

AI

AI DANIŞMAN

ÜCRETSİZ SÜRÜM

Otomasyon için Yapay Zekâ

Kurmak kolay, ayakta tutmak zor. Süreç çizimi, sessiz yanlışın kökeni, hata senaryosu, bildirim ve sahiplik, veri güvenliği, portföy yönetimi ve yedi günlük plan.

AI Danışman – Ücretsiz Eğitim Serisi

Bu belgeyi dilediğin gibi paylaşabilirsin. Satılamaz, değiştirilemez.

Eylül 2026 · [aidanisman.pages.dev](#) · [t.me/aidanisman](#)

İÇİNDEKİLER

- 01 Otomasyonlar neden ölür
- 02 Hangi iş otomasyona uygun
- 03 Süreci önce çizmek: kutu, ok, karar
- 04 Üç katman: tetik, iş, çıkış
- 05 Yapay zekâ nereye girer
- 06 Sessiz yanlış nereden doğuyor
- 07 Hata senaryosu — asıl iş
- 08 Bildirim ve sahiplik
- 09 Piyasa kurulumu öğretiyor, bakımı öğretmiyor
- 10 Araç seçimi
- 11 Veri güvenliği: otomasyonun görünmeyen tarafı
- 12 İlk otomasyon: yedi gün
- 13 Otomasyon portföyü: birden çoğa geçince
- 14 Yedi günün sonunda

01 Otomasyonlar neden ölür

Otomasyon kurmak, on yıl öncesine göre kıyaslanamayacak kadar kolay. Buna rağmen kurulan otomasyonların çoğu birkaç ay içinde çalışmaz hâle geliyor. Sebep neredeyse hiç "kurulum yanlıştı" değil.

Ölüm sebebi	Nasıl olur	Ne kadar sürede
Sessiz durma	Bir bağlantı kopar, akış durur, kimse fark etmez	Günler — bazen aylar
Girdi değişimi	Form alanı eklenir, tablo sütunu adı değişir	İlk değişiklikte
Yetki bitmesi	Token süresi dolar, şifre değişir	3–12 ay
Sahipsizlik	Kuran kişi ayrılır, kimse nasıl çalıştığını bilmez	İlk arızada
Sessiz yanlış	Akış çalışıyor ama yanlış sonuç üretiyor	Fark edilene kadar

Çıkarım: Listenin tamamı *kurulumla* değil, **kurulumdan sonrasıyla** ilgili. Otomasyonun zor kısmı kurmak değil; bu ders o yüzden otomasyon kurmayı değil, **ayakta kalan otomasyon kurmayı** anlatıyor.

En tehlikelisi: sessiz yanlış

Duran otomasyon er geç fark ediliyor. **Çalışıp yanlış sonuç üreten** otomasyon fark edilmiyor — ve arada geçen sürede yanlış veriyi kaydediyor, yanlış mesajı gönderiyor, yanlış raporu üretiyor. Yapay zekâ içeren otomasyonlarda bu risk özellikle yüksek, çünkü model hata verip durmuyor: *inandırıcı ve yanlış* bir çıktı üretip akışı devam ettiriyor.

02 Hangi iş otomasyona uygun

Yanlış işi otomatikleştirmek, otomasyon kurmamaktan pahalı. Beş ölçüt:

Ölçüt	Soru	Neden
Tekrar	Ayda en az 10 kez oluyor mu	Altındaki işlerde bakım maliyeti kazancı yiyor
Kural	Adımları yazıya dökabiliyor musun	Yazamadığın işi otomatikleştiremezsin
Girdi kararlılığı	Girdi hep aynı biçimde mi geliyor	Değişken girdi, sürekli bakım demek
Hata maliyeti	Yanlış giderse ne olur	Yüksekse insan onayı şart
Sahip	Bozulunca kim bakacak	Cevabı yoksa kurma

KURAL: ÖNCE ELLE YAP

Bir işi **en az on kez elle yapmadan** otomatikleştirme. Elle yaparken fark ettiğin istisnalar, otomasyonun yarısını oluşturuyor. Elle yapmadan kurulan otomasyonlar, ilk istisnada kırılıyor — çünkü istisnanın varlığı bile bilinmiyordu.

