İstisna işleme nedir?

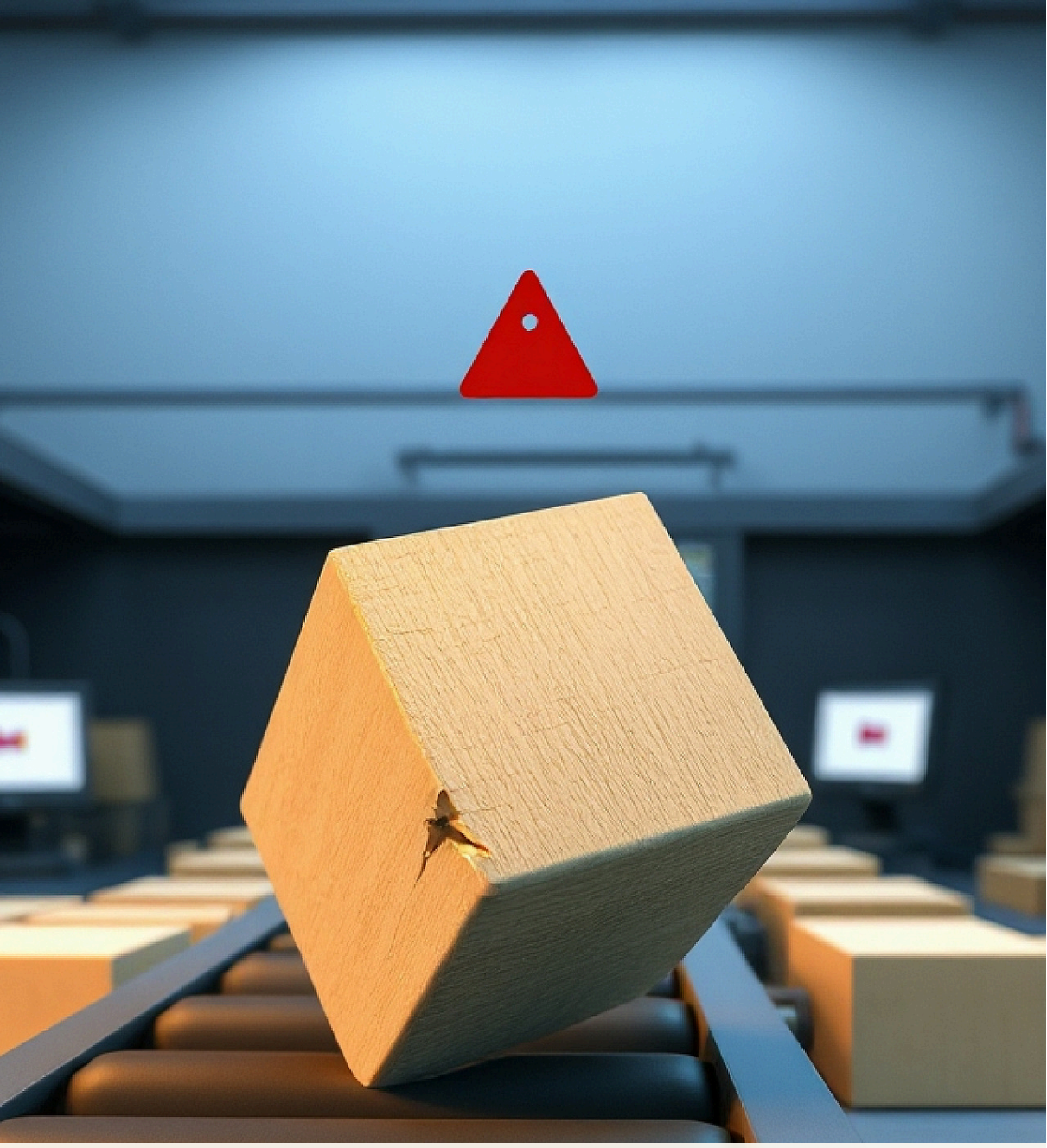


www.yzokul.com.tr

İstisna işleme nedir?



İstisna işleme, bir programın çalışırken ortaya çıkan belirli sorunlara düzenli biçimde yanıt verme yoludur. Her hatayı gizlemek ya da tüm sorunları yok etmekle ilgili değildir. Bunun yerine, hangi sorunu ele alacağına ve nasıl yanıt vereceğine açıkça karar vermene yardımcı olur. Bu yanıt kitabı, bu fikri basit örnekler ve açık işleme kurallarıyla adım adım açıklar.



