party ecosystems, Cloud	Mobile, DS/ML/AI
Objective C	3.0 M	AR/VR, IoT devices	Desktop, Apps for 3rd-party ecosystems
Rust	2.8 M	AR/VR, IoT apps	Mobile, Web
Ruby	2.4 M	IoT devices, Apps for 3rd-party ecosystems	Embedded, Web
Dart	1.9 M	Mobile, Apps for 3rd-party ecosystems	Web
Lua	1.9 M	IoT devices, AR/VR	Mobile, Embedded

DATA



başlamış durumda. **Makine öğrenimi** ve **Derin öğrenme** algoritmalarının yazılım geliştirmede ve iş akışlarını optimize etmede kullanılması daha akıllı uygulamalar geliştirilmesine imkân sağlamaktadır.

Yapay Zekanın hayatın her alanına girmesi ve sektörleri dönüştürmeye başlaması yeni bir dönemin başlangıcı anlamına geliyor.

2022'nin sonlarında Teknoloji dünyası bir Yapay Zekâ, Büyük Dil Modeli (LLM)

uygulaması olan “ChatGPT” ile çalkalandı. ChatGPT, Chat (Sohbet) kelimesi ile GPT (Generative Pre-Trained Transformer) “Üretken Ön Eğitilmiş Dönüştürücü” ifadesinin birleşiminden adını alan ve çok çeşitli konularda kullanıcılarla sohbet edebilen bir robotudur. Bir makine öğrenimi algoritması kullanan bu sohbet robotu, internetteki üzerinden topladığı bilgileri sorulan soruya yanıt olarak bir araya getirip kullanıcıya sunuyor. Bunun yanında;

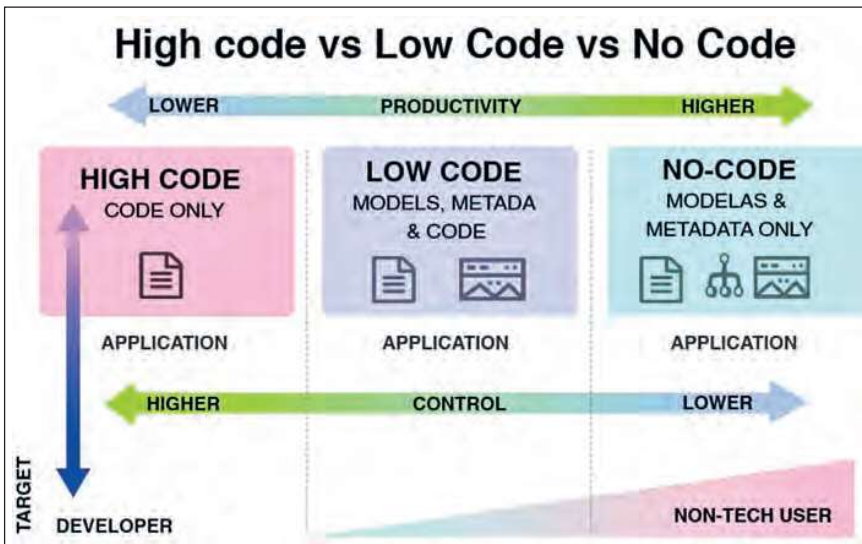
- Sizinle her konuda **sohbet edebiliyor**,
- İstedığınız bir konuda (bilimsel makale, masal, şiir, roman vb.) **yazı yazabiliyor**,
- Çeşitli dillerde **tercüme yapabiliyor**,
- İstedığınız işi yapabilecek programlama için **Kod yazabiliyor**.

OpenAI tarafından geliştirilen ChatGPT yanında Apple, Google, Microsoft ve Meta firmaları da kendi **Üretken Yapay Zekâ** çözümlerini pazara sunma aşamasında.

Kod Yazmadan Yazılım Geliştirme Dönemi

Artık robotların kod yazdığı, uygulama geliştirdiği bir aşamaya geldik. Gerek kaynak ve kapasite açığını giderip kıt kaynak olan mevcut insan kaynakları ile daha fazlasını yapabilmek ve gerekse insan kaynaklı hataları ortadan kaldırmak için yazılım alanında çığır açacak bu gelişme yakın dönemde teknoloji dünyasında önemli değişim ve dönüşümlere yol açacaktır.

Yazılım ve uygulamalar, geliştirme yöntemlerine göre 3 ayrı kategoriye ayrılabilir.





araç veya IDE (Entegre Geliştirme Ortamı) tarafından oluşturulur.