İyi ve kötü adaylar

İyi aday	Kötü aday
Gelen taleplerin sınıflandırılıp yönlendirilmesi	Müşteriye denetimsiz otomatik yanıt
Belge ve formlardan yapılandırılmış veri çıkarma	Fiyat ve teklif kararı verme
Tekrar eden raporun ilk taslağının hazırlanması	Raporun onaysız gönderilmesi
Toplantı kaydından not ve madde çıkarma	Kişi hakkında değerlendirme üretme
İçerik türevlerinin hazırlanması	İçeriğin onaysız yayımlanması

Kötü adayların ortak yanı: **çıktı, insan onayı olmadan dışarı çıkıyor**. Otomasyonun sınırı burada.

03 Süreci önce çizmek: kutu, ok, karar

Otomasyon kurmadan önce sürecin **çizilmesi** gerekiyor. Bu, kurumsal dünyada süreç modelleme diye adlandırılan işin en yalın hâli ve tek bir kâğıt yetiyor.

Dört sembol yeterli

Sembol	Ne demek	Otomasyondaki karşılığı
Başlangıç	Süreci ne başlatır	Tetik
Kutu	Yapılan bir iş	Bir adım
Elmas	Bir karar noktası	Dallanma — kuralla mı, modelle mi
Bitiş	Süreç nerede biter	Çıkış

Çizerken ortaya çıkan üç şey

- Gerçek karar noktaları.** "Basit bir iş" sandığın şeyin içinde genelde üç dört karar çıkıyor — ve her karar bir hata kaynağı.

2. **Kim yapıyor.** Her kutunun bir sahibi var: sen, bir çalışan, bir sistem, ya da otomasyon. Karışık süreçlerde bu sütun tek başına aydınlatıcı.
3. **Bekleme noktaları.** Kutular arasındaki boşluklar — onay bekleme, cevap bekleme. Süreyi asıl yiyen yer genelde burası, adımların kendisi değil.

OTOMASYONUN EN BÜYÜK KAZANCI ADIMLARDA DEĞİL, BEKLEMELERDE

Bir işi 10 dakikadan 6 dakikaya indirmek iyi. Ama iki gün süren bir onay beklemesini iki saate indirmek, çok daha büyük bir değişiklik. **Çizim yapmadan bu görünmüyor** — çünkü bekleme kimsenin zamanına yazılmıyor.

Karar noktalarında kritik ayrım

Çizimdeki her elmas için tek bir soru: **bu karar bir kuralla verilebilir mi?**

Kuralla verilebilir	Model gerekiyor
Tutar 10.000'den büyük mü	Bu talep şikâyet mi, soru mu
Bu müşteri kayıtlı mı	Metnin tonu aciliyet içeriyor mu
Dosya PDF mi	Bu belge hangi kategoriye giriyor
Tarih geçmiş mi	Bu iki kayıt aynı kişi mi (yazım farklıysa)

Sol sütun modelden **ucuz, hızlı ve öngörülebilir**. Ve en önemlisi: sol sütun aynı girdide her zaman aynı çıktıyı veriyor. Bu, otomasyonda çok değerli bir özellik.

Çıkarım: İyi bir otomasyon çiziminde genelde bir tane model kullanılan nokta oluyor — en fazla iki. Üçten fazla model çağrısı olan bir akış, ya yanlış tasarlanmış ya da gerçekte iki ayrı otomasyon.

04 Üç katman: tetik, iş, çıkış

Her otomasyon üç katmandan oluşuyor. Sorunlar da hangi katmanda olduğuna göre farklı çözülüyor.

Katman	Ne yapar	Tipik arıza
1. Tetik	Akışı ne başlatır: yeni form, yeni e-posta, saat, yeni satır	Tetik gelmiyor veya iki kez geliyor
2. İş	Veriyi al, dönüştür, karar ver	Girdi biçimi değişmiş, mantık kırılmış
3. Çıkış	Sonucu bir yere yaz veya gönder	Yetki bitmiş, hedef değişmiş

Bir kural: her katman ayrı denenmeli

Otomasyon bozulduğunda "çalışmıyor" demek işe yaramıyor. Üç katmanı ayrı ayrı test edebiliyorsan arıza bulmak dakikalar, edemiyorsan saatler alıyor. Kurarken katmanları birbirinden ayrı çalışabilir biçimde kur.

İKİ KEZ TETİKLENME

En sinsi arıza tipi. Aynı olay iki kez işlenirse iki kayıt, iki mesaj, iki fatura oluşuyor. Çözüm basit ve neredeyse hiç uygulanmıyor: her olayın **benzersiz bir kimliği** olsun ve akış, o kimliği daha önce işleyip işlemediğine baksın. Kurulumda beş dakika, sonrasında saatlerce hata ayıklamadan tasarruf.

05 Yapay zekâ nereye girer

Yapay zekâ otomasyonun tamamı değil, **ikinci katmanın bir parçası**. Doğru yerleştirildiğinde otomatikleştirilemeyen işleri otomatikleştirilebilir hâle getiriyor; yanlış yerleştirildiğinde sistemin en kırılgan parçası oluyor.

Girer	Girmez
Serbest metni yapılandırılmış veriye çevirme	Aritmetik ve hesaplama
Sınıflandırma ve etiketleme	Kesin kuralla çözülebilen dallanma
Özetleme ve madde çıkarma	Kayıt arama ve eşleştirme
Metin taslağı üretme	Onaysız gönderim
Duygu/aciliyet tespiti	Para, hukuk veya kişi hakkında karar

Çıkarım: "Bu adım basit bir kuralla çözülüyor mu?" diye sor. Çözülüyorsa kuralı kullan. Kural, modelden *ucuz, hızlı ve öngörülebilir*. Yapay zekâyı yalnızca kuralın yetmediği yerde kullan — genelde bu, işin belirsiz ve serbest metin içeren tek adımıdır.

Modeli otomasyonda kullanırken üç kural

- Çıktı biçimini zorla.** "JSON döndür, şu alanlar olsun, başka bir şey yazma." Serbest metin çıktısı, sonraki adımı kırıyor.
- Bilmiyorum diyebilisin.** Emin olamadığında uydurmak yerine belirsiz döndürmesini iste ve o durumu insana yönlendir. Bu tek madde, sessiz yanlışların çoğunu engelliyor.
- Çıktıyı doğrula.** Beklenen alanlar var mı, değerler beklenen kümede mi? Yoksa akışı durdur — devam ettirme.

OTOMASYON İÇİ MODEL ÇAĞRISI – İSKELET

Görev: Aşağıdaki [gelen talep metnini] sınıflandır.

Yalnızca şu JSON'u döndür, başka hiçbir şey yazma:

```
{
  "kategori": "fatura" | "teknik" | "satis" | "diger" | "belirsiz",
  "aciliyet": "yuksek" | "normal" | "dusuk",
  "gerekce": "en fazla 15 kelime"
}
```

Kurallar:

- Kategoriden emin değilsen "belirsiz" yaz. Tahmin etme.
- Metinde olmayan bilgiyi çıkarma.
- Kişi adı veya iletişim bilgisi döndürme.

Metin: [icerik]

06 Sessiz yanlış nereden doğuyor

Otomasyonun en tehlikeli arıza tipi "durmak" değil, **çalışıp yanlış sonuç üretmek**. Model içeren akışlarda bunun teknik bir kökeni var ve bilmek korunmayı kolaylaştırıyor.

Modelin dört özelliği

Özellik	Otomasyonda görünümü
Token tahmini	Hata verip durmuyor; makul görünen bir çıktı üretip akışı devam ettiriyor
Bilgi sınırı	Güncel olmayan bilgiyi güncelmiş gibi döndürüyor
Çalışma belleği	Uzun belgede ortayı görmüyor ama yine de özet üretiyor
Yönlendirilebilirlik sınırı	Çok kısıtlı talimatta kuralların bir kısmını uyguluyor

Asıl hatalar çarpışmalarda doğuyor

Tek tek bu özellikler yönetilebilir. Sessiz yanlış, **ikisi aynı anda devreye girdiğinde** oluşuyor:

Çarpışma	Ne olur	Neden fark edilmez
Uzun belge + token tahmini	Belgenin ortasını görmeden özet üretir	Özet akıcı ve kendinden emin
Bilgi sınırı + boşluk doldurma	Bilmediği alanı uydurur	Format doğru, içerik yanlış
Çok kısıt + yönlendirilebilirlik	Kuralların yarısını uygular	Çıktı geçerli görünür, kural ihlali görünmez
Belirsiz girdi + tahmin eğilimi	Emin olmadığı yerde de karar verir	"Belirsiz" seçeneği verilmemiştir

DÖRDÜNCÜ SATIR: EN KOLAY ÖNLENEN VE EN ÇOK ATLANAN

Modele **"bilmiyorum"** diyebilme seçeneği vermezsen, belirsiz girdide de bir cevap üretiyor. Sınıflandırma çıktısına **"belirsiz"** kategorisi eklemek ve o kayıtları insan kuyruğuna yönlendirmek, sessiz yanlışların büyük kısmını tek hamlede kesiyor.

Üç savunma katmanı

- 1. Girdide:** modele verilen içeriği daralt. Tüm belge yerine ilgili bölüm.
- 2. Çıktıda:** biçimi zorla ve doğruyla. Beklenen alan yoksa, değer beklenen kümede değilse, sayı makul aralıkta değilse akışı durdur.
- 3. Sonrasında:** elle düzeltme sayısını izle. Bu sayı, akışın fark *etmediği* hataları gösteren tek göstergedir.

Çıkarım: Üçüncü katman en değerlisi ve en az kurulan. "Hata oranı" yalnızca akışın yakaladığı hataları sayıyor. Sessiz yanlışlar tanım gereği orada görünmüyor — yalnızca **bir insanın sonradan düzelttiği yerlerde** görünüyorlar.

07 Hata senaryosu — asıl iş

Bu bölüm dersin en önemli kısmı. Otomasyonun başarısını belirleyen şey, iyi gittiğinde ne yaptığı değil — **kötü gittiğinde ne yaptığı**.

Dört soru

#	Soru	Cevabı yoksa
1	Ne zaman durur? Hangi durumda devam etmemeli	Yanlış veriyle çalışmaya devam eder
2	Kim haber alır? Hata olunca bildirim nereye gider	Sessizce ölür
3	Ne kaydedilir? Sonradan bakınca ne olduğu anlaşılır mı	Arıza bulunamaz
4	Elle nasıl yapılır? Otomasyon çalışmazken iş nasıl yürür	İş tamamen durur

Dördüncü soru en çok atlanan. Otomasyon bir işi *hızlandırmalı*, o işin tek yapılma yolu hâline gelmemeli. Bozulduğunda işin elle yürüyebilmesi gerekiyor — ve bunu bilen birinin olması.

Üç hata sınıfı, üç farklı davranış

Hata sınıfı	Örnek	Doğru davranış
Geçici	Ağ hatası, servis kısa süre yanıt vermiyor	Bekle ve tekrar dene — artan aralıklarla, sınırlı sayıda
Veri	Beklenen alan yok, biçim bozuk	Durdur, kaydet, bildir. Tekrar deneme — aynı sonucu verir
Yetki	Token süresi dolmuş, erişim kesilmiş	Durdur ve hemen bildir — kendi kendine düzelmez

SONSUZ TEKRAR TUZAĞI

Veri hatasında tekrar denemek, aynı hatayı sonsuz üretiyor — bazı araçlarda bu ücretsiz kotayı bir gecede bitiriyor, e-posta gönderiyorsa yüzlerce mesaj atıyor. Kural: **yalnızca geçici hatalarda tekrar dene, sayıyı sınırla.**

08 Bildirim ve sahiplik

Bildirim: iki tür

Tür	Ne zaman	Nereye
Hata bildirimi	Akış durduğunda, anında	Sahibinin gerçekten baktığı yere — Telegram, e-posta değil
Yaşam sinyali	Günlük veya haftalık özet	Aynı yer

İkincisi sık atlanıyor ama kritik: **hata bildirimi de gelmiyorsa**, otomasyon hiç çalışmıyor olabilir. "Bugün 14 talep işlendi" gibi tek satırlık bir günlük özet, sessiz ölümü fark etmenin en ucuz yolu.

Bildirimde bulunması gerekenler

- Hangi otomasyon
- Ne zaman
- Hangi adımda
- Hata mesajının kendisi
- Etkilenen kayıt (varsa kimliğiyle)

"Bir hata oluştu" diyen bildirim, bildirim değil. Bu beş bilgi olmadan sahibin otomasyonu açıp baştan araştırması gerekiyor.

Sahiplik

SAHİPSİZ OTOMASYON KURULMAZ

Her otomasyonun tek bir sahibi olmalı ve sahip üç şeyi bilmeli: **ne yaptığını, nasıl durdurulacağını, bozulduğunda işin elle nasıl yürüyeceğini**. Sahibi olmayan otomasyon, kuran kişinin ayrıldığı gün ölü sayılır — çalışmaya devam etse bile, kimse dokunamadığı için ilk değişiklikte kırılır ve kimse tamir edemez.

Her otomasyon için tek sayfa

Sahibin bıraktığı belge en az şunları içermeli: ne işe yarıyor · neyle tetikleniyor · hangi servislere bağlı · hangi hesap/yetkiyle çalışıyor · nasıl durdurulur · bozulunca iş elle nasıl yapılır. Altı satır, ve bir yıl sonra bu otomasyonu kurtaran şey bu.

09 Piyasa kurulumu öğretiyor, bakımı öğretmiyor

Otomasyon eğitimlerine baktığında bir örüntü görünüyor. Giriş seviyesi kurslar şunları anlatıyor: arayüz, verinin nasıl yapılandığı, adımların yapılandırılması, akışın kaydedilip paylaşılması.

Anlatmadıkları: **hata senaryosu, bildirim, sahiplik ve devir**.

Çıkarım: Bu bir eksiklik değil, bir odak tercihi — kurulum öğretilebilir bir beceri, bakım ise bir disiplin. Ama sonucu şu oluyor: insanlar otomasyon kurmayı öğreniyor ve kurdukları otomasyonlar birkaç ay içinde ölüyor. Bu rehberin tamamı o boşluğu doldurmak üzerine kurulu.

Kurulum becerisi ile bakım disiplini

Kurulum (öğretiliyor)	Bakım (öğretilmiyor)
Arayüzü kullanmak	Hata sınıflarını ayırmak
Adımları bağlamak	Ne zaman durulacağına karar vermek
Veriyi dönüştürmek	Bildirim kurmak ve <i>test etmek</i>
Akışı çalıştırmak	Yaşam sinyali kurmak
Paylaşmak	Devir belgesi yazmak
Yeni akış eklemek	Ölü akışı silmek

Bakım disiplininin dört alışkanlığı

- Her akışın bir sahibi var** ve sahip üç şeyi biliyor: ne yaptığını, nasıl durdurulacağını, bozulunca işin elle nasıl yürüyeceğini.
- Her akışın bir devir belgesi var** — altı satır yeterli.
- Her akışın yaşam sinyali var** — günlük veya haftalık tek satırlık özet.
- Ayda bir ölü akışlar siliniyor.** Kullanılmayan akış, ekipte "bu işler yürümüyor" algısı yaratıyor.

DÖRDÜNCÜ ALIŞKANLIK EN ÇOK DİRENÇ GÖRÜYOR

Kurulmuş bir otomasyonu silmek, emeğin boşa gittiği hissi veriyor. Ama kullanılmayan akış bedava değil: kota tüketiyor, yetki tutuyor, bakım listesini şişiriyor ve bir gün beklenmedik biçimde tetiklenebiliyor. **Silmek bir başarısızlık değil, bir ölçüm sonucudur.**

10 Araç seçimi

Araç, otomasyonun en az önemli kararı — ama en çok tartışılanı. Üç aile var:

Aile	Mantığı	Uygun olduğu durum
Görsel akış araçları Zapier, Make	Kutuları bağlarsın, altyapı onlarda	Hızlı başlangıç, az sayıda akış, teknik bilgi az
Kendi sunucunda çalışanlar n8n gibi	Aynı mantık, ama senin ortamında	Çok sayıda çalıştırma, veri kontrolü, maliyet
Kod betik + zamanlayıcı	Doğrudan yazarsın	Karmaşık mantık, sürüm takibi, tam kontrol

Fiyatlandırma modelini anlamak, fiyatı bilmekten önemli

Araçların ücretlendirmesi genelde **çalıştırma sayısı** veya **adım sayısı** üzerinden işliyor. Bu ayrım, sonsuz tekrar hatasında faturanın büyüklüğünü belirliyor. Ayrıca ücretsiz katmanların sınırları sık değişiyor — karar vermeden önce sağlayıcının güncel sayfasından kontrol et.

Çıkarım: İlk otomasyonunu *elindeki* araçla kur. Araç araştırmasına harcanan hafta, kurulup çalışan tek bir otomasyondan daha az öğretiyor. Araç değiştirmek kolay; hangi işin otomasyona uygun olduğunu öğrenmek zor.

11 Veri güvenliği: otomasyonun görünmeyen tarafı

Sohbet arayüzünde ne girdiğine sen karar veriyorsun. Otomasyonda **akış karar veriyor** — ve genelde sen kurduktan sonra bir daha bakmıyorsun.

Üç sızıntı yolu

Yol	Nasıl olur	Önlem
Girdi genişlemesi	Akış "tüm kaydı" modele gönderiyor; kayıta kişisel veri de var	Modele yalnız gereken alanları gönder, tüm nesneyi değil
Ek dosyalar	E-postadan gelen ek otomatik olarak işleniyor; içinde ne olduğu bilinmiyor	Dosya tipi ve boyut filtresi; şüpheliyi insana yönlendir
Log ve hata kaydı	Hata mesajı, işlenen veriyi olduğu gibi kaydediyor	Loglarda hassas alanları maskeley

ÜÇÜNCÜSÜ EN SİNSİ

Bir hata olduğunda çoğu araç, işlenen veriyi hata kaydına yazıyor — ve hata kayıtları genelde daha geniş erişimli, daha uzun saklanan ve daha az düşünülen yerler. Kişisel veri işleyen bir akışta **logların da veri sızıntısı yüzeyi** olduğunu hesaba katmak gerekiyor.

Otomasyon kurmadan önce dört veri sorusu

1. Bu akışta **kişisel veri** var mı? (Ad, e-posta, telefon, T.C. no — müşteri kaydı neredeyse her zaman içeriyor.)
2. Modele giden veriyi **daraltabilir miyim**? Genelde evet: tüm kayıt değil, üç alan.
3. Sonuç **nereye yazılıyor** ve oraya kimler erişiyor?
4. Hata durumunda **ne kaydediliyor**?

Çıkarım: İkinci soru çoğu riski tek başına çözüyor. "Müşteri talebini sınıflandır" işi için modele müşterinin adı, telefonu ve geçmiş siparişleri gerekmiyor — **yalnız talep metni gerekiyor**. Akışı kurarken varsayılan davranış her şeyi göndermek oluyor; bilinçli davranış daraltmak.

Yurt dışına aktarım — otomasyonda daha kritik

Elle kullanımda ne gönderdiğine her seferinde karar veriyorsun. Otomasyonda akış her gün, her kayıt için gönderiyor. Yani **süreklilik arz eden bir aktarım** kuruyorsun — ve bu, tek seferlik kullanımdan farklı bir hukuki kategori.

Pratik sonuç: kişisel veri içeren bir akış kurmadan önce bu sorunun cevabı netleşmeli. Kurumsal bağlamda bunun ayrıntısı *Kurumsal İşyerleri için Yapay Zekâ* dersinde; bireysel ve küçük ölçekte ise en pratik çözüm aynı: **kişisel veriyi akışa hiç sokmamak**.

12 İlk otomasyon: yