

Yazılım Mühendisliğine (YM) Giriş

2026 GÜZ

İyi bir YM Nasıl Olunur?

1. İyi bir YM olmak kesinlikle sadece kodlama yapmak değildir.

- ❑ C++, JavaScript ya da Python gibi güncel programlardan birini bilmek yeterli değildir.
- ❑ YM, çalıştığı ekiple iyi ilişkiler kurmalı ve empati yapabilmelidir.

Sonuç: Kullanıcının (user) / müşterinin (customer) geliştirilen ürünle (product) etkileşimi artacaktır.

İyi bir YM Nasıl Olunur?

2. YM fikir sahibi olarak güçlü tartışma becerilerini aktarmalıdır.

- ❑ Karşılıklı iletişim sonunda ikna edebilme gücü ön plana çıkar.
- ❑ Takım çalışmasında gelişebilmek önemlidir.
 - ❖ Farklı fikirlere karşı kişi kendini savunabilmelidir.
- ❑ Değişen öncelikler ve gereksinimlerde, acil işlerde işbirlikçi olmak ve olumlu yaklaşımlarla iletişim ön plandadır.

İyi bir YM Nasıl Olunur?

3. Gerçek dünyada yazılım geliştirme, eğitim amaçlı kodlamadan çok farklıdır.

- ❑ Kurumsal uygulamalar, sürekli ve kesintisiz çalışmalı ve süreç boyunca güvenilir olmalıdır.
 - ❖ Kodun performansı ve sürekliliği, kullanıcı ile daima iletişimde olarak problemlerin anında düzeltilebilmesi ile sağlanır.
- ❑ Ürün tasarlanırken gelecekteki büyüme (ölçeklenebilirlik) düşünülmelidir.

İyi bir YM Nasıl Olunur?

3. devam:

- ❑ Yazılım ürünü Planlama, Analiz, Tasarım, Uygulama, Test ve Entegrasyon ve Bakım gibi farklı aşamalar sonunda geliştirilir.
- ❖ Etkin /güçlü bir yazılım geliştirme döngüsü sonunda geliştirilen ürün **KALİTE** ölçütlerini sağlayabilir.
 - ✓ Sürecin düzenli olarak **test edilmesi** de kritik önem taşır.

İyi bir YM Nasıl Olunur?

4. Geliştirme sırasında farklı takımlarla işbirliği gerekir.

- ☐ Çalışılan birim dışında bir ekiple de çalışabilmelidir.
 - ❖ Farklı bakış açıları kişinin yeteneklerini geliştirir.
- ☐ Takımlar arası çalışmalarda **açık kaynaklı kodlu** yazılımlar tercih edilebilir.
- ☐ Kişinin bir fikri varsa, hangi birimde olduğunun önemi yoktur, farklı bir takım da yönlendirebilir.
- ☐ Yaratıcılığın değerlendirildiği kişilerle birlikte ürüne katkı sağlamak hedeflenir.

İyi bir YM Nasıl Olunur?

5. YM hangi rolü üstlenirse üstlensin programlama temeli çok güçlü olmalıdır

2011 yılında, teknoloji yatırımcısı Marc Andreessen, büyük bir teknolojik ve ekonomik değişimin yaşandığını ve yazılım şirketlerinin dünya ekonomisinde büyük yol alacağını ifade etmiştir.

https://en.wikipedia.org/wiki/Marc_Andreessen

https://en.wikipedia.org/wiki/Tim_Berners-Lee

<https://www.w3.org/People/Berners-Lee/>

İyi bir YM Nasıl Olunur?

5. devam

- ❑ Bilişim sektörünün tüm alanlarında daha fazla programlama yeteneğine ihtiyaç vardır.
- ❑ YM, profesyonel bir kod yazan (programcı) olmayabilir;
 - ❖ Bu rolü ile yazılımın nasıl çalıştığını ve sektöre kattığı değerleri bilmelidir.

İyi bir YM Nasıl Olunur?

6. Girişimci (entrepreneur) YM olmak bir en istenilendir.

- ❑ Yazılım geliştiricileriyle yani programcılarla ilgili genel düşünce gece gündüz durmaksızın çalıştığıdır. Gerçek ise çok farklıdır.
- ❑ Geliştirici zamanın çoğunu ekip çalışmasıyla işbirliği ve etkileşim halinde geçirir.
 - ❖ Başarılı bir **kariyer gelişiminin** temelidir.
 - ❖ Ortak gerçekleştirilen güçlü çalışmalar **KALİTELİ ürün** ortaya çıkaracaktır.

Sonuç olarak: Yazılım Mühendisliği ve Yazılım Mühendisi

Yazılım mühendisliği kariyer yolunda sürekli mücadele ederek öğrenmektir.

Başarılı **bir yazılım mühendisi** matematik, mühendislik ilkeleri ve programlama becerilerine sahip olmalıdır.

Yazılım Mühendisliği Disiplini

❑ Her yazılım ürünü için başarı kriterleri temel olarak

❖ maliyet (cost)

❖ süre (time)

❖ nitelik (quality)

değişmez üçlüsü ile değerlendirilir.

Kısaca: Yazılım ürünlerinin zamanında, maliyetinde ve nitelikli olarak geliştirmesi temel hedeftir.

NİTELİK / kalite soyut bir sözcük olarak disiplinin temel kriteridir (ölçütüdür).

Yazılım Mühendisliği'nin Ortaya Çıkışı

“*Yazılım Mühendisliği*” terim olarak ilk defa 1968 yılında NATO konferansında [Brian Randell](#) tarafından önerilmiş ve [F.L. Bauer](#) tarafından popüler hale getirilerek yaygınlaştırılmıştır.

Yazılım Mühendisliğinin Kapsamı

□ Yazılım Mühendisliği aşağıdaki süreçleri gerçekleştirmek üzere bilginin işlendiği *geliştirme araçları ve metotlardan* oluşmaktadır.

- ❖ Yazılım Gereksinimleri (Software Requirements)
- ❖ Yazılım Tasarımı (Software Design)
- ❖ Yazılımın Yapılandırılması (Software Implementation)
- ❖ Yazılım Testi (Software test)
- ❖ Yazılımın Sürekliliğinin Sağlanması (Software Maintenance)

yazılım ürünü geliştirilirken izlenen temel adımlardır.

Yazılım Nedir ? (Britannica tanımı)

- ❑ Yazılım bilgisayara neler yapacağını söyleyen direktiflerdir (instructions).
- ❑ Yazılım, bir bilgisayar sisteminin çalışmasıyla ilgili tüm programları, prosedürleri ve rutinleri içerir.
- ❑ Yazılım terimi, bu talimatları donanımdan, yani bir bilgisayar sisteminin fiziksel bileşenlerinden ayırmak için oluşturulmuştur.