Bir program çalışır ve planlamadığı sorunlarla karşılaşabilir. Bu sorunlar, program yazılırken değil, çalışırken ortaya çıkar. Örnekler arasında eksik bir dosya, kullanıcıdan gelen hatalı girdi ya da başarısız olan bir ağ bağlantısı yer alır. Bu tür çalışma zamanı sorunlarına istisna denir ve bunlara karşı bir şey yapılmazsa programın normal akışını kesintiye uğratabilirler.



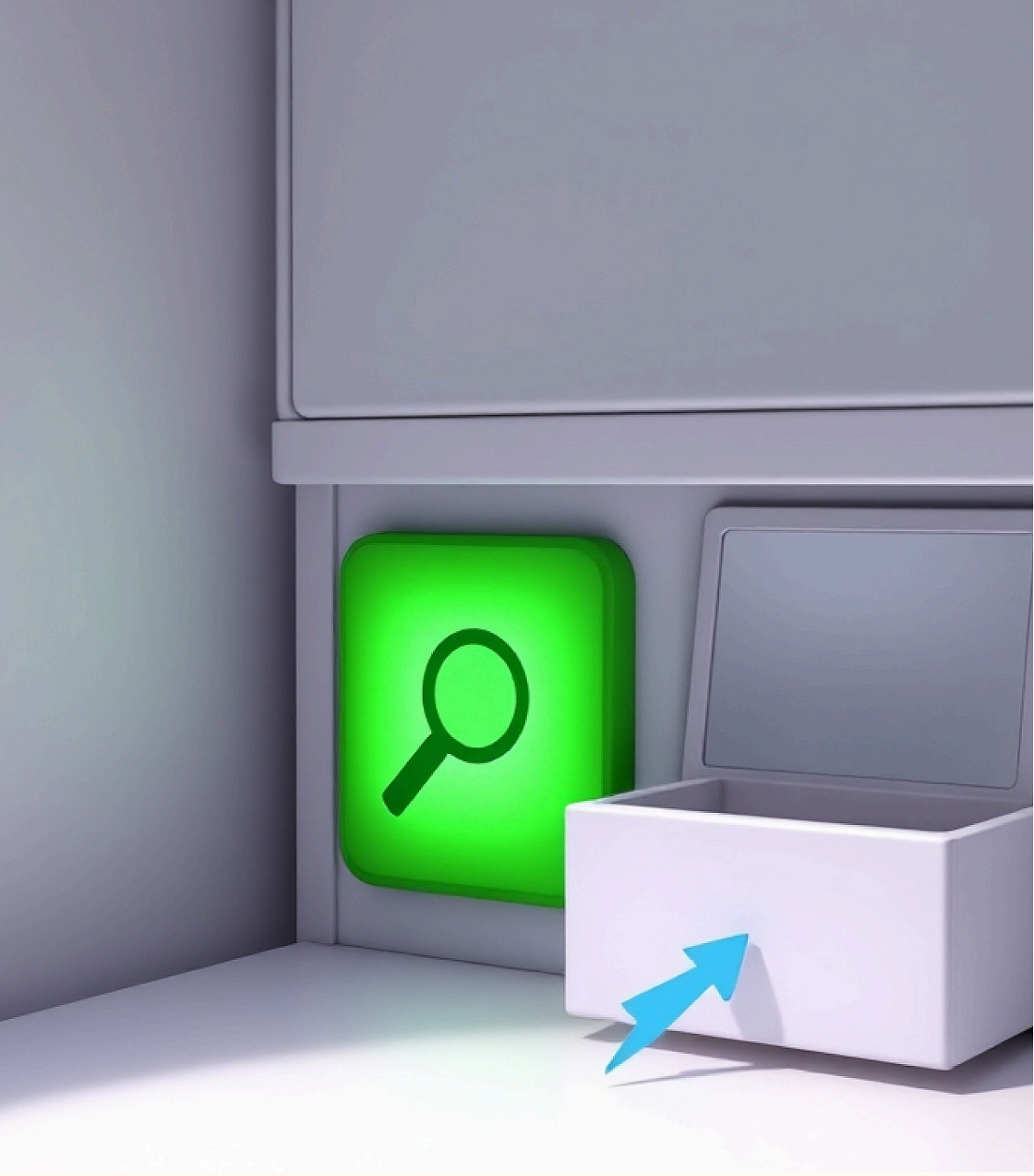
report.txt adlı bir dosyayı açmaya çalışan bir program düşün. Dosya orada değildir çünkü biri onu silmiştir ya da adı yanlıştır. Hiçbir planı olmayan program, bundan sonra ne yapacağını bilemez. Basitçe bozulur, çalışmayı durdurur ve kullanıcıyı ne olduğu ve programın neden kapandığı konusunda şaşkın bırakır.



