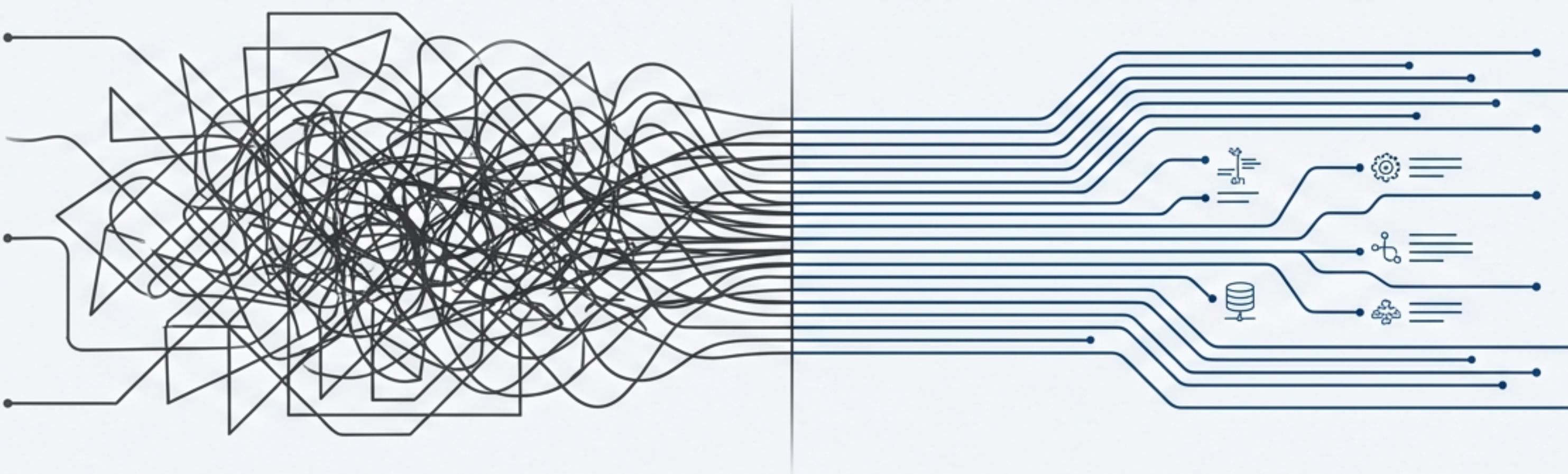


Yapay Zeka Destekli Geliştirmede Kaostan Düzene

Explicit Context-Driven Development (ECDD) ile Tanışın: Proje bağlamını ve yönergeleri versiyon kontrolü altında yönetmek için yapısal bir metodoloji.



Bağlam Kapsüllemesi (Context Encapsulation)

Projeye özgü bilgileri, terimleri ve kararları yapısal olarak tek bir kaynakta toplar. Bilgi dağılığını ve yanlış anlamaları önler.



Sürdürülebilir Direktifler (Sustainable Directives)

AI asistanına verilen talimatları net, güncellenebilir ve tekrar kullanılabilir hale getirir. Tutarlı ve kaliteli çıktı sağlar.



Versiyon Kontrolüne Alınmış Yönergeler (Version-Controlled Guidelines)

Proje bağlamı ve talimatları kod tabanıyla birlikte versiyonlanır. Değişiklikler izlenebilir ve geri alınabilir.

```
# montest.yaml (ESB8 fizesi)
project_name: "MyAIProject"
language: "Python"
frameworks:
  - "Django"
  - "MongoDB"
database: "PostgreSQL"
directives:
  - name: "Code Style"
    description: "Proje standartlarını upelast, doğrulanıcıları açıklayıcı: elasti."
  - name: "Security"
    description: "Kullanıcı girdilerini doğrulayan ve toplanan."}
```

Günümüzün YZ Destekli Geliştirme Süreci Kırılgan ve Geçici



Kayıp Bağlam (Lost Context):

Sohbet pencelerindeki değerli bilgiler ve kararlar, oturum kapandığında kaybolur. Aynı bağlamı tekrar tekrar sağlamak zaman kaybına yol açar.



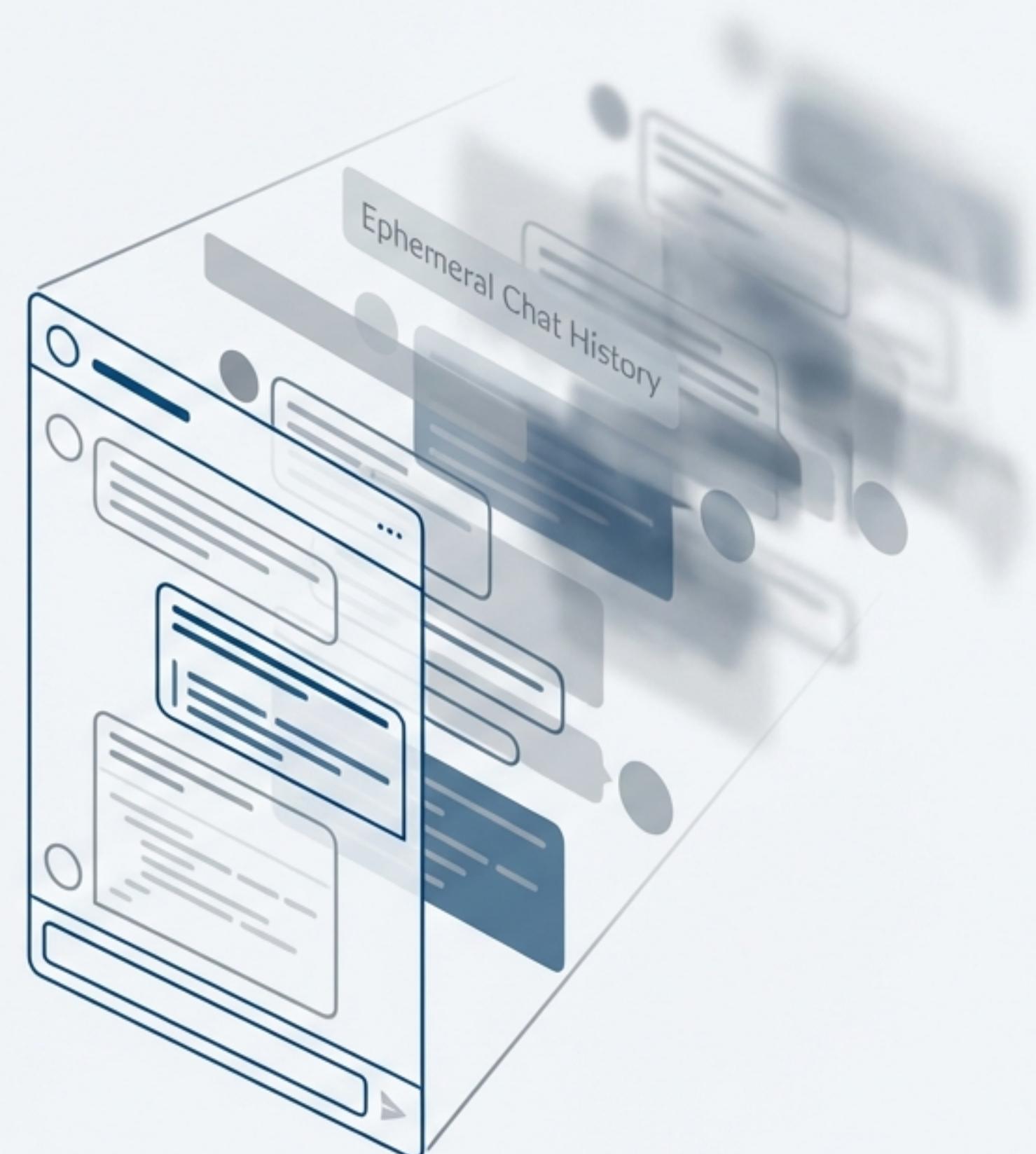
Tekrarlanamaz Sonuçlar (Irrepeatable Results):

Aynı görev için farklı zamanlarda YZ'den alınan yanıtlar tutarsız olabilir. Standartlaşmış bir süreç olmadan kaliteyi ve tutarlılığı sağlamak zordur.



Zorlaşan İşbirliği (Difficult Collaboration):

Bir geliştiricinin YZ ile yaptığı sohbet geçmişini ve ulaştığı sonuçları bir başkasıyla paylaşması ve süreci dahil etmesi neredeyse imkansızdır.