Özet olarak yazılım:

- ❑ Bir bilgisayarın donanımını bir görevi yerine getirmesi için yönlendiren bir dizi talimat, program veya yazılım programı olarak adlandırılır.

Yazılımların Sınıflandırması (Britannica)

- ❑ Sistem Yazılımı
- ❑ Uygulama Yazılımı
- ❑ Ağ (network) Yazılımı

Sistem Yazılımı

Bilgisayarın dahili (internal) işleyişini esas olarak bir işletim sistemi aracılığıyla kontrol eder, ayrıca monitörler, yazıcılar ve depolama aygıtları gibi çevre birimlerini de (peripherals) kontrol eder.

İşletim Sistemi (Operating System) Yazılımı

- ❑ Bilgisayar donanımı ve bilgisayar sisteminin çalışmasını sağlar.
- ❑ Bilgisayar yazılımı işletim sistemi, aygıt sürücüler (device drivers), işlevselliği doğrulama araçları (diagnostic tools), bilgisayarı pek çok kişinin kullanılabilmesinin sağlanması, vs içerir.
- ❑ En genel işletim sistemi yazılımları: Windows, Mac OS X, GNU/Linux

Uygulama (Application) Yazılımı

- ❑ Uygulama yazılımı, bilgisayarı kullanıcı tarafından verilen komutları yürütmesi için yönlendirir
- ❑ Uygulama yazılımı bir kullanıcı için verileri işleyen herhangi bir programı içerir.
- ❑ Uygulama yazılımı kelime işlemcileri, elektronik tabloları, veri tabanı yönetimini, stok kontrolü ve maaş bordrosu programlarını ve diğer birçok uygulamayı içerir.

Ağ (Network) Yazılımı

- ❑ Bu sınıflandırmada bilgisayarların bağlı oldukları ağlarla iletişim sağlanır.
- ❑ Ağ yazılımı, bilgisayar ağlarının tasarımı, uygulanması ve işletilmesi ve izlenmesi için kullanılan geniş bir yazılım yelpazesidir.
- ❑ Geleneksel ağlar, yazılım gömülü donanım tabanlıydı.
- ❑ Günümüzde, Yazılım Tanımlı Ağ (Software Defined Network /SDN) iletişimi ile yazılım donanımdan ayrıldı .
 - ❖ Bilgisayar ağının sürekli değişen doğasına daha uyumlu hale getirildi.

Yazılım Mühendisi Roller

Backend developer(engineer): Bir sunucu veya ağ üzerinde bulunan ve kullanıcı arayüzünü yönlendiren yazılımdan sorumludur

Frontend developer(engineer) Kullanıcıların etkileşimde bulunduğu arayüzden sorumludur (bazen web geliştiricileri olarak da adlandırılır)

Fullstack developer(engineer):Hem istemci hem de sunucu tarafında yazılım tasarlar ve yazar

Programlama Araçları (tools)

- ❑ Programlanmış yazılım araçları (tools) programların oluşturulmasını ve korunmasını kolaylaştırır.
- ❑ Bu araçların en önemlilerinden biri derleyicidir (compiler)
- ❑ Derleyici yüksek düzeyde yazılmış C, Java, C# gibi bilgisayar programlarını ikili komutlara dönüştürür

Diğer Yazılım Türü

Gömülü (Embedded) Yazılım

❑ Makinelerin içerisinde küçük bilgisayarlarda çalışan yazılımlardır.

Çamaşır makineleri , arabalar ya da tıbbi aygıtlar örnekleridir. olarak verilebilir.

❑ Bu tür programlar diğerlerine göre daha az kaynağa erişerek çalışır.

❑ Daha az bellek(memory), daha ucuz ağ(networking), daha az işlemci gücü (processor power), daha az depolama alanı (disk storage) kullanılır ve kullanıcı arayüzü (user interface) yoktur.

Nesnelerin İnterneti (Internet of Things / IOT)

- ❑ Gömülü yazılımlarla bilgisayar donanımı küçülür; ağ (networking) teknolojisi ucuzlar ve daha fazla sayıda gömülü bilgisayar birbirine bağlanabilir.
- ❑ Bu yapılanma Internet of Things (Nesnelerin İnterneti) olarak adlandırılmaktadır.

Diğer Yazılım Türü

Sunucu Yazılımı (Server Software)

- ❑ Sunucu programları merkezi olarak yerleştirilmiş büyük makinelerde çalışır.
- ❑ Bankanın milyonlarca müşteri kaydını tutabilmek için bir sunucu veri tabanına ihtiyacı vardır.
- ❑ Müşteri (client) makinesi (masa üstü bilgisayarda çalışan program) herhangi bir müşteri hesabına bir istem (request) gönderebilir.
- ❑ Sunucu program arama yaparak bulunan bilgiyi (data) geri gönderir.

YM ile Doğrudan İlişkili Diğer Disiplin I : Bilgisayar Bilimi (Computer Science)

- ❑ Bilgisayar donanımı ve yazılımının tasarımı ve işlenmesi ile ilgilidir.
- ❑ Bilgisayar teknolojisinin bilim, iş dünyası ve sanata uygulanmasıdır.
 - ❖ Bilişsel bilim (cognitive science), ekonomi, matematik, fizik (kuantum hesaplama) ve dilbilim (linguistics) uygulamaları ile ilgilenir.

Bilgisayar Bilimi ve Yazılım Mühendisliği İlişkisi

David Parnas tanımları

- ❑ **Bilgisayar Biliminin** temel ilkesi genel olarak hesaplamanın (computation) özelliklerinin incelenmesidir.
- ❑ **Yazılım Mühendisliğinin** temel ilkesi hedeflenenleri başarabilmek için özel hesaplamaların (**specific computations**) gerçekleştirilmesidir.

Bilgisayar Bilimi (wikipedia tanımı)

- ❑ Bilgisayar Bilimi (**computer/computing science**) **bilginin** (information) ve **hesaplamanın** (computation) **teorik temelleri** incelenir.
- ❑ İlgili pratik tekniklerin kullanılması ile bilgisayar sistemlerinin **implementasyonunu** (çalıştırılmasını) ve **uygulamalarını** gerçekleştirir.
- ❑ Genellikle **algoritmik proseslerin** sistematik olarak incelenmesi şeklindedir.
- ❑ Bilginin oluşturulması, betimlenmesi, işlenmesi ve transformasyonudur.

Bilgisayar Biliminin Geçmişi

- ❑ Genç bir disiplin olarak 1940'lı yıllarda görülmeye başlamıştır.
- ❑ Teorik çalışmalar, deneysel yöntemler ve mühendislik tasarımları tek bir disiplinde toplanmıştır.
- ❑ Pek çok fiziksel bilim dalından ayrılır.
- ❑ Bilimi anlamak (bilmek), bilimin uygulamalarından ayrılır.

Bilgisayar Bilimi (Computer Science)

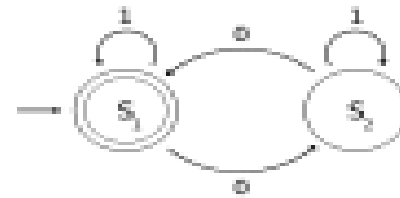
Çalışma Alanları

- ❑ Teorik bilgisayar bilimi
- ❑ Hesaplama Teorisi (Theory of Computation)
- ❑ Algoritmalar ve veri yapıları
- ❑ Programlama metotları ve programlama dilleri
- ❑ Bilgisayar parçaları ve mimarisi
- ❑ Sayısal ve sembolik hesaplama
- ❑ Uygulamalar

Teorik Bilgisayar Bilimi

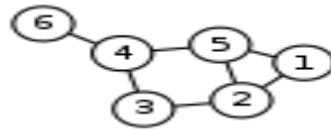
❑ Matematik Lojik $P \rightarrow Q$

❑ Otomatlar Teorisi

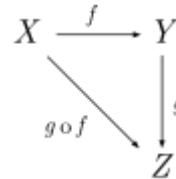


❑ Sayılar Teorisi

❑ Graf Teorisi



❑ Tip Teorisi $\Gamma \vdash x : Int$

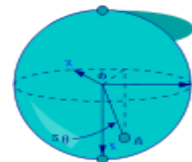


❑ Kategori Teorisi

❑ Hesaplamalı Geometri



❑ Kuantum Hesaplama



YM ile Doğrudan İlişkili Disiplin II:

Bilgisayar Mühendisliği (BM)

- ❑ BM, **hesaplamanın** pek çok yönü ile ilgilenir
- ❑ İlgili alanları **mikroişlemciler, kişisel bilgisayarlar ve süper bilgisayarlar ve devre tasarımı** vs.. yayılmıştır.
- ❑ Sadece bilgisayar sistemlerinin **nasıl çalıştığına** değil, aynı zamanda **daha büyük bir sisteme** nasıl genişletildiğine de odaklanılır.
- ❖ Bilgisayar donanımının imalatı ile ilgili tasarım, planlama, geliştirme ve test süreçleri gerçekleştirilir.
- ❖ Donanımın çiplerinden devre kontrollerine kadar tüm prosesler BM 'nin ilgi alanıdır.

Bilgisayar Mühendislerinin Çalışma Alanları

- ☐ İşletim sistemleri ve yazılım
- ☐ Veri iletişimi ve multimedya ile ilgili olarak bilgisayar ağlarının kullanımı
- ☐ Lojik tasarım
- ☐ Mikroişlemci sistemlerinin tasarımı
- ☐ Bilgisayar mimarisi
- ☐ Bilgisayar arayüzü tasarımı
- ☐ Sistem gereksinimleri ve tasarımı

YM ile Doğrudan ilişkili Disiplin III: Sistem Mühendisliği

□ Mühendisliğin disiplinler arası bir alanıdır

❖ Karmaşık mühendislik problemlerinin tasarlanması ve yönetimi ile ilgilenir.

□ Büyük ve karmaşık problemlerin incelenmesinde

❖ Farklı makineler ve makine parçalarının otomatik kontrolü ve

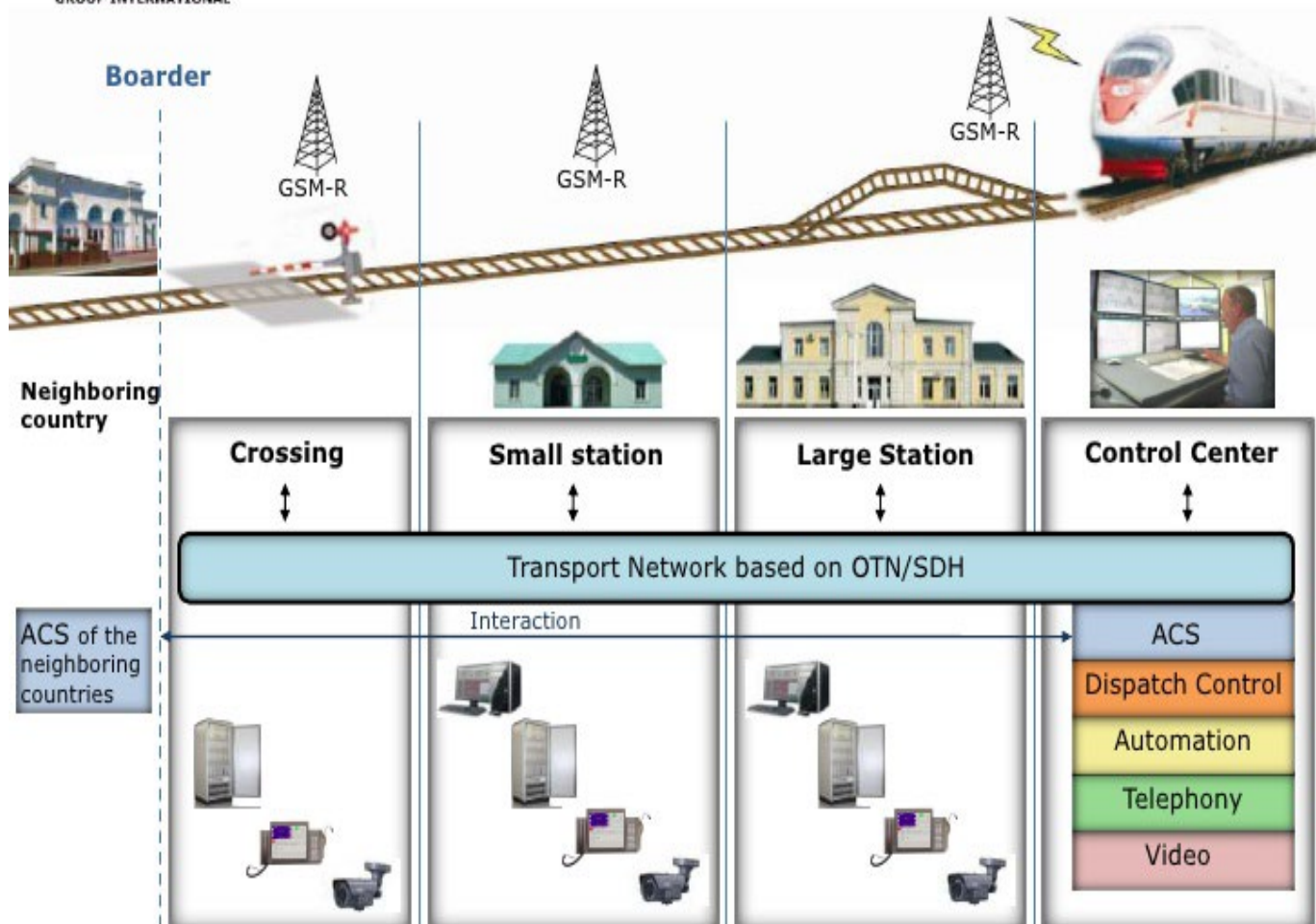
❖ Farklı çalışma takımların koordinasyonu önemlidir

❖ Lojistik önemli ve kompleks bir çalışma alanı olmuştur.

Sistem Mühendisliği Prosesleri

- ❑ Karmaşık projelerin yürütümü için iş-süreçleri geliştirir ve geliştirme araçları kullanır
- ❑ Kontrol mühendisliği ve proje yönetimi gibi hem teknik hem de insan-odaklı disiplinlerle örtüşmektedir.
- ❑ Modelleme ve simülasyon yapılır
- ❑ Gereksinimler analizi gerçekleştirilir.
- ❑ Karmaşıklığı yönetmek için planlamalar yapan bir dizi yazılım geliştirme araçları kullanır (SysUML).

The scope and coverage of the whole railway infrastructure



Integrated Railway Transport Management System

Yazılım Mühendisliği ve Biyomedikal Mühendisliği İlişkisi



- ❑ Biyomedikal Mühendisliği lisans eğitimi Türkiye’de ve dünyada hızla artmaktadır.
- ❑ Özellikle tıp fakülteleri olan üniversiteler lisans eğitim programlarına Biyomedikal Mühendisliğini de eklemektedir.

İdeaal Bir Ürün Geliştirme Takımı



Kaynak: 10 Key Roles in Software Development Team & Their Responsibilities | Alcor BPO

<https://alcor-bpo.com/recruitment-news/10-key-roles-in-a-software-development-team-who-is-responsible-for-what/>

Hangisi Takım Çalışması Yapıyor?

Functional

Common functional expertise



System analysts



Developers



Testers

Cross - Functional

Representatives from the various functions



Development Team

Projede çalışan farklı takımların da birbirleri ile iletişimi önemlidir. Farklı görüşlere sahip insanlar vizyonu yükseltirler.



Collaboration is vital for the success of the project. In fact, a team is like a clock, and each team member plays the role of a gear. For the clock to function, each gear should do its part. The same goes for the team.

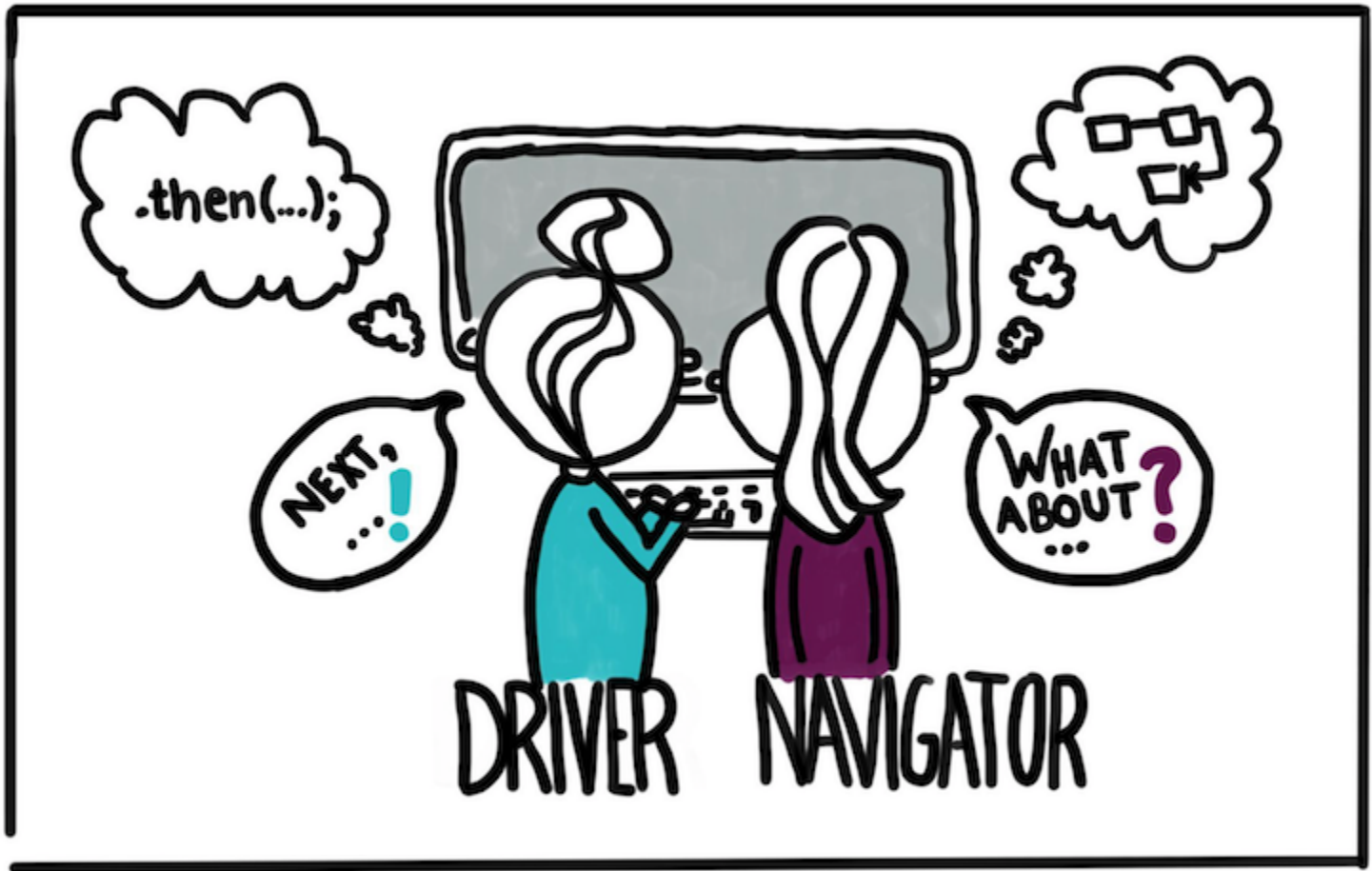
Alexander Glavatskiy
project manager at Steelkiwi inc

Data from: Steelkiwi

Verimli (efficient), diğer bir ifade ile üretken (productive) olabilmek için takımın işbirliğine ihtiyacı vardır. Sonuç KALİTELİ ürünün geliştirilmesidir.

Kaynak: Shake: Bug and crash reporting

<https://www.shakebugs.com/blog/collaboration-in-software-development/>
Benefits of teamwork and collaboration for software development teams



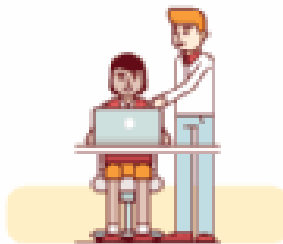
Kaynak : Martin Fowler

<https://martinfowler.com/articles/on-pair-programming.html>

Kod incelemeleri (code review) ve pair-programlammig (çift programlama, eşli programlama veya ikili programlama işbirliğine dayanan ürün geliştirmede oldukça etkindir.

Eşli Programlama

Pair programming styles



Unstructured

In unstructured pair programming, the developers can trade off who takes the lead, and should discuss decisions about the code.



Driver/Navigator

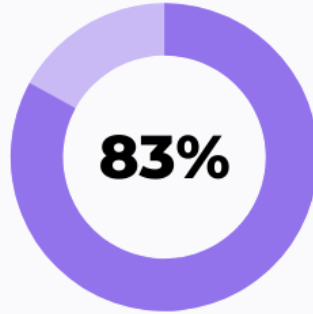
In the driver/navigator approach to pair programming, one developer sets the architectural or strategic direction, and the other implements these decisions as code.



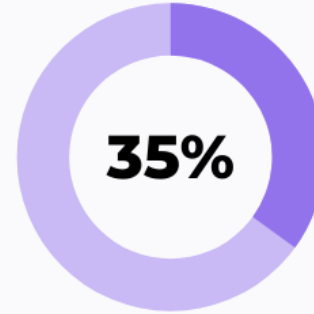
Ping-pong

Ping-pong pair programming shifts rapidly back-and-forth between the two developers, like a game of ping pong, where the software is the ball.

Collaboration in the workplace



of tech leaders that that
cross-functional collaboration
is critical for successful
software development



of those tech leaders
say investment in better
cross-functional collaboration
is a top company priority

Data from: Atlassian

Kaynak: Shake

Sonuç:

Şirketlerin üçte ikisinden fazlası, projelerinde etkili işbirliği için gereken becerileri geliştirmeye yatırım yapmıyor anlamına geliyor.

Neden ??

Alışkanlıklardan vazgeçmek kolay mı?

❑ Çoğu yazılım geliştirme şirketi işlevsellikler etrafında örgütlenir.

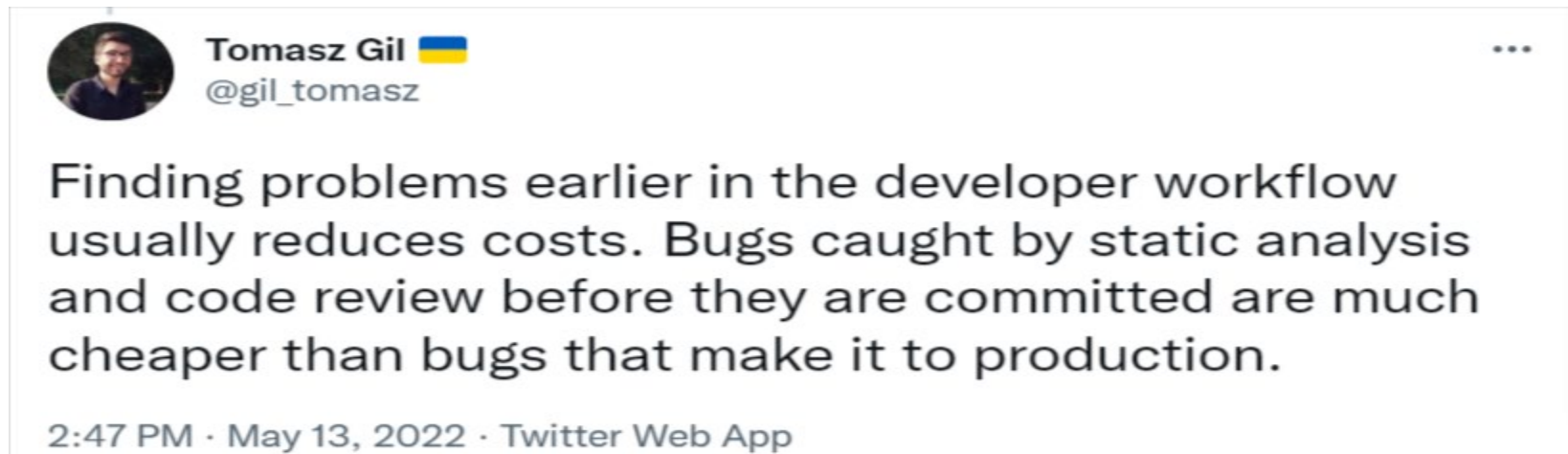
❖ İş analistleri, geliştiriciler, QA (quality assurance) test uzmanları sadece kendi işlerini yapar.

❑ Oysa çoğu proje fazla sayıda hareketli parçaya sahiptir ve bir ekibin elemanlarının herşeyden bilgisi olması imkansızdır.

❑ Sonuçlar: a) yanlış iletişim, son teslim tarihlerinin kaçırılması ve verimliliğin azalmasıdır.

b) Şirketler için zamana ve paraya mal olur.

Kod İncelemesi (Code review)



- ❑ Kod incelemeleri (code reviews), kodun kalitesini artıran ve işbirliğiyle (collaboration) gerçekleşen bir süreçtir.
- ❑ Geliştiriciler sentezlemeden önce (parçaları birleştirmeden önce) koddaki olası değişiklikleri tartışır.
 - ❖ Bunun için birbirlerinin kodları incelenir, hatalar veya çözümün doğru uygulanmadığı alanlar belirlenir.
 - ✓ Böylece sorunlar kullanıcılar veya test uzmanları tarafından görülmeden belirlenebilir.

Hackathon



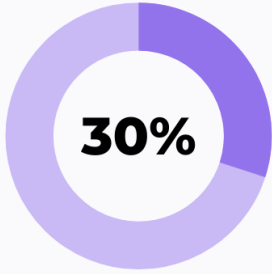
Kaynak : Microsoft Vancouver on YouTube

Hackathon nedir?

- ❑ Akranlar, yani aynı düzeyde bilgi sahibi olanlar (peers) arasında öğrenme gerçekleştirir.
- ❑ Yazılım probleminin çözümünde hızlı prototip (rapid prototyping) oluşturulur.
- ❑ Probleme uygun eni teknolojilee kullanılır.
- ❑ Böylece yazılım geliştiricilerin yeni beceriler ve teknikler konusunda uzmanlaşması sağlanır.

Brainstorm gerekli mi?

Collaboration drives innovation



Successful collaboration at work improves innovation **by 30%**

Data from: Zippia

Başarılı bir işbirliği yenilikçiliği %30 oranında artırır.

Çalışanların %80 'i hesap vermeyi ya da Sorumluluğu cezalandırma olarak düşünür.

80% of people
see accountability as punishing



Data from: Anne Loehr

YM Çalışma Alanları I

1. Bilgi Teknoloji (BT) ile ilgili kuruluşlar

BT konularında Danışmanlıklar

Büyük BT Sağlayıcı Şirketleri

Yazılım Geliştirme

Internet Servis Sağlayıcıları

Eğitim Firmaları

Perakendeciler

Hukuk Firmaları

İş Zekası ve Pazar Araştırma

Kuruluşları

Eğitim kurumları

Askeri kurumlar

Kamu sektörü

Sivil toplum kuruluşları

ÖZETLE:

BT yazılımı, sistemi ve
aygıtları kullanılan
tüm kurumlar

YM Çalışma Alanları -II

2. İmalat Sanayi

Otomotiv

Navigasyon

Telekomünikasyon

İmalat ve inşaat
şirketleri dahil.

3.Finansal hizmetler

Küresel yatırım bankaları,

Finans / bankacılık

kuruluşları,

Güvenlik sektörü uzmanları

Sigorta şirketleri

Kamu hizmetleri

Enerji ve su tedariki

Enerji çıkarımı ve nakliyesi.

Bilgi Teknolojisi (BT) nedir?

Bilginin üretilmesi, toplanması, biriktirilmesi, işlenmesi yayılması ve korunmasına yardımcı olan araçlara verilen isimdir.

BT yazılım, donanım, veri, ses iletişimi, ağlar bileşiminin temel adıdır.

BT, bu gün bilgisayar alanında **bir alt alandır** ve **bilgi işlem** ile ilgili bütün işlemleri içerir.

Bilgi Sistemleri Bileşenleri

1. Bilgisayar Donanımı

- ❑ Bilgi (information) ile çalışan fiziksel teknolojidir.
- ❑ Donanım, bir cepteki akıllı telefon ya da bir binayı dolduran bir süper bilgisayar olabilir.
- ❑ Klavye, harici disk sürücüler ve yönlendiriciler (routers) gibi bilgisayarlarla çalışan çevre aygıtlarını da (peripheral devices) içerir.
- ❑ IOT'in gelişimi ile ev aletlerinden arabalara ve kıyafetlere kadar her şeyi veri alıp gönderebilecektir
 - ❖ Çünkü bilgisayarlarla etkileşime geçen sensörlerin hızla artması insan yaşamının içine hızla girmektedir.

Bilgi Sistemleri Bileşenleri

2. Bilgisayar Yazılımı

- ❑ Donanım ne yapılacağını bilmek ister, bunu da doğal olarak yazılım gerçekleştirir.
 - ❖ Öncelikle mutlaka olması gereken yazılım Sistem Yazılımı olarak donanımın çalışmasını yönetir (işletim sistemi).
 - ❖ Daha sonra elektronik tablo kullanma, belge oluşturma veya Web sayfası tasarlama gibi belirli görevler için Uygulama Yazılımı tasarlanır.

Bilgi Sistemi Bileşenleri

3. İletişim (Telecommunications)

- ☐ Bir ağ oluşturmak için makineleri (donanımı) birbirine bağlar.
- ☐ Bağlantılar Ethernet kabloları veya fiber optik teller veya Wi-Fi yoluyla kablosuz olabilir.
- ☐ Bir ağ, bir ofis veya okul gibi belirli bir alandaki bilgisayarları **Yerel Alan Ağı (LAN)** yoluyla birbirine bağlamak için tasarlanabilir.
- ☐ Bilgisayarlar daha dağınıksa, ağa **Geniş Alan Ağı (WAN)** denir.
- ☐ İnternetin kendisi bir ağlar ağı olarak düşünülebilir.

Bilgi Sistemi Bileşenleri

4. Veri Tabanları (Databases) ve Veri Ambarları (Data Warehouses)

- ❑ Veri tabanı, verilerin toplandığı (tutulduğu) yerdir.
- ❑ Bir veya daha fazla sayıda belirlenmiş ölçütün (kriterin) tanımlanarak sorgulanması ile istenilen veriler veri tabanından çekilir.
- ❑ Veri ambarı, bir kuruluşa ait her türlü bilgiyi içerir ve aktarır.
- ❑ Veri tabanları ve veri ambarları, bilgi sistemlerinde büyük verinin (big data) ortaya çıkmasıyla daha da önem kazandı.

Büyük veri : toplanabilecek ve analiz edilebilecek büyük miktardaki veriler için kullanılan terimdir.

Bilgi Sistemi Bileşenleri

5. İnsan Kaynakları ve Süreçleri

(Human Resources and Procedures)

- ❑ Bilgi sistemlerinin en önemli bileşeni olarak değerlendirilecek bileşen insan unsurudur.
- ❑ Sistemi çalıştırmak için gerekli olan insanlar ve onların izledikleri süreçlerdir.
- ❑ Büyük veri tabanlarındaki ve veri ambarlarındaki bilgi, geçmişte olduğu gibi gelecekte de tüm veri hareketlerini yönlendirmeye devam edecektir.

Yazılım Mühendisliğinin Ortak Zorlukları

Ürün geliştirilmesi pek çok zorlukları nedeniyle karmaşık bir iştir.

- ❑ Belirli bir amaca yönelik en uygun yazılım sistemi oluşturulmalı, programda hatalar, çökmeler (başarısızlıklar) ya da farklı ilave sorunlar doğmadan mevcut bir sorun çözülmelidir.
- ❑ Belirli gereksinimleri veya istenen sonuçları karşılayan yazılım sistemlerin tasarlayanları, büyük ölçüde *yaratıcı problem çözümlerine* güvenirler.
- ❑ Çeşitli platformlar arasında uyumluluğu güvenilir olarak sağlarken yeni teknolojilere uyum da zordur.
- ❑ Sürekli değişen teknoloji ortamında yazılımın etkinliğini ve doğruluğunu sağlamak, yeni trendlere ve kalıplara ayak uydurabilmeyi zorunlu kılar. Sonuç olarak:

Araştırma ve geliştirme çalışmalarında başarı için kişinin *yaratıcılığı, becerisi, uzmanlığı ve deneyimi* gerekir.

YM Çalışma Alanı –I

Quality Assurance (QA) Engineering

Kalite Güvence Mühendisliği

- ❑ Yazılım ürünün geliştirilmesinde önemli bir süreçtir.
- ❑ Geliştirme ekiplerinin doğru bir iz sürmesine yardımcı olacak nitelikli **geri bildirimler** sağlanır.
 - ❖ Ürün veya hizmetlerin bir dizi standardı karşılaması istenir.
 - ✓ The Capability Maturity Model Integration (CMMI)
 - ✓ Six Sigma
- ❑ Veri toplama ve araştırma ile test uygulamaları geliştirilir ve süreçler değerlendirilir.
- ❑ Mevcut ürün tasarımları doğrulanır ve tasarım sürecindeki potansiyel kusurlar belirlenir.
 - ❖ Müşteri ihtiyaçlarını karşılayan ürünler yaratılır.

YM Çalışma Alanı II

Video Game Design

- ❑ Tasarım ekibi, sanatçılar, kodlayıcılar ve test uzmanlarından oluşur
- ❖ Güçlü işbirliği gerektiren bir çalışma alanıdır.
- ❑ Milyonlarca insanın etkileşime girebileceği ve keyfini çıkarabileceği sürükleyici **sanal deneyimler** oluşturulur.
- ❖ Çalışma alanı oluşturma, modelleme, hikaye yazma, seviye tasarımı, çizim, programlama, seslendirme ve ses mühendisliği gibi çeşitli disiplinlerden unsurları birleştirir.
- ❑ Dünya tasarımı (örneğin <https://wdo.org/>) , seviye tasarımı, sistem tasarımı, içerik tasarımı ve kullanıcı arayüzü tasarımı gibi alt disiplinleri de içerir.
- ❖ Yaratıcılığı ve teknoloji dikkatli planlamayla birleştirilir.
 - ✓ Tasarımcılar, her oyun deneyimini dijital dünyaya giren herkes için benzersiz ve unutulmaz kılmak için çabalar.

Çalışma Alanı III

Software Integration Engineering (SIE)

- ❑ SIE, yazılım bileşenlerini çalışan çözümlerde birleştirir.
- ❑ SIE, çeşitli kod tabanlarının geliştirilmesi ve yapılandırılması, bileşenler arasındaki veri akışlarının yönetilmesi ve sistemdeki sorunların ayıklanması gibi geniş alana yayılır.
- ❑ SIE ile istenilen sonuca ulaşmak ve uygulamak için, yazılım mühendisliği, programlama ve veritabanı yönetimi deneyimi zorunludur.
- ❑ Sistem entegrasyon mühendisleri, yazılım uygulamalarının karmaşıklığından doğan sorunları çözümleyebilirler.
 - ❖ modern dijital çözümlerin hayati bir parçasıdır.
- ❑ Yazılım entegrasyon mühendisleri, farklı uygulamaların birbirleri ile iletişimini ve bir bileşendeki değişikliklerin tüm sistemi nasıl etkilediğini anlar ve farklı ölçekteki şirketlere güvenilir çözümler sağlar.

Çalışma Alanı -IVFront-End Engineering

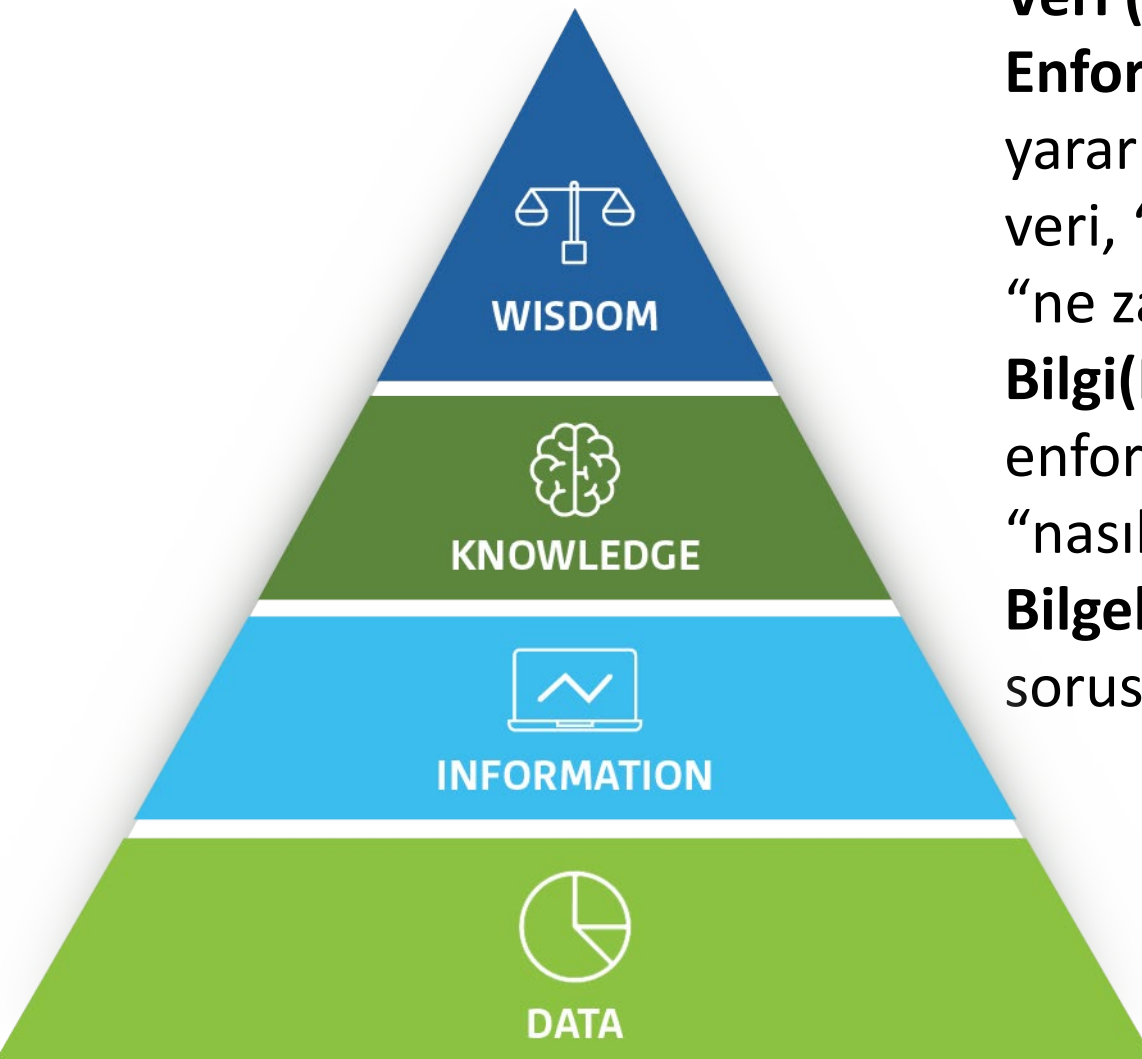
- ❑ Ön uç mühendisliği, gelişmiş erişilebilirlik, görsel tasarım ve hızlı yanıt verme ile web sitelerini ve uygulamaları kullanıcı için daha iyi hale getirir.
- ❑ Ön uç mühendisleri web sitesini veya uygulamayı keyifli hale getirmek için kullanılabilirliği artıran özellikleri uygular.
- ❑ Bu, yeni modellerin kodlanmasını, sayfa öğelerinin yapılandırılmasını, bileşenlerin şekillendirilmesini, çeşitli sonuçların test edilmesini, yanıt eylemleri oluşturulmasını ve çözümlerin hatalarını ayıklamayı içerebilir.
- ❑ Sonuçta bir ön uç mühendisin hedefi iyi performans gösteren ve **kullanıcı deneyimini (UX/UI)** geliştiren bir dijital ürün yaratmaktır.

Çalışma Alanı V

Full-Stack Development

- ❑ Uçtan uca yazılım geliştirme metodolojilerinin bilinmesidir. Aynı zamanda tüm alanlardaki programlama dillerine hakim olunmasını gerektirir.
- ❑ Arka uç ve ön uç teknolojilerinin birleşimine dayalı web uygulamaları geliştirme sürecidir.
- ❑ Geliştirici, veritabanları ve sunuculardan kullanıcı arayüzü tasarımına ve istemci tarafı komut dosyalarına kadar tüm farklı bileşenleri çok iyi bilmeli, ayrıca hızlı geri dönüş süresi olması nedeni ile daha iyi yönetilebilecek projeler için tüm bileşenlerin nasıl birleştiğini anlamalıdır.
- ❑ HTML, CSS, JavaScript, Swift ve Node.js gibi birçok kodlama dilinde iyi bilgi sahibi olunmalıdır.
 - ❖ Zorlayıcı bir iştir, ama problem çözebilen ve mantıksal akıl yürütebilenler için ödüllendiricidir.

DIKW Piramidi / Bilgi Hiyerarşisi



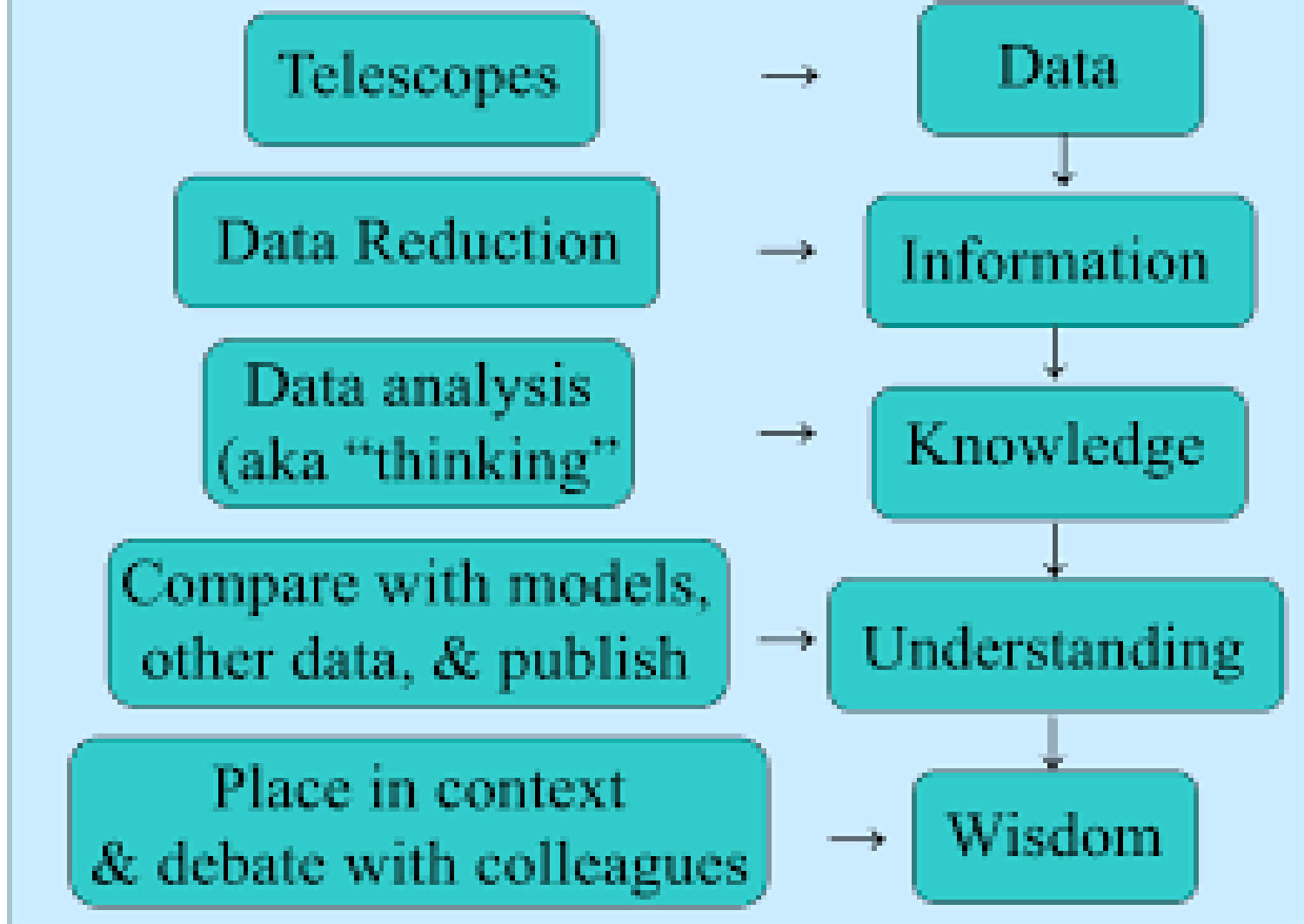
Veri (DATA) : semboller

Enformasyon(INFORMATION):

yararlı olacak şekilde işlenmiş veri, “kim”, “ne”, “nerede” ve “ne zaman” sorularına yanıt verir

Bilgi(KNOWLEDGE): veri ve enformasyonun uygulaması, “nasıl” sorusuna yanıt verir

Bilgelik(WISDOM) : Neden sorusu yanıtlanarak değerlendirir



Teleskopdan sürekli olarak veri alınır. DATA

Bunlardan bir kısmı kullanılmayacaktır, elenir. INFORMATION

Kaydedilenler analiz edilerek anlamlı sonuçlar çıkarılır. KNOWLEDGE

Sonuç diğer ortamlarda da ilgililere kullanıma açılır, gerekirse yeniden değerlendirilir ve yeniden işlenir. WISDOM

DIKW Piramidi

Use Knowledge to
Establish and
Achieve Goals

Wisdom

Analyze and Synthesize
Derived Information

Knowledge

Give Meaning
to Obtained Data

Information

Obtain
Raw Facts

Data