Bir plan olmadan program aniden durur. Kullanıcı için hiçbir anlamı olmayan, tanıdık olmayan sözcüklerle dolu uzun bir teknik mesaj gösterebilir. Kimse neyin yanlış gittiğini ya da kullanıcının bundan sonra ne yapması gerektiğini açıklamaz. Kullanıcı tahmin etmekle baş başa kalır ve program işine devam edemez. Bu dağınık sonuç, tam olarak istisna işlemenin önlemeyi amaçladığı şeydir.



Bilinen bir sorun ortaya çıktığında, program çökmek yerine planlanmış bir yanıtı izleyebilir. Düzenli olmak, programın paniğe kapılmadan adım adım bundan sonra ne yapacağını bilmesi demektir. Planı belirli bir sorun için önceden yazarsın. Sonra o sorun ortaya çıktığında program senin planını izler ve durumun kontrolünü elinde tutar.



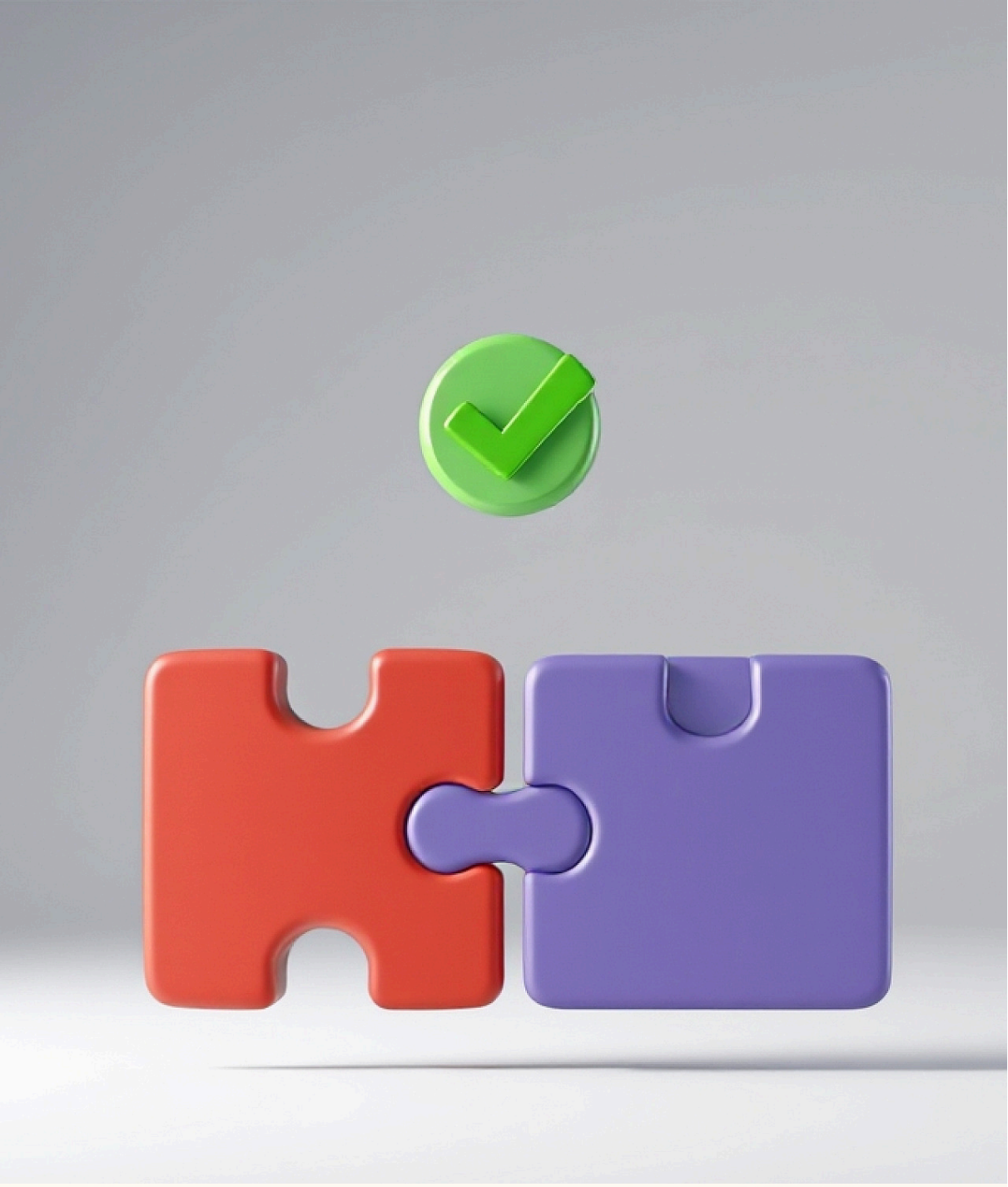
Aynı eksik dosya durumunda program sorunu iyi bir şekilde ele alabilir. Kullanıcıya dosyanın bulunamadığını sade sözcüklerle söyler. Kullanıcı sorunu anlar ve tahmin etmekle baş başa kalmaz. Program daha sonra farklı bir dosya adı isteyebilir ya da temiz ve anlaşılır bir şekilde durabilir. Açık bir mesaj, sorunu görünür ve yönetilebilir kılar.