Çözüm: Explicit Context-Driven Development (ECDD)

ECDD, yapay zeka yönergelerini (prompts) ve proje bağlamını birinci sınıf, versiyon kontrollü varlıklar (version-controlled artifacts) olarak ele alan, YZ destekli bir yazılım geliştirme metodolojisidir.

Temel Felsefe

“Otopilot Değil, Yardımcı Pilot”

Bu yaklaşımın, LLM'ler planlama, kodlama ve gözden geçirme süreçlerine yardımcı olur, ancak insan her zaman döngünün merkezindedir ve tüm önemli bağlam proje deposunda (repository) yaşar.



GitHub Copilot



Cursor IDE



Windsurf



Herhangi bir YZ destekli IDE

ECDD'nin Üç Temel Taşı



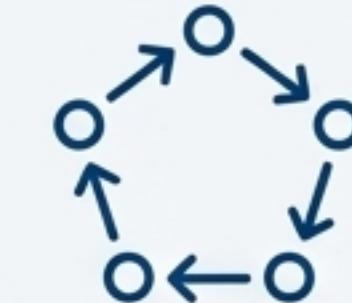
Yönerge Odaklı YZ Yönetişimi Yönetişimi (Prompt-Driven AI Governance)

Proje hedefleri, mimari ve kodlama standartları, kalıcı yönerge dosyalarında yakalanısır ve versiyon kontrol sistemine dahil edilir.



Denetlenebilir Görev ve Durum Yönetimi (Auditable Task & State Management) Inter SemiBold

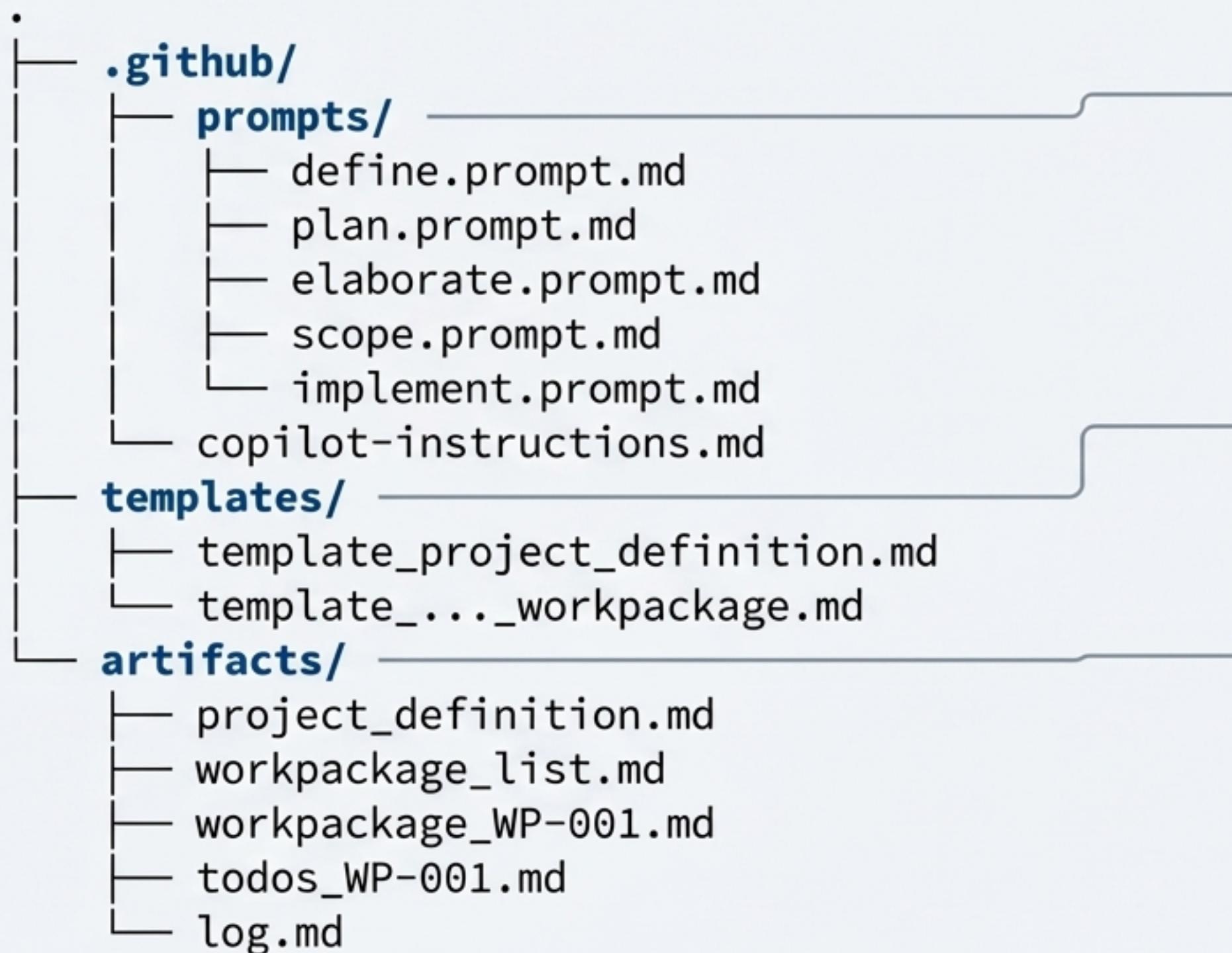
İş paketleri ve çıktıları, geçici sohbet mesajları yerine artifacts/ klasöründe düz metin (plain-text) dosyaları olarak takip edilir. Her adım denetlenebilir ve kalıcıdır.



Beş Aşamalı İş Akışı (Five-Phase Workflow) Inter SemiBold

Her proje, **Tanımla**, **Planla**, **Detaylandır**, **Kapsamı Belirle** ve **Uygula** adımlarından oluşan yapısal bir iş akışını takip eder.

Bir ECDD Projesinin Anatomisi



.github/prompts/

ECDD iş akışını yönlendiren beş temel yönergeyi içerir. Bunlar YZ asistanınızla etkileşime geçeceğiniz “giriş noktalarıdır”.

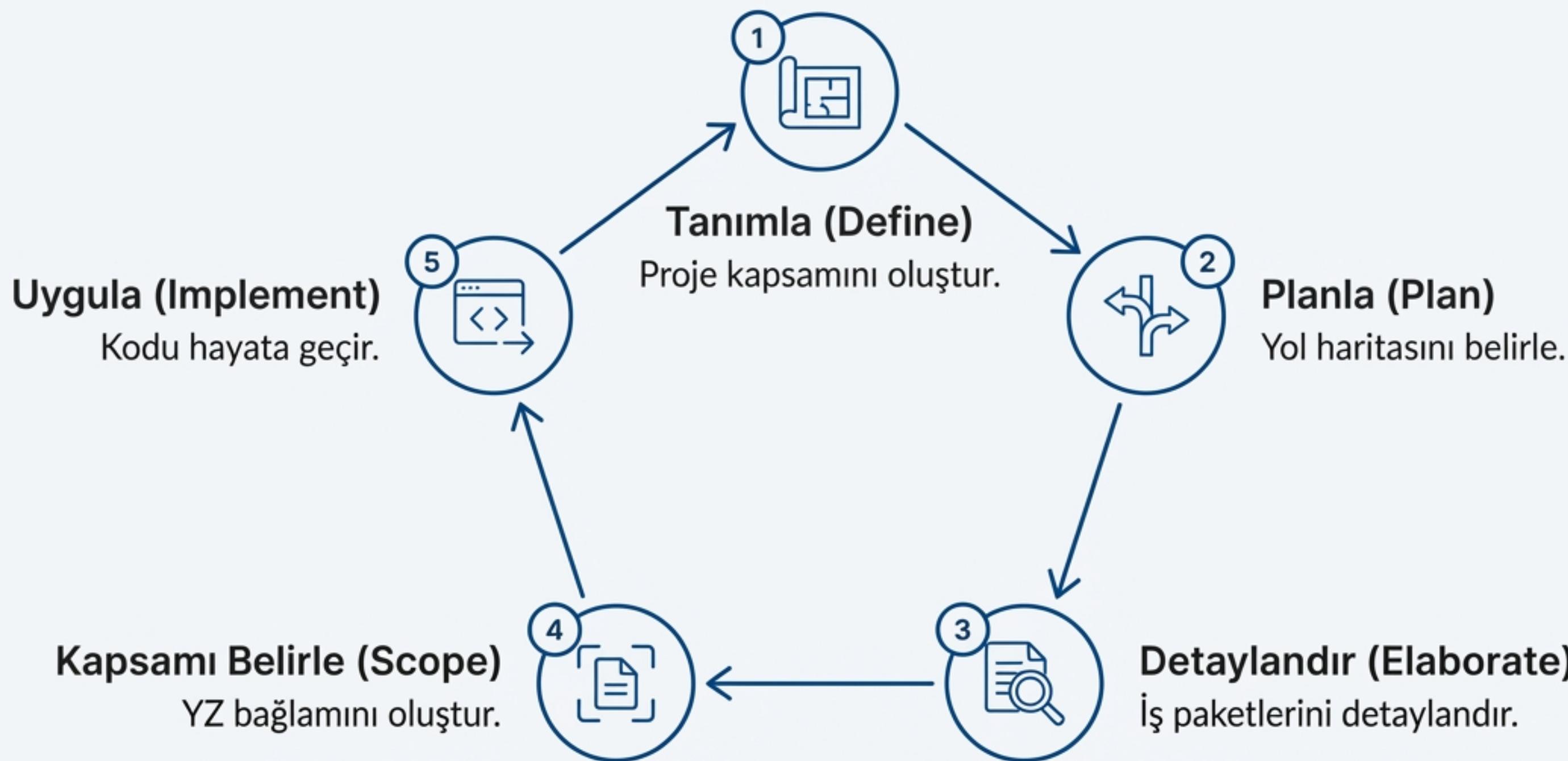
templates/

Proje dokümanları için yapı tanımlayan Markdown şablonları.

artifacts/

Proje bağlamını, planlarını ve logları yakalayan, YZ tarafından YZ tarafından üretilen (ancak manuel olarak düzenlenebilen) dosyalar.

Beş Aşamalı ECDD İş Akışı: Bir Bakışta



Her aşama, bir sonraki için bağlam oluşturan, versiyon kontrollü bir doküman üretir.
Süreç, insan denetiminde ve yinelemeli olarak ilerler.

Faz 1: Tanımla (Define)

Amaç

Projenin kapsamını, hedeflerini ve teknik temelini oluşturmak.

Komut (VS Code)

```
/define [proje tanımınız]
```

Süreç

- YZ, projenizle ilgili netleştirici sorular sorar.
- template_project_definition.md şablonunu doldurur.
- artifacts/project_definition.md dosyasını oluşturur.

Örnek

```
/define a web app named "AI Concepts Dictionary" – ...  
Use the Gemini API key from the GEMINI_API_KEY ...  
...  
frontend (React + Vite), styling (Tailwind), backend (Node + Express) ...
```

Eylem (Action)

Oluşturulan artifacts/project_definition.md dosyasını gözden geçirin ve vizyonunuzu doğru yansittığından emin olmak için düzenleyin.

Faz 2: Planla (Plan)

Amaç

Projeyi yönetilebilir İş Paketlerine (Work Packages - WP) ayırmak.

Komut (VS Code)

```
/plan
```

Süreç

1. YZ, `artifacts/project_definition.md` dosyasını okur.
2. Projeyi bir İş Paketleri listesine ayırtırır.
3. `template_simple_workpackage.md` formatını kullanarak `artifacts/workpackage_list.md` dosyasını oluşturur.

Eylem (Action)

Yol haritasını gözden geçirin. Öncelikleri, bağımlılıkları veya açıklamaları gerektiği gibi ayarlayın.

Faz 3: Detaylandır (Elaborate)

Amaç

Bir İş Paketini detaylı, uygulanabilir bir spesifikasyona dönüştürmek.

Komut (VS Code)

```
/elaborate WP-001
```

Süreç

- YZ, yol haritasını okur ve belirtilen WP'yi seçer.
- Kabul kriterleri, teknik özellikler ve test gereksinimleri içeren detaylı bir şartname oluşturur.
- template_complete_workpackage.md formatını kullanarak artifacts/workpackage_WP-001.md dosyasını oluşturur.

Eylem (Action)

Şartnameyi gözden geçirin.
Uygulama adımlarını, teknik seçimleri veya kabul kriterlerini ayarlayın.

Not: Bu faz, uygulamak istediğiniz her bir İş Paketi için tekrarlanır.

Faz 4: Kapsamı Belirle (Scope)

Amaç

GitHub Copilot'un proje standartlarınızı ve mimarinizi anlaması için bir `copilot-instructions.md` dosyası oluşturmak.

Komut (VS Code)

```
/scope
```

Süreç

- YZ, tüm `artifacts` dosyalarını (`project_definition.md`, `workpackage_list.md` vb.) okur.
- Proje özeti, teknoloji yiğini, kodlama yönergeleri gibi bölümler içeren .github/copilot-instructions.md dosyasını oluşturur.

 **Önemli Not:** VS Code ve GitHub Copilot ile bu dosya, sohbet bağlantısına otomatik olarak dahil edilir.

Eylem (Action)

Yönerge dosyasını gözden geçirin. Projeye özgü kuralları veya standartları ekleyin.

Faz 5: Uygula (Implement)

Amaç

Belirli bir İş Paketi için kodlama işlemini yürütmek.

Komut (VS Code)

```
/implement WP-001
```

Eylem (Action)

Üretilen kodu gözden geçirin, testleri çalıştırın ve gerekirse sürece devam etmek için komutu yeniden çalıştırın.

Süreç

1. YZ, detaylı şartnameyi (workpackage_WP-001.md) okur.
2. artifacts/todos_WP-001.md dosyasının varlığını kontrol eder:
 - Varsa, işaretlenmemiş ilk maddeden devam eder.
 - Yoksa, yeni bir yapılacaklar listesi oluşturur.
3. Liste üzerinden ilerleyerek kodu yazar ve artifacts/log.md dosyasını günceller.

Döngünün Merkezindeki İnsan: Siz

ECDD otonom bir ajan sistemi değildir; insan ve YZ araçları arasında **yapılandırılmış bir işbirliği modelidir**. Sürekli insan denetimi için tasarlanmıştır.



Fazlar Arası Gözden Geçirme:

Her adımdan sonra artifacts/ klasöründeki dosyaları kontrol edin ve düzenleyin.



Kodlamadan Önce Onaylama:

YZ'nin kod yazmasına izin vermeden önce yapılacaklar listesini (todos_WP-XXX.md) onaylayın.



Uygulamaları Doğrulama:

Kodlama sonrası her zaman testleri çalıştırın ve uygulamayı manuel olarak kontrol edin.



İhtiyaç Halinde Yineleme:

İstediğiniz zaman herhangi bir fazı yeniden çalıştırabilir veya herhangi bir dosyayı manuel olarak düzenleyebilirsiniz.

3 Adımda Başlarken

1.



2.

Depoyu Klonlayın (Clone the Repository)

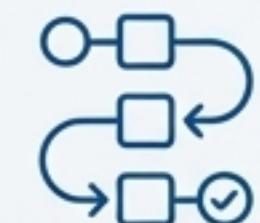
Yeni bir proje için:

```
git clone  
https://github.com/kmkarakaya/ECD  
D.git .ecdd
```

Mevcut bir projeye entegre etmek için:

```
cd your-project-folder && git  
clone ... .ecdd
```

3.



Beş Aşamalı İş Akışını Çalıştırın (Run the Workflow)

Açıklanan beş fazı takip edin. Her fazdan sonra üretilen dokümanları dokümanları gözden geçirmeyi ve düzenlemeyi unutmayın.



- **GitHub Copilot (VS Code):** .github/copilot-instructions.md otomatik olarak kullanılır.
- **Cursor / Windsurf / Diğerleri:** Yönergeleri ve dokümanları YZ'ye manuel olarak referans gösterin.

Başarı İçin İpuçları



'Tanımla' Fazını Atlamayın: project_definition.md her şeyin temelidir.



Yol Haritasını Yineleyin: workpackage_list.md ilk denemede mükemmel olmak zorunda değil.



Her Seferinde Bir WP Detaylandırın: YZ'yi (ve kendinizi) bunalmaktan kaçının.



Log Dosyasını Kullanın: artifacts/log.md neyin inşa edildiğini takip etmek için kritik öneme sahiptir.



Şablonları Özelleştirin: templates/ klasörü size aittir. Yapıyı projenizin ihtiyaçlarına göre uyarlayın.

Sohbet Penceresinin Ötesine Geçin

ECDD, YZ destekli geliştirmeyi geçici sohbetlerden kalıcı, denetlenebilir ve işbirliğine dayalı bir süreç'e dönüştürür.



**Profesyonel yazılım mühendisliği ilkelerini
yapay zeka destekli iş akışınıza taşıyın.**

ECDD'yi bugün deneyin ve projenizi yapılandırın.

github.com/kmkarakaya/ECDD