

# CHATGPT İLE ÜRETKEN YAPAY ZEKA

## PROMPT MÜHENDİSLİĞİ UYGULAMALARI

**VEBENDE A.Ş.**

CYBER SECURITY, DEVSECOPS,  
KUBERNETES, INFRASTRUCTURE AS CODE,  
CI/CD, DIGITAL TRANSFORMATION, LINUX,  
CODE QUALITY

@[Linkedin.com/company/vebende/](https://www.linkedin.com/company/vebende/)  
@[Linkedin.com/in/tuncerkaraarslan/](https://www.linkedin.com/in/tuncerkaraarslan/)

[www.vebende.com.tr](https://www.vebende.com.tr) 🔍



# YAPAY ZEKÂ İLE KONUŞ, İŞİNİ GÜÇLENDİR

PRATİK YAKLAŞIMLAR, SOMUT ÇÖZÜMLER

# GPT MODELLERİNİN KISA TARİHİ (2018 – 2024)

## ◆ GPT-1 (2018) – Teorik Kanıt ve İlk Adım

- **Model Boyutu:** 117 milyon parametre
- **Yayın:** OpenAI, "Improving Language Understanding by Generative Pre-training" başlıklı bir akademik makale ile tanıttı.
- **Amaç:** Dil modellemesinde **önceden eğitilmiş** modellerin gücünü göstermek.
- **Teknik Özellik:**
  - **Transformer** mimarisi kullanıldı.
  - Sadece **unsupervised** (etiketsiz) öğrenmeye dayalıydı.
- **Sınırlamalar:** Çok küçük parametre sayısı, zayıf bağlam kurma, üretim kalitesi düşüktü.

## ◆ GPT-2 (2019) – Dikkat Çekici Dönüm Noktası

- **Model Boyutu:** 1.5 milyar parametre
- **Kapasite Artışı:** GPT-1'in 10 katından fazla veriyle eğitildi.
- **Gelişmeler:**
  - **Zero-shot, one-shot, few-shot learning** kavramlarının güçlü örnekleri bu modelde ilk kez görüldü.
  - Özetleme, çeviri, soru-cevap gibi görevlerde etkileyici performans sergiledi.
- **Yayın Kararı:** Başlangıçta "kötüye kullanım riski" nedeniyle tam modeli yayınlamadı. Bu, **etik AI tartışmalarını** tetikledi.

## ◆ GPT-3 (2020) – Üretkenliğin Yeni Çağı

- **Model Boyutu:** 175 milyar parametre
- **Veri Kaynağı:** Common Crawl, Books, Wikipedia ve Web verileri.
- **Öne Çıkanlar:**
  - İnsan benzeri metin üretimi
  - Kod yazma (ilk Copilot prototipleri)
  - "Biraz bağlam ver, görev çözülsün" tarzı etkileşim modeli
- **Sınırlamalar:**
  - Güvenilirlik sorunları (hallüsinasyon)
  - Gerçeklik denetimi eksikliği
  - Uzun bağlamlı konuşmalarda tutarlılık kaybı

## ◆ GPT-4 (2023) – Çok Modlu ve Daha Derin Anlayış

- **Parametre Sayısı:** Açıklanmadı, ancak 1 trilyon seviyelerinde olduğu tahmin ediliyor.
- **Yeni Yetkinlikler:**
  - **Multimodalite:** Görsel ve metin girdisi birlikte işlenebiliyor (GPT-4-Vision)
  - Daha uzun bağlam hafızası
  - Daha az hallüsinasyon
  - Kod üretimi, argümantasyon ve teknik açıklamalarda güçlenmiş yapay zekâ
- **Kullanım:**
  - ChatGPT Plus abonelerine sunuldu.
  - OpenAI API ve Microsoft Copilot sistemlerine entegre edildi.

## ◆ GPT-4o o: omni (2024 Mayıs) – Gerçek Zamanlı Çok Modlu Model

- **Tek Temel Model:** Metin + Görsel + Ses girişi/çıkışı destekleniyor.
- **Tepki Süresi:** İnsan konuşmasına benzer hız (320 ms'ye kadar ses üretimi).
- **Yenilikler:**
  - Duygusal ton algılama ve ifade etme
  - Anında çeviri
  - Sesli çok modlu uygulamalar için SDK duyuruldu.
- **Etkisi:**
  - GPT artık klasik “chatbot” sınırlarını aştı, gerçek zamanlı etkileşimli yardımcı haline geldi.

## OpenAI'nin Stratejik Yaklaşımı

OpenAI'nin yaklaşımı, üretken yapay zekânın farklı kullanım senaryolarına hizmet edebilecek **çok katmanlı bir ürün-ekosistemi** oluşturmak üzerine kuruludur. Bu yapı, bireysel kullanıcıdan büyük ölçekli kurumsal entegrasyonlara kadar geniş bir alana hizmet eder.

OpenAI'nin stratejik yaklaşımlarını üç ana başlıkta inceleyelim

## 1. API Tabanlı Platform: Geliştiriciler İçin Esneklik ve Güç

Geliştiriciler API'ler üzerinden kendi uygulamalarına doğal dil işleme (NLP), metin üretimi, çeviri, kod yazma gibi yetenekleri entegre edebilir.

## Temel Özellikler:

- **Model Seçimi:** gpt-3.5-turbo, gpt-4, gpt-4o gibi farklı versiyonlara erişim
- **Bağlam yönetimi:** messages, temperature, top\_p gibi ayarlarla çıktının kontrolü
- **Dönüştürülebilir Kullanım:**
  - Chatbotlar
  - Kod üreticiler
  - İçerik üretim platformları
  - Analitik asistanlar

## Kurumsal Örnek:

Bir finans firması, müşteri e-postalarını sınıflandırmak için OpenAI API'yi kullanarak otomatik etiketleme sistemi geliştirebilir.

## 2. ChatGPT Uygulaması: Kullanıcı Dostu Arayüzde Güçlü Model Erişimi

ChatGPT, OpenAI'nin son kullanıcılar için sunduğu arayüzdür. Hem bireysel hem de iş amaçlı kullanım için optimize edilmiştir.

## Kullanım Biçimleri:

- **Web Uygulaması:** [chat.openai.com](https://chat.openai.com)
- **Mobil Uygulama:** iOS ve Android versiyonları
- **Planlar:**
  - Ücretsiz (GPT-3.5)
  - Plus (GPT-4)
  - Team ve Enterprise sürümleri (GPT-4o, özel güvenlik/kontrol özellikleri ile)

## Yenilikçi Özellikler:

- Dosya yükleme ve analiz
- Kod editörü (Python çalıştırma)
- Görsel algılama (GPT-4o ile)
- Kendi GPT'ni oluştur (Custom GPTs)

## Kurumlar için Kullanım Senaryoları:

- Teknik dokümantasyon yazdırma
- Eğitim materyali oluşturma
- İç sistemler için danışman (Prompt + API ile birlikte)

### 3. Özel Modeller (Custom GPTs & Fine-Tuning): Kurumsal Özelleřtirme Yeteneęi

### 3.1. Custom GPTs (Kapsül Modeller)

- ChatGPT arayüzü üzerinden herkesin oluşturabileceği özel GPT'lerdir.
- **Bileşenler:**
  - Özel talimatlar
  - Dosya yükleme
  - API bağlantısı (ör. bir veri tabanına, dış servise)
- **Kullanıcılar İçin:** Kodu bilmeyen ama kendi iş akışına uyarlamak isteyen kişiler için idealdir.

## 3.2. Fine-Tuning (Model Eđitimi)

- GPT-3.5 modeli zerine kurumsal verilerle zel eđitim yapılabilir.
- Ama: Modelin spesifik terim, slup, ierik kalıplarına ađına hale gelmesi.
- **Uygulama Alanı:**
  - Mđteri destek botları
  - Marka diline zel ierik reticiler
  - Teknik destek asistanları

### 3.3. Enterprise-Level Deployment:

- Büyük firmalar, OpenAI'nin altyapısını kendi veri güvenlik gereksinimlerine göre kullanabiliyor.
- Azure OpenAI entegrasyonu sayesinde veri yerelliği, denetim ve uyumluluk sağlanabiliyor.
- **Kurum içi LLM mimarisi:**
  - ChatGPT + şirket verisi
  - RAG (Retrieval-Augmented Generation) ile arama entegrasyonu
  - Team Workspace yönetimi, erişim rolleri

### Özet Perspektif:

| Yapı          | Hedef Kitle                       | Avantaj                                      |
|---------------|-----------------------------------|----------------------------------------------|
| API           | Geliştiriciler                    | Uygulama içi AI entegrasyonu, otomasyon      |
| ChatGPT       | Bireysel ve kurumsal kullanıcılar | Hızlı üretkenlik, etkileşimli kullanım       |
| Özel Modeller | Kurumlar ve sektör uzmanları      | Özelleştirme, marka dili, iç veriye uyarlama |

## ◆ ChatGPT'nin Kurumsal Hayatta Kullanım Alanları

## 1. Dokümantasyon Üretimi

### Senaryolar:

- API dökümantasyonu üretimi
- Yazılım kütüphanelerinin kullanım örnekleriyle açıklanması
- Kullanıcı kılavuzu, kurulum adımları, SSS belgeleri

### Prompt Örneği:

“Aşağıdaki Python kodunu temel alarak teknik bir kullanım dökümanı üret. Başlıklar: Tanım, Gereksinimler, Kurulum, Örnek Kullanım”

## 2. Teknik Asistan ve Kodlama Yardımcısı

### Kullanım Amacı:

Yazılım ekiplerinin günlük geliştirme süreçlerinde destek almak, kod şablonları oluşturmak, refactor yapmak veya dil/dokümantasyon dönüşümü gerçekleştirmek.

### Senaryolar:

- Bir fonksiyonun başka dile çevrilmesi (örneğin Python → Java)
- Kodun açıklanması ve optimize edilmesi
- SQL sorgularının yeniden yazımı

### Prompt Örneği:

"Aşağıdaki C# kodunu Java Spring Boot'a çevir ve her adımı açıklayıcı notlarla açıkla."

### 3. Test Senaryoları ve Test Verisi Üretimi

#### Kullanım Amacı:

Yazılım test ekiplerinin sistem, entegrasyon ve uçtan uca test süreçleri için senaryo ve test verisi oluşturmalarını hızlandırmak.

#### Senaryolar:

- Bir user story'ye karşılık test case oluşturulması
- Unit test şablonlarının jenerasyonu (ör. xUnit, JUnit, PyTest)
- Edge case ve negatif test üretimi

#### Prompt Örneği:

"Aşağıdaki kullanıcı hikayesi için 5 adet pozitif, 3 adet negatif test senaryosu üret.  
Format: Test Adı, Giriş Verisi, Beklenen Çıktı"

## 4. Eğitim İçeriği ve Yazılım Geliştirme Desteği

### Kullanım Amacı:

İç eğitim birimlerinin, teknik eğitimcilerin veya bilgi yönetimi departmanlarının eğitim dokümanları, öğrenme materyalleri ve örnek uygulamalar üretmesi.

### Senaryolar:

- Eğitim modülü oluşturma (konu başlıkları, öğrenme hedefleri, uygulama alıştırmaları)
- Teknik kavramların farklı seviyelerde açıklanması (beginner–intermediate–advanced)
- Yazılım eğitimleri için örnek kod üretimi (multi-language)

### Prompt Örneği:

“Yeni başlayanlar için ‘Dependency Injection’ konusunu sade bir şekilde açıkla. C#, Python ve Java Spring örnekleri ver.”

## Kurumsal Kullanım Örnekleri (Özet Tablo)

| Alan            | Amaç                                       | Kazanım                            |
|-----------------|--------------------------------------------|------------------------------------|
| Dokümantasyon   | Sürekli güncel teknik içerik üretmek       | Zaman tasarrufu, içerik kalitesi   |
| Teknik Asistan  | Geliştirici destek, dil dönüşümü, refactor | Kod kalitesi ve üretkenlik artışı  |
| Test Otomasyonu | Test senaryosu ve veri üretimi             | QA süreci hızlanır ve çeşitlenir   |
| Eğitim İçeriği  | Teknik eğitim modülü ve uygulama üretimi   | Kurum içi bilgi paylaşımı güçlenir |

# ChatGPT Nasıl Çalışır? Teknik Temel ve Kavramlar

## 1. Transformer Mimarisi

### Tanım:

GPT, "Transformer" isimli bir derin öğrenme mimarisine dayanır. Bu mimari, dil modelleme görevlerinde bağlamı paralel olarak öğrenme yeteneği sayesinde önceki yöntemlere göre çok daha etkilidir.

### Öne Çıkan Yapılar:

- Self-Attention (Kendine Dikkat):
- Her kelime (token), diğer tüm kelimelerle bağ kurar. Böylece dilin yapısal ilişkileri öğrenilir.
- **Katmanlı Yapı:**
- Her bir "layer", bir önceki çıktıyı alıp daha karmaşık örüntüler öğrenir.

## 2. Parametre Sayısı & Dil Modeli Nedir?

### **Dil Modeli (Language Model):**

Doğal dil girdileri alır, çıktıyı istatistiksel olarak tahmin eder. GPT gibi modeller, verilen bir girişin ardından hangi kelimenin geleceğini öngörmeye çalışır.

### **Parametre:**

Modelin öğrendiği ağırlıklardır. Ne kadar çok parametre varsa, o kadar çok bağlam, örüntü ve ilişki öğrenilebilir.

### 3. In-Context Learning (Bağlamdan Öğrenme)

#### **Tanım:**

GPT modelleri, yeni görevleri öğrenmek için eğitilmek zorunda değildir. Sadece örnek verilerek bağlam içinde görev "öğrenebilir".

#### **Senaryo:**

"Bu metni özetle" → Model daha önce "özetle" komutunun ne anlama geldiğini öğrenmişse, sadece bağlamdan yola çıkarak görev tanımını anlar.

#### **Avantajı:**

Modeli yeniden eğitmeye gerek kalmaz; prompt üzerinden görev tanımı yapılır.

## 4. Token Nedir? Neden Önemlidir?

### Token:

Metindeki kelimelerin veya kelime parçacıklarının makineye çevrilmiş hâlidir.

- “Merhaba dünya!” → 3 token (yaklaşık)
- Token sayısı ≠ kelime sayısı

### GPT'de Token Sınırları:

- GPT-3.5: ~4.096 token
- GPT-4 (Turbo): 128.000 token

### 5. Model Versiyonları: GPT-3.5 vs GPT-4 vs GPT-4o

| Özellik        | GPT-3.5    | GPT-4         | GPT-4o             |
|----------------|------------|---------------|--------------------|
| Yayın Tarihi   | Kasım 2022 | Mart 2023     | Mayıs 2024         |
| Girdi          | Metin      | Metin, Görsel | Metin, Görsel, Ses |
| Yanıt Kalitesi | Orta       | Yüksek        | Çok yüksek         |
| Token Hafızası | 4k         | 32k / 128k    | 128k               |
| Hızı           | Hızlı      | Daha yavaş    | En hızlı           |

#### GPT-4o'nun Öne Çıkan Farkı:

- Gerçek zamanlı sesli iletişim yeteneği (300-500 ms gecikme ile)
- Tek bir model içinde çok modlu işleme (metin + ses + görsel)

## 6. Kontekst, Hafıza ve Oturum Yönetimi

### Kontekst (Bağlam):

Modelin bir oturum boyunca hatırladığı tüm girdiler ve çıktılarıdır.

### Hafıza:

- **ChatGPT Plus kullanıcıları için bazı bilgiler model tarafından “kalıcı” hafızaya alınabilir** (isteğe bağlı olarak)
- "Senin adım Tuncer" dediğinizde bunu sonraki görüşmede hatırlaması örneği

### Oturum Yönetimi:

- Her yeni konuşma “stateless”tir, ancak önceki mesajlar prompt içinde tekrar edilerek bağlam sağlanabilir.
- Kurumsal senaryolarda **uzun bağlam yönetimi (128k token)** kritik hale gelir.

# Prompt Mühendisliğine Giriş

## 1. Prompt Nedir? Prompt vs Soru

### Tanım:

**Prompt**, yapay zekâya verilen açık talimattır.

Bir sorudan farklı olarak sadece bilgi istemekle kalmaz, aynı zamanda modelin **nasıl, ne şekilde ve ne bağlamda cevap vereceğini** belirler.

### Fark:

| Yapı   | Örnek                                                                                | Açıklama                                   |
|--------|--------------------------------------------------------------------------------------|--------------------------------------------|
| Soru   | "Java nedir?"                                                                        | Genel, yönlendirme içermez                 |
| Prompt | "Java'nın nesne yönelimli özelliklerini sade bir dille açıkla ve C# ile karşılaştır" | Bilgi + görev + biçim yönlendirmesi içerir |

## 2. Prompt Neden Önemlidir?

### Temel Gerçek:

Aynı bilgiye ulaşmak için **farklı yollarla, farklı detay seviyelerinde** sonuç alabilirsiniz. Prompt, bu yolculuğu yöneten kılavuzdur.

### Analojik Açıklama:

Yapay zekâ, bir çırak gibidir. Ne kadar açık, yönlendirici ve yapılandırılmış konuşursanız, o kadar verimli çalışır.

### 3. Kötü Prompt Örnekleri

#### Örnek 1 – Yetersiz:

“Projem hakkında fikir ver.”

→ **Sorun:** Bağlam yok, konu yok, çıktı beklentisi belirsiz.

#### Örnek 2 – Belirsiz:

“Yardım eder misin?”

→ **Sorun:** Görev belirtilmemiş, model ne yapacağını bilemez.

#### Örnek 3 – Bağlamdan Kopuk:

“Kodum çalışmıyor.”

→ **Sorun:** Hangi kod? Hangi hata? Hangi teknoloji? Girdi eksik.

## 4. İyi Prompt Tasarımının 3 İlkesi

### ✓ 1. Bağlam Verme (Contextualization)

Kullanılan teknolojiyi, hedef kullanıcıyı, işlevi veya ortamı açıkla.

#### Örnek:

“Bir e-ticaret web sitesinde sepet yönetimi yapan bir Node.js servisinin açıklamasını yaz.”

## ✓ 2. Yönlendirme Yapma (Instructional Clarity)

Modelden ne yapmasını istediğini açıkça belirt: Açıkla mı, listele mi, kodla mı?

### Örnek:

“Node.js servisinin işlevini açıklayıp, potansiyel güvenlik açıklarını listele.”

### ✓ 3. Biçim Belirtme (Format Expectation)

Cevabın nasıl formatlanması gerektiğini belirt: Madde madde mi? JSON formatında mı? Markdown olarak mı?

#### Örnek:

“Sonuçları JSON formatında ver: { risk\_adı, açıklama, çözüm } şeklinde.”

## Özet: Kötü → İyi Dönüşüm

| Kötü Prompt       | İyi Prompt                                                                                                           |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| "Kodum bozuk."    | "Aşağıdaki Python kodu, veritabanına bağlanırken hata veriyor. Lütfen hata nedenini açıkla ve düzeltme önerisi sun." |
| "Bana rapor yaz." | "Aşağıdaki satış verilerine göre haftalık trend analiz raporu üret. Tablo ve kısa bir özet paragraf içersin."        |

# 🔑 Uygulamalı Prompt Teknikleri



## 1. Rol Atama (Role Prompting)

### Örnek Prompt:

"Bir siber güvenlik uzmanı gibi konuş. Bir e-ticaret sitesinin kullanıcı girişi kısmındaki potansiyel açıkları değerlendir."

### Analiz:

- Modelin bilgi düzeyi derinleşir.
- Teknik terminoloji artar.
- Öneriler sektörel olur.

### ✓ İpucu:

Rolü net belirt, varsa hedef kitleyi de tanımla.

## 2. Biçim Belirtme (Output Formatting)

### Örnek Prompt:

"Aşağıdaki kullanıcı şikayetlerini özetle ve {şikayet\_tipi, frekans, çözüm\_önerisi} formatında JSON çıktısı üret."

### Analiz:

- Doğrudan entegrasyona uygun veri çıkışı.
- Otomasyon ve analiz sistemlerine uygun yapı.

### ✓ İpucu:

Dilersen Markdown, XML, CSV gibi formatları da talep edebilirsin.

### 3. Aşamalı Görev Paylaşımı (Chain of Thought Prompting)

#### Örnek Prompt:

"Aşağıdaki algoritmanın neden yavaş çalıştığını adım adım analiz et. Önce kodu oku, sonra darboğazı bul, en son çözüm öner."

#### Analiz:

- Model mantıksal sıralı düşünmeye zorlanır.
- Hata analizlerinde çok faydalıdır.

#### ✓ İpucu:

Süreç odaklı ifadeler kullan: önce, sonra, son olarak...

## 4. Örnekle Öğretme (Few-shot Prompting)

### Örnek Prompt:

\*"Aşağıdaki gibi örnek sorular üret:

1. 'Java'da bir sınıf nasıl tanımlanır?'
2. 'Python'da liste nasıl sıralanır?' Lütfen bu tarzda 5 yeni teknik soru üret."

### Analiz:

- Çıktı stili, içerik tonu ve yapı örneğe sadık kalır.
- Eğitim, sınav, test üretimi için çok uygundur.

### ✓ İpucu:

Kaliteli birkaç örnek, modelin kalitesini doğrudan belirler.

## 5. Görev Bölme ve Parçalara Ayırma (Decomposition Prompting)

### Örnek Prompt:

\*"Bir kurumsal eğitim sunumu hazırlanacak. Lütfen:

- 1.Konu başlıklarını oluştur,
- 2.Her başlık için 3 alt madde öner,
- 3.En sonunda her başlığa 1 gerçek yaşam örneği ekle."

### Analiz:

- Model karmaşık görevlerde hata yapmadan adım adım içerik üretir.
- Geliştirici belgeleri, eğitim materyali ve rapor şablonları için idealdir.

## Ekstra Teknikler

| Teknik                  | Amaç                            | Örnek                                   |
|-------------------------|---------------------------------|-----------------------------------------|
| Negatif Yönlendirme     | Ne yapmaması gerektiğini belirt | "Çok teknik terim kullanma, sade anlat" |
| Çok Modlu Prompting     | Görsel, metin, ses kombinasyonu | "Bu diyagramı analiz et ve anlat"       |
| İçerik Yeniden Yazdırma | Biçim dönüşümü                  | "Bunu teknik blog yazısına çevir"       |

✓ Uygulamalı Senaryo: Yazılım, Eğitim veya CRM için Prompt Tasarımı

## ◆ a. Yazılım Geliştirme Senaryosu

### 📌 **Senaryo Prompt'ı:**

"Bir .NET uygulaması için RESTful API örneği üret. Swagger ile dokümante et. Açıklamaları Türkçe yaz."

## 🔍 İlk Yanıt Analizi:

- Genellikle basit bir controller sınıfı, temel GET/POST metodları ile döner.
- Swagger etiketi eklenmiş olabilir ama dokümantasyon eksik olabilir.
- Veri modeli (DTO, Entity) eksik olabilir.

## ⚙️ Geliştirilmiş Prompt Örneği:

"C#, .NET 6 ile yazılmış bir RESTful API örneği oluşturun. 'Product' nesnesi için CRUD metodlarını yaz. Swagger/OpenAPI açıklamalarını dahil et. Açıklamalar Türkçe olsun. Veri modeli, controller ve startup yapılandırmasını da ekle. Kodu bölümlere ayır ve her bölümü kısa açıklamalarla sun."

## ✓ İyileştirme Noktaları:

- **Bağlam verildi:** [.NET 6, Product nesnesi]
- **Biçim belirtildi:** CRUD, Swagger/OpenAPI, açıklamalı çıktı
- **Yönlendirme sağlandı:** Kod bölümlere ayrıldı, her bölüm açıklamalı olacak şekilde belirtildi

## ◆ b. Eđitim İeriđi Senaryosu

### 📌 **Senaryo Prompt'ı:**

"Siber gvenlik eđitimi iin 3 modll bir eđitim planı oluřtur. Her modlde đrenme hedefleri ve uygulama nerileri ver."

## İlk Yanıt Analizi:

- Modüller listelenir (örneğin: ağ güvenliği, saldırı tespiti vs.)
- Öğrenme hedefleri yüzeysel olabilir.
- Uygulamalar genellikle jenerik kalabilir (örneğin: “bir laboratuvar yapın”).

## ⚙️ Geliştirilmiş Prompt Örneği:

\*"Siber güvenlik alanında temel seviye hedef kitle için 3 modüllük bir eğitim planı hazırla.

### **Her modül için:**

- 1.Öğrenme hedeflerini (SMART formatında),
- 2.2 uygulamalı etkinlik (örneğin: Wireshark analizi, port taraması gibi),
- 3.Değerlendirme yöntemi (quiz, proje vb.) belirt.
- 4.İçeriği Türkçe yaz."\*

## ✓ İyileştirme Noktaları:

- **Kapsam netleşti** (temel seviye, hedef kitle)
- **Uygulama önerisi somutlaştı**
- **Değerlendirme unsuru eklendi**



## ◆ c. Müşteri Hizmetleri / CRM Senaryosu

### 📌 Senaryo Prompt'ı:

"Bir e-ticaret firmasında gelen iade talebi sonrası müşteriye gönderilecek 3 farklı tonlamada mail taslağı üret (resmi, samimi, özür odaklı)"

## İlk Yanıt Analizi:

- Üç farklı mesaj dönebilir, ancak tonlamalar arasındaki farklar çok keskin olmayabilir.
- Mesajlar benzer içerikte olabilir (yalnızca hitap değişebilir).

## ⚙️ Geliştirilmiş Prompt Örneği:

\*"Bir e-ticaret firmasına gelen ürün iadesi sonrası müşteriye gönderilecek 3 farklı tonda e-posta metni oluştur:

1. Kurumsal ve resmi,
2. Samimi ve markaya özel bir dilde,
3. Özür odaklı, empati içeren.
4. Her e-posta en fazla 100 kelime olsun. Açılış ve kapanış cümleleri birbirinden farklı olmalı. Mesajlar Türkçe yazılsın."\*

## ✓ İyileştirme Noktaları:

- Tonlar net belirlendi
- Kelime sınırı getirildi (kısa yazım zorlandı)
- Açılış/kapanış farkı vurgulandı (çeşitlilik artırıldı)

## Genel Analiz Soruları:

1. Prompt yeterince açık mıydı?
2. Bağlam verildi mi? Kime yönelik yazıldığı belirtildi mi?
3. Yanıtın biçimi net mi? Paragraf mı, tablo mu, kod mu?
4. Sonuç tatmin edici mi, eksik kalan kısımlar nelerdi?
5. Prompt geliştirildikçe çıktı kalitesi nasıl değişti?



## Soru-Cevap ve Tavsiyeler

### 1. ! Sık Yapılan Hatalar

- **Yetersiz veya bağlamsız prompt kullanımı**
- “Kod yazar mısın?” gibi soyut istekler, çoğunlukla yetersiz veya yanlış cevap üretir.
- **Sadece ilk cevabı yeterli görmek**
- Modelin çıktısı doğrusal değil; genellikle iteratif denemelerle geliştirilmeli.
- **Rol belirtmeden teknik bilgi istemek**
- “API örneği ver” yerine “Deneyimli bir backend geliştirici gibi RESTful API örneği yaz” daha etkili olur.

## 2. ⚠️ ChatGPT'nin Sınırları ve Riskleri

### 🧠 Hallüsinasyon Riski (Uydurma Bilgi)

- Model bazen olmayan kaynaklar, uydurma kodlar veya yanlış istatistikler verebilir.
- Örnek: "Python'da fastlog kütüphanesi nedir?" gibi bir soruya model var olmayan bir açıklama yapabilir.

✅ **Çözüm:** Cevapları her zaman ikinci bir kaynakla doğrulamak gerekir. Akademik veya hukuki alanda doğrudan kullanmak sakıncalıdır.

## Kaynak Doğrulama Gerekliliđi

- ChatGPT bir bilgi tabanı deđil, olasılıksal dil modelidir.
- Kod örnekleri genelde çalışır ama her zaman doğru deđildir.

✓ **Çözüm:** Teknik içerikler için çıktılar test edilmeli; alıntılar varsa özgün kaynađa yönlendirme yapılmalıdır.

## Gizli Veri ve Güvenlik Riski

- Kullanıcı tarafından girilen bilgiler, ChatGPT'nin eğitim verisi **olmasa da** bazı sürümlerde oturumlar analiz amaçlı incelenebilir.

### Tavsiye:

- **Kişisel veri, müşteri verisi, finansal veri** gibi bilgileri asla doğrudan girme.
- Kurumsal kullanım için OpenAI API kullanılırken **gizlilik ve veri işleme politikaları** mutlaka incelenmeli.

### 3. ✓ Tavsiyeler: Profesyonel Kullanım İçin En İyi Uygulamalar

#### 📁 Kendi Prompt Kütüphaneni Oluştur

- Tekrarlayan görevler için (rapor taslağı, kod şablonu, e-posta metni vb.) iyi çalışan promptları sakla ve yeniden kullan.

➡ **Örnek:** prompt-templates.md dosyasında her görev için "rol + biçim + yönlendirme" içeren yapı tutulabilir.

## **İteratif Yaklaşım Kullan**

- Tek seferde mükemmel sonuç bekleme.
  - “Yanıtı sadeleştir”, “yeni örnek ver”, “bunu tabloya dök” gibi tekrarlarla cevabı şekillendir.
-  İyi bir promptçu, cevaptan memnun kalmadığında modelden değil, kendi yönlendirmesinden şüphe eder.

## Kurumsal Kullanımda Persona + Çıktı Şablonları Tanımla

- Özellikle müşteri hizmetleri, eğitim, teknik dokümantasyon gibi işlerde “persona” tanımı çok kritik.

### ➔ Örnek:

- **Persona:** “5 yıllık deneyimli teknik yazar”
- **Çıktı formatı:** Markdown ile başlıklandırılmış belge, her başlık altında max. 4 madde

# CHATGPT İLE ÜRETKEN YAPAY ZEKA



YAPAY ZEKA İLE KONUŞ,  
İŞİNİ GÜÇLENDİR



VEBENDE A.Ş.

CYBER SECURITY, DEVSECOPS, KUBERNETES,  
INFRASTRUCTURE AS CODE, CI/CD, DIGITAL  
TRANSFORMATION, LINUX, CODE QUALITY

[www.vebende.com.tr](http://www.vebende.com.tr) 🔍

TEŞEKKÜRLER

TUNCER KARAARSLAN