Düşük kodlama gerektiren yöntemle yazılım geliştirilmesi, daha az emek gerektirmesi, daha az insan kaynağına ihtiyaç duyulması, daha hızlı ve daha az maliyetli geliştirmeye imkân tanınması, insan kaynaklı hataların daha az olup daha güvenli yazılımları mümkün kılması gibi nedenlerle popülerliği hızla artmaktadır.

Microsoft, Google, Oracle, SAP gibi büyük teknoloji firmaları, düşük kod kervana katılarak ürün portföylerine düşük kod platform ve araçlarını konumlandırıp sunmaya başlamışlardır.

Uygulanması giderek yaygınlaşmakta olan ve faydaları daha da belirginleşen Düşük Kodlu yazılım geliştirme yöntemi ile sağlanan faydalar;

Hızlı Geliştirme: Düşük Kodlu platform ve araçları ile geleneksel yöntemlere göre yazılım geliştirme ve teslim sürelerinde %50'ye varan iyileşme sağlanabilmektedir. Daha az geliştirme süresi ve daha az kaynak tüketimi, toplam sahip olma maliyetlerinde azalması demektir.

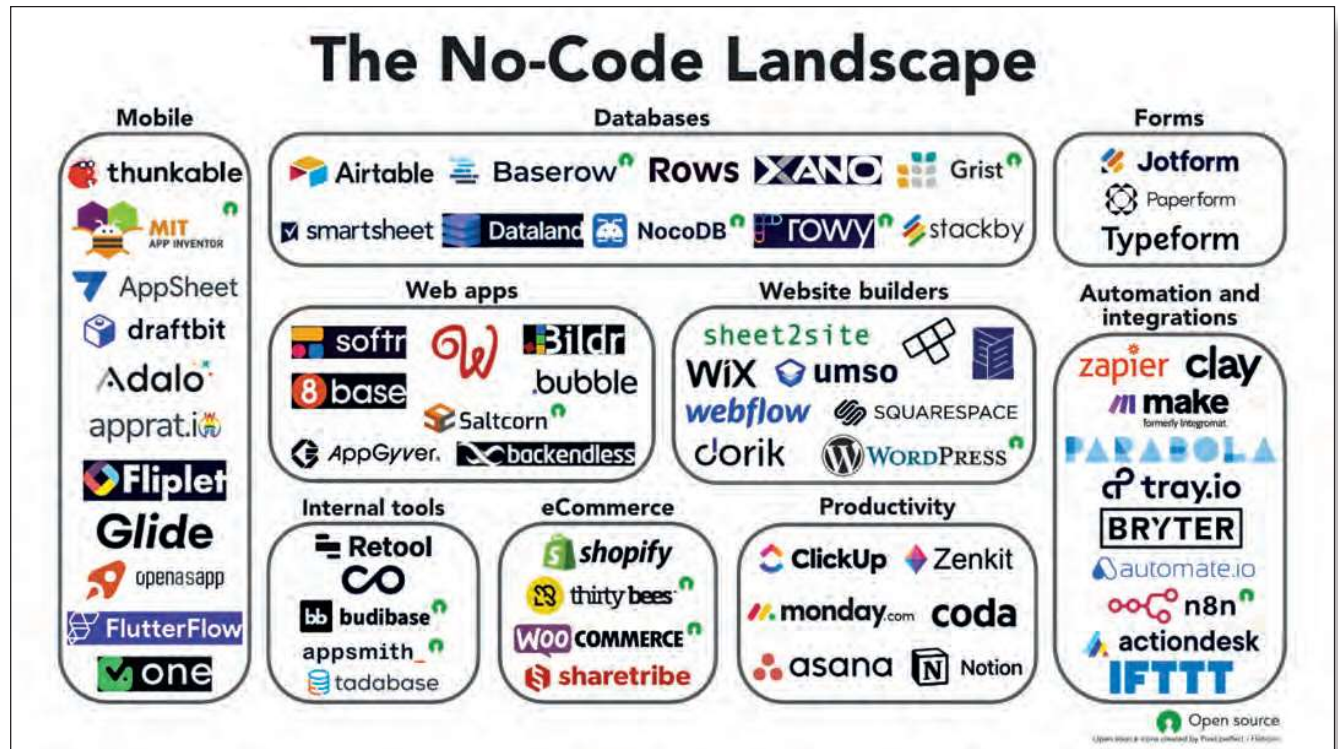
Kaliteli Kodlama: Hızlı yazılım teslimi baskısı genellikle kodun kalitesinin bozulmasına bu da yazılımın kullanılabil-

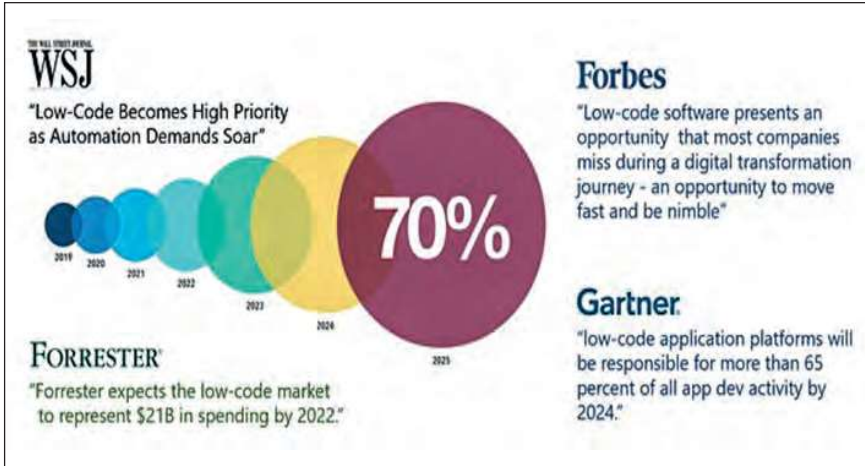
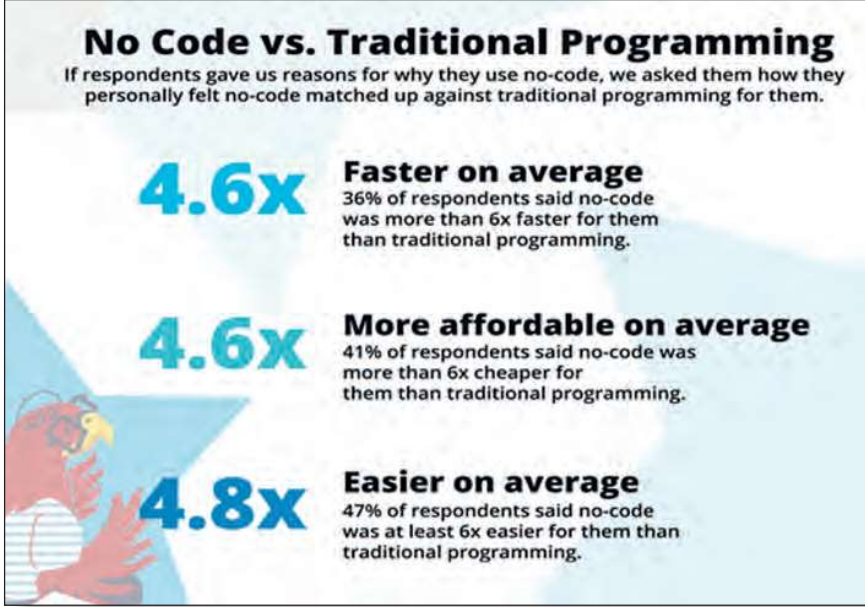
- Yazılımın tamamının kodlanmış geleneksel yöntem (**High-Code**)
- Yazılımın önemli bir kısmının hazır, az bir kısmının kodlandığı (**Low-Code**)
- Yazılımın kodlamadan, hazır parçalar bir araya getirilerek oluşturulduğu (**No-Code**)

Tüm bu programlama yöntemlerinin birbirine göre çeşitli üstünlükleri vardır. Bazı uygulamalar için bazı yöntemler diğerlerine göre daha avantajlı olabilmektedir.

Uygulamanın sıfırdan kodlanması (geleneksel yazılım geliştirme) esneklik, müşteri isterilerine uyumluluk vb. açılardan avantajlı olabilmekle birlikte zahmetli ve emek yoğun olması, daha uzun zaman alıp daha maliyetli olması, insan hatalarına daha açık olması vb. nedenlerle giderek tercih edilen bir yöntem olmaktan çıkmaktadır.

