Agent ekibimi taşıdım, iki aracı yan yana çalıştırdım ve birinci günün sonunda beklemediğim bir sonuçla karşılaştım.

Merhaba,

Geçen yazıda OpenCode Go aboneliğinde DeepSeek limitimin %91’ini 13 günde tükettiğimi, ardından yerel modelle denediğim yolun işe yaramadığını anlatmıştım. Sonunda da Command Code’u araştırdığımı ve kararımı bir sonraki yazıda paylaşacağımı söylemiştim. Bu yazı o söz.

Kısa özet: Command Code’a geçtim, çalıştığım projedeki tüm agent yapımı taşıdım ve bir süre iki aracı birlikte kullanacak şekilde kurdum. Şu an birinci günün sonundayım ve OpenCode’dan daha verimli çalıştığını görmek beni şaşırttı. Bunun henüz bir sonuç değil, bir ilk izlenim olduğunu baştan söyleyeyim. Sonraki günlerde ne olacağını bilmiyorum, deneyip göreceğim.

Çekinceyle başladım

Geçişe iyimser başlamadım. Elimde oturmuş bir agent ekibi vardı: proje yöneticisi, kalite güvence, ön yüz ve arka yüz agent’ları. Kurallar, task’lar ve kararlar proje klasörünün içinde duruyordu. Başka bir araca geçmek, bu düzeni bozma riski demekti. Üstelik kullandığım model aynı kalacaktı: DeepSeek V4.1 Flash, yine ücretli sürüm. Yani fark yaratacak şey model değil, araç olacaktı.

Taşıma: Asıl emek limit takip yapısındaydı

Command Code’un OpenCode’dan gelen kullanıcılar için bir taşıma desteği var. Ticari açıdan bakınca bu beni şaşırttı ve açıkçası hoşuma gitti: rakip araçtan kaçan kullanıcıyı karşılamak için bir kapı açmışlar. Skills, agent’lar, slash komutları, MCP ayarları ve hafıza dosyalarını tek komutla alabiliyorsunuz.

Benim tarafımda asıl zaman alan kısım başka yerdeydi. Daha önce kurduğum günlük, haftalık ve aylık limitleri tutan yapıyı Command Code’a taşımak 4-5 saatimi aldı. Bu yapıyı hem Command Code hem OpenCode ile çalışacak şekilde ayarladım. Yani şu an projem çift başlı: iki araçtan hangisini açarsam açayım aynı kurallara ve

aynı limit takibine bağlı.

Bu noktada bir ders çıkarmam gerekirse: taşıma aracı agent dosyalarını taşıyor, ama sizin özel düzeninizi, yani kurallarınızın birbirine nasıl bağlandığını, limit mantığınızı, agent’lar arası işleyişi yeniden gözden geçirmeniz gerekiyor. Ben ayarı baştan yaptım ve sonrasında düzgün çalıştığını gördüm.

Beni şaşırtan üç şey

1. Mockup’lar tek yerde

Ön yüz tarafında OpenCode’a “mockup oluştur” dediğimde mockup’ı oluşturuyordu, ama ortaya çıkan örnekleri bulmak için bin takla atmam gerekiyordu. Dosyalar arasında dolaşıp hangisi hangisi diye bakmak, karşılaştırıp karar vermeyi gereksiz yorucu hale getiriyordu.

Command Code’da mockup’ları kendi yapısına yüklüyor ve tek bir yerden, tek bir bağlantıyla açabiliyorsunuz. Hangisinde karar kılacağınızı rahatça görebiliyorsunuz. Küçük bir detay gibi duruyor ama tasarım kararı verirken günlük işin akışını değiştiriyor.

2. Oturum yönetimi yükü hafifledi

OpenCode’da limitleri sağlıklı kullanmak için ayrı bir uğraşım vardı: ikinci bir oturumda sürekli komut vererek yeni session’lar açtırmak ve böylece bağlamı düşük tutmak. Geçen yazıda da anlattım, uzun bağlam pahalı ve istek sayısı çok. Bunu yönetmek bana ayrıca yük oluyordu.

Command Code’da ayarı baştan yaptıktan sonra bu sorunun çok daha az yaşandığını gözlemliyorum. İlk günün verisi bu kadar. Uzun vadede aynı şeyi söyleyip söylemeyeceğimi göreceğim.

3. Genel hissiyat: OpenCode geride kalmış gibi

İki aracı yan yana kullanınca bana şöyle bir izlenim geldi: OpenCode geride kalmış gibi. Bunu bir benchmark sonucu gibi değil, kendi iş akışımda hissettiğim şey olarak yazıyorum. Aynı modelle, aynı agent yapısıyla, aynı projede Command Code’un daha verimli çalıştığını gördüm. Önce “ilk gün heyecanı mı” diye düşündüm, bunu

tamamen dışlayamam.

Kâğıt üzerinde iki plan aynı limiti veriyor

Her iki planda da DeepSeek V4.1 Flash için aylık 100 $ limit var; fark, limit dolunca ne olduğunda ve araç çevresindeki yapıda çıkıyor.

Kalem	OpenCode Go	Command Code GOAT
Aylık ücret	10 $	10 $
DeepSeek V4.1 Flash aylık limit	100 $	100 $
5 saatlik limit	20 $	en fazla 20 $
Haftalık limit	50 $	en fazla 50 $
Resmî tahmini aylık istek	216.670	256.000
Tahminin varsayımı (istek başı)	410 yeni girdi, 71.300 cache, 310 çıktı token’ı	~800 yeni girdi, ~50.000 cache, 125-200 çıktı token’ı
Limit dolunca	Ek Kullanım açıksa bakiyen, değilse ücretsiz modeller	Ek krediler kısıtlanmaz; kredi bitince ücretli modeller durur
Ek kredi	Alırken %4,6 + 0,31 $ işlem ücreti	5-500 arası alınır, süresiz geçerli
Ücretsiz yedek modeller	Sınırlı süreli; denemelerimde “limit doldu” (429) hatası aldım	Beam 1.0 (günlük sınır yok), Glyph Cluster ve Crouton Alpha (günde 1.000 istek)
Benim ilk gün deneyimim	Mockup’lar dosyalara dağılıyor, oturum ve bağlamı elle yönetiyorum	Mockup’lar tek bağlantıdan açılıyor, oturum yükü daha az

Rakamlar, 6 Kasım tarihli önceki yazımdaki resmî sayfalardan. Limitler kampanya etiketiyle sunuluyor ve her an değişebilir; kendi panelinizdeki rakama bakın. Son satır benim tek günlük izlenimim, bir ölçüm değil.

Neden henüz kesin konuşmuyorum

Tek günlük veri. Limit tüketimi, hata oranı ve oturum davranışı ancak birkaç hafta sonra anlamlı olur.
Tek kullanıcı, tek iş yükü. Çok agent’lı bir yazılım akışı kullanıyorum. Sizin iş yükünüz farklıysa sonuç değişebilir.
Limitler değişebilir. Geçen yazıda da yazdığım gibi, iki araçta da yüksek limitler kampanya etiketiyle sunuluyor ve her an değişebilir.
Model aynı. Farkı yaratan şey DeepSeek V4.1 Flash değil, çevresindeki yapı. Yani modele göre değil, araca göre bir karşılaştırma bu.

OpenCode’a ne oldu?

Limitlerim önümüzdeki günlerde yeterli olursa, çalıştığım projede OpenCode’a gerek kalmayacak gibi görünüyor. OpenCode’daki kalan yaklaşık %5 limiti ise boşa harcamak istemedim. Hobi olarak geliştirmeye başladığım bir strateji oyununda kullanmaya devam ediyorum. Ana projeden ayrı, riski düşük bir iş olduğu için bunun için ideal.

Bu arada çift başlı yapıyı bilerek bozmadım. Command Code’da bir sorun yaşarsam geri dönebileceğim bir yolun açık olması, geçişi rahat yapmamı sağladı.

Taşımayı düşünenler için notlar

Bu bölüm benim deneyimimden çok, Command Code’un resmî dokümanında okuduklarımdan derlenen kısa bir kontrol listesi. Kendi projenizde mutlaka doğrulayın.

Taşıma komutu Command Code’un komut satırı sürümünde çalışıyor. Masaüstü uygulamasında bu komut bulunmuyor.
Agent dosyalarının başlık alanları iki araçta aynı değil. OpenCode’daki izin ayarları Command Code’da okunmuyor, bu yüzden salt okunur olmasını istediğiniz agent’ların araç listesini elle yazmanız gerekiyor.
OpenCode’daki ek kural dosyası listesi Command Code’da karşılıksız. Kurallarınızı o listeyle yüklüyorsanız bunları hafıza dosyasından referansla göstermeniz gerekir.

@Copyright Serkan Kaya www.serkankaya.net OpenCode’dan Command Code’a Geçtim: İlk Günün Notları | 4

Masaüstü uygulaması beta aşamasında. Önce yedek alın.

Sorular

Bir sonraki yazıda birkaç haftalık verimle döneceğim. O zamana kadar sizden duymak istediklerim:

OpenCode’dan Command Code’a (ya da başka bir araca) geçen oldu mu? Taşıma sürecinde en çok neyi elle düzeltmeniz gerekti?
Çok agent’lı bir kurgunuz varsa limit takibini nasıl yapıyorsunuz?
Uzun oturumlarda bağlamı küçük tutmak için hangi alışkanlıkları edindiniz?
Tasarım çıktılarını (mockup gibi) toplu görüp karşılaştırmanın bir yolunu buldunuz mu?
Yorumlara bırakın, hepsini okuyorum.

Kaynaklar

Command Code, taşıma (import) dokümanı: commandcode.ai/docs/import
Command Code, agent dokümanı: commandcode.ai/docs/agents
Command Code, masaüstü uygulaması: commandcode.ai/docs/desktop
Önceki yazılarım: OpenCode Go ile 13 günde limit tüketimi ve yerel model denemem
Fiyatlar ve limitler sık değişiyor, karar vermeden önce resmî sayfalara bakın.