Herhangi bir yanıt yazmadan önce hangi sorunu ele alacağını açıkça belirlemelisin. Yalnızca herhangi bir sorunu değil, belirli sorunu adlandırırsın; örneğin bulunamayan bir dosyayı. Önce sorunu seçmek, planını odaklı ve dürüst tutar. Bunu atlarsan, sonunda her şeyi aynı anda ele almaya çalışabilirsin ve bu da karışıklığa yol açar.



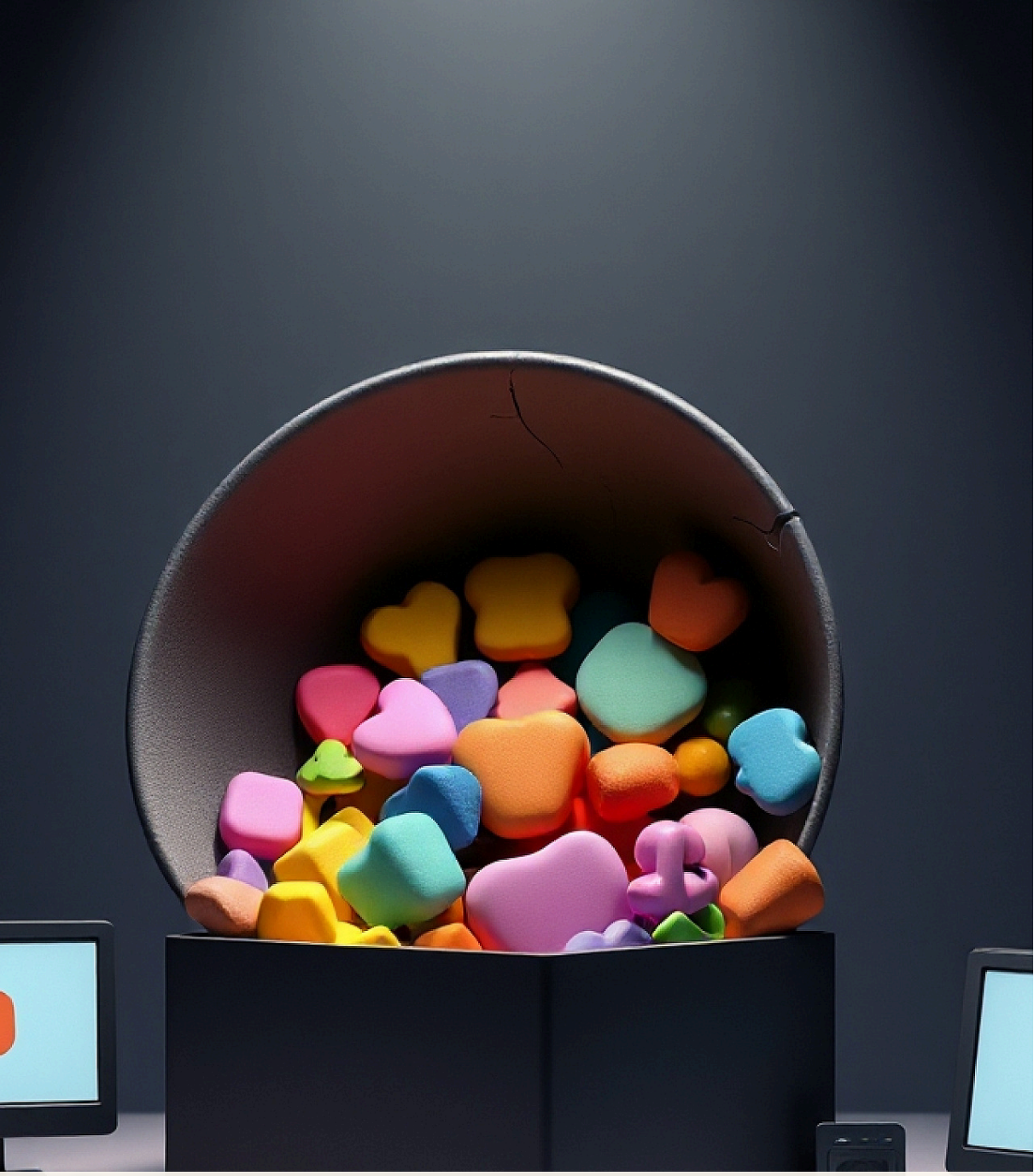
Sorunu seçtikten sonra programın bu konuda ne yapması gerektiğine karar verirsin. Yanıt tesadüfen değil, bilinçli olarak seçilir. Eksik bir dosya için bir mesaj gösterip yeni bir dosya adı isteyebilirsin. Hatalı girdi için kullanıcıdan yeniden denemesini isteyebilirsin. Yanıta karar vermek, programın atacağı tam eylemi yazmak demektir.



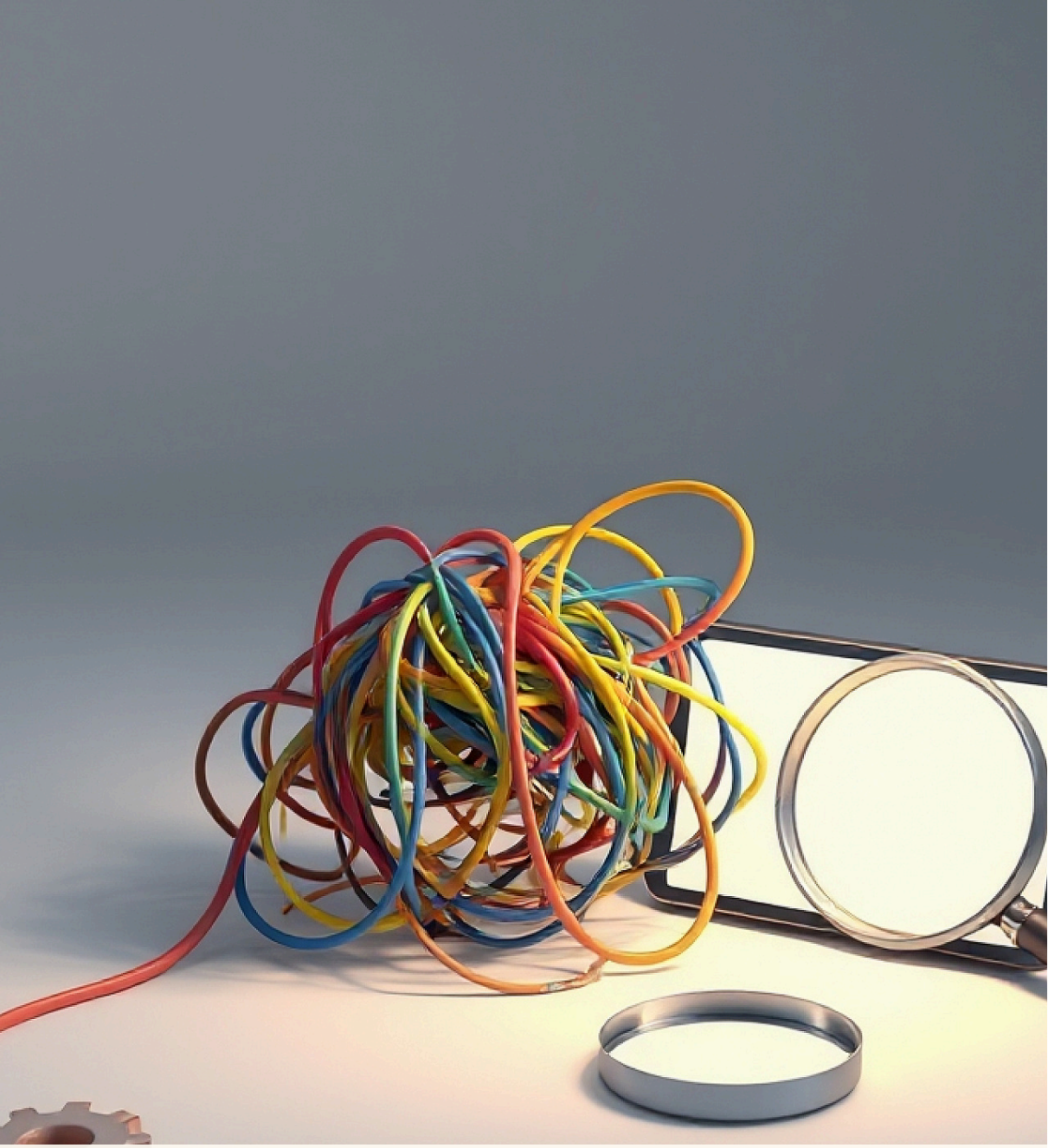
Basit bir eşleşme düşün: belirli bir sorun, belirli bir planlanmış yanıtla eşleştirilir. Eksik dosya sorunu, dosyanın bulunamadığına dair mesajı alır. Ağ arızası gibi diğer sorunlar bu plana dahil edilmez. Bir sorun, bir plan kodunu açık ve anlaşılması kolay tutar. Her plan tam olarak tek bir iş yapar.



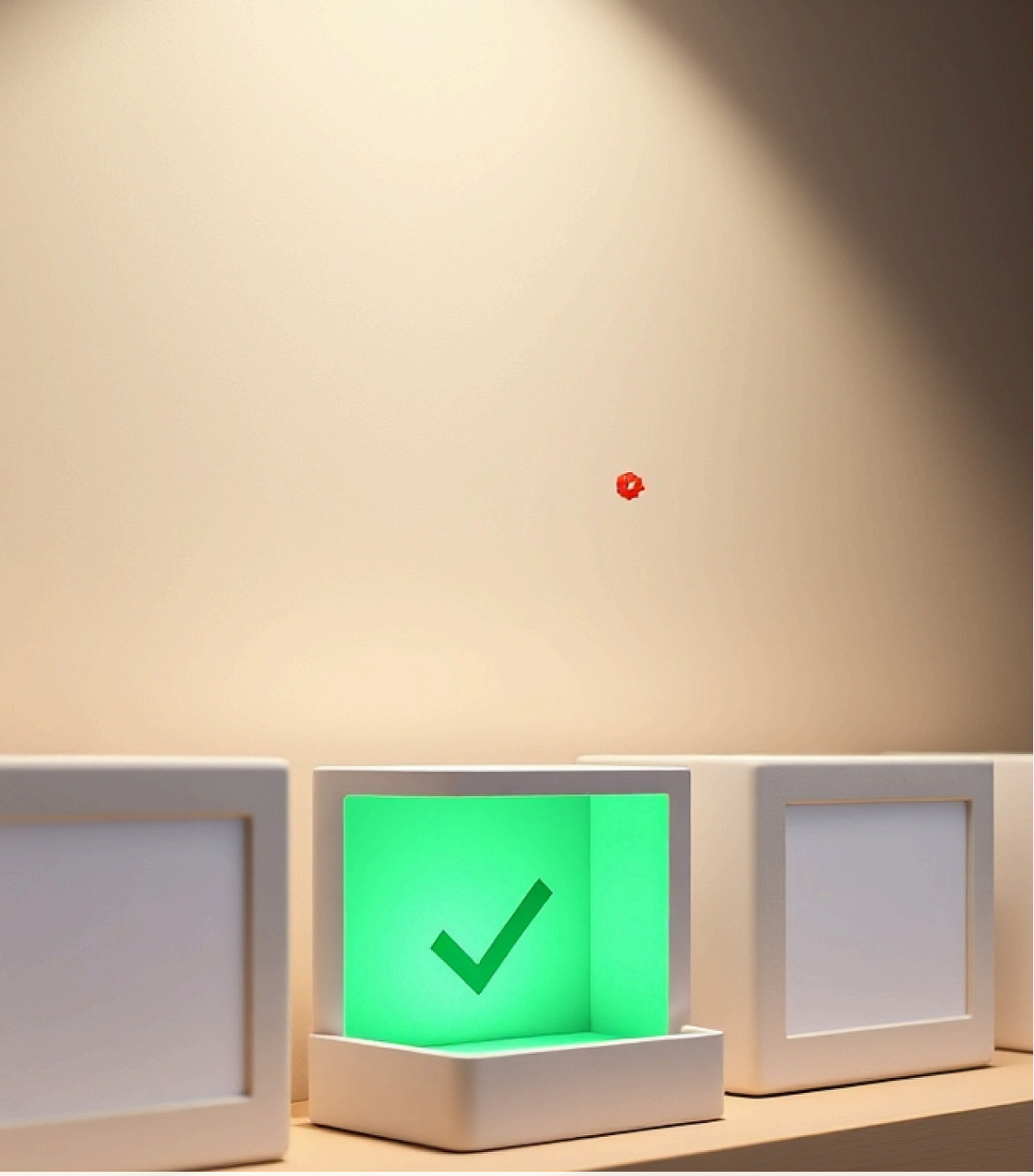
Amaç tüm hataları yok etmek değildir. Bir sorunu ele almak, onu bilerek yanıtlamak demektir; olmamış gibi davranmak değil. Sorunu kabul eder ve mantıklı bir eylem seçersin. Hataları gizlemek düzenli görünebilir, ama gerçek sorunları görünmez bırakır. İstisna işleme, belirli bilinen sorunlara dürüst ve planlı yanıtlar vermekle ilgilidir.



Bazı programlar her hatayı yakalar ve hiçbir şey olmamış gibi devam eder. Bu yaygın bir başlangıç hatasıdır. Program çalışmaya devam eder, ama artık yanlış sonuçlar üretebilir çünkü asıl sorun hiç düzeltilmemiştir. Her şeyi yakalamak güvenli hissettirir, ama aslında gerçek sorunları gizler. Çıktısı yavaş yavaş güvenilmez hale gelirken program iyi görünür.



Yanlış sonuçlar genellikle daha sonra ortaya çıkar ve gerçek nedene kadar izlenmesi zordur. Asıl sorun, çalışmaya devam eden kod katmanlarının altında gömülü kalır. Garip bir sayı ya da bozuk bir rapor görürsün ama nedenini söyleyemezsin. Bu karışıklık zaman ve güven kaybına yol açar. Hataları gizlemek, hata ayıklamayı baştan açıkça ele almaktan çok daha zor hale getirir.



Yalnızca açıkça anladığın ve bir planın olan sorunları ele al. Geri kalanı görünür bırak ki sessizce göz ardı edilmesinler. Bir soruna nasıl yanıt vereceğini bilmiyorsan, onu gizlemektense görünmesine izin vermek daha iyidir. Bu güvenli kural programını dürüst tutar ve gerçek sorunların daha sonra bulunup düzeltilmesini kolaylaştırır.



Planını eksik dosya gibi bilinen tek bir sorun için küçük ve açık tut. Basit ve belirli olmak, her şeyi yakalamaktan iyidir. Kısa bir plan okunması, test edilmesi ve değiştirilmesi kolaydır. Yalnızca anladığını ele aldığında programın öngörülebilir kalır. Küçük planlar ayrıca beklenmedik bir şey olduğunda bunu açıkça gösterir.



Tipik hatayı gözden geçir: her hatayı yakalamak ve hiçbir şey olmamış gibi devam etmek. Bu, gerçek sorunları gizler ve sonradan anlaşılması zor yanlış sonuçlar yaratır. Başlangıç seviyesindekiler bunu genellikle çökmeleri durdurmak için yapar, ama bu açık bir başarısızlığı kafa karıştırıcı bir başarısızlıkla değiştirir. Bunun yerine belirli bir sorun seç, bir yanıta karar ver ve diğer sorunları görünür bırak.



İstisna işleme, bir programın eksik dosya gibi belirli çalışma zamanı sorunlarına düzenli biçimde yanıt vermesini sağlar. Hangi sorunu ele aldığına ve nasıl yanıt vereceğine açıkça karar vermeyi unutma. Sırf devam etmek için her şeyi asla yakalama, çünkü bu gerçek sorunları gizler ve yanlış sonuçlar yaratır. Yalnızca anladığını ele al, planı basit tut ve geri kalanın görünür kalmasına izin ver.