Düşük kodlu (**Low-Code**) yazılım geliştirme, yazılımın kodlanması, hata ayıklaması, testi ve dağıtımını otomatikleştirmektedir. Düşük kodlu geliştirmede yazılım, bir platform, düşük kodlu bir





lirliği, güvenilirliği ve performansının olumsuz etkilenmesine neden olur. Daha Kaliteli Kodlama ile müşteri memnuniyetsizliği, müşteri kaybı ya da tekrarlı işler nedeniyle maliyet artışları önlenmektedir.

Güvenlik Kod: Siber tehditlerin her geçen gün arttığı günümüzde veri güvenliği kurumlar için en hassas konuların başında gelmektedir. Düşük kodlu platformlar, veri güvenliği ve bütünlüğü konularında en yüksek standartlara uyumu sağlar.

Endüstri standardı ve en iyi uygulamalar: Düşük kodlu platformlar, endüstri standartlarını ve en iyi uygulamaları takip eder, yüksek kaliteli kod üretir ve endüstrinin tavsiye ettiği kullanıcı deneyimlerini oluşturur.

Esnek ve Genişletilebilir: Düşük kodlu

platformları esnek ve genişletilebilirdir. Geliştiriciler bu sayede yazılımın işlevselliğini genişletmek için kolayca paylaşılabilen ve düşük kodlu platformlara takılabilen modüller oluşturabilir.

Üçüncü Taraf API'lerle Entegrasyon: Düşük kodlu platformları, geliştiricilerin herhangi bir kod yazmadan kullanıcı arayüzleri aracılığıyla API'leri kolayca yapılandırarak üçüncü taraf API'lerle entegre olmasına olanak tanır.

Otomatik Dağıtım ve Test: Düşük kodlu platformlar, daha hızlı yazılım teslim döngüleriyle ilgili olan test ve dağıtım sürecini otomatikleştirir.

Kodsuz geliştirme platformları (NCDP): No-Code Development Platform), geleneksel kod yazarak yapılan bilgisayar programlama yerine hiç kod yazmadan

grafik kullanıcı arayüzleri ve konfigürasyon araçları yazılım oluşturmaya imkan sağlarlar. NCDP'lar, uygulama geliştirme sürecini hızlandırmak için tasarlandığından, düşük kodlu geliştirme platformlarıyla da yakından ilişkilidir. Bununla birlikte, düşük kodludan farklı olarak, kodsuz geliştirme platformları hiçbir kod yazımı gerektirmez ve genellikle işletmelerin uygulama oluşturabileceği önceden oluşturulmuş şablonlar sunar.

No-code platformlar, daha önce yalnızca yazılım uzmanlarının yapabileceği karmaşık bir web sitesi, mobil uygulama oluşturmasını ya da tekrarlayan görevleri otomatikleştirmeyi bu konuda uzman olmayan insanların da yapabilmesine olanak tanır.

Son dönemde yazılım alanında yaşanan yetkin insan kaynağı eksikliği ve iş gücünde yaşanan şirketler hatta ülkeler arası hareketlilik yazılım şirketlerini sınırlı sayıda yetkin yazılımcı ile iş yapmak zorunda bırakmıştır. Bu yeni durum bu tür platformların popülaritesi arttırmış ve ve çok sayıda firmayı bu alanda çözüm geliştirmeye yöneltmiştir.

Bu araçlar kullanılarak yazılım geliştirme zaman ve maliyet azaltımı sağlamanın yanında, yazılım geliştirmeyi de oldukça kolaylaştırmaktadır. Yapılan çeşitli araştırmalara göre Kodsuz yazılım geliştirme yöntemi, geleneksel kod yazarak yazılım geliştirme yöntemine göre 6 kata kadar daha hızlı, 6 kata kadar daha ucuz ve 6 kata kadar daha kolaylıkla yazılım geliştirmesini mümkün olabilmektedir.

Wall Street Journal'a göre, düşük kodlu, kodsuz platformlar 2025 yılına kadar %70 büyüyebilecek. Ayrıca Gartner, Düşük kodlu platformların 2024 yılına kadar tüm uygulama geliştirme faaliyetlerinin %65'inden fazlasından sorumlu olacağını öngörüyor.

Kaynakça:

- <https://www.tildevtech.com/history-of-computer-programming/>
- <https://cbddo.gov.tr/SharedFolderServer/Genel/21.Chatbot-Uygulamas%C4%B1-ve-ChatGPT-%C3%96rne%C4%9Fi-De%C4%9Ferlendirme-Raporu.pdf>
- https://en.wikipedia.org/wiki/No-code_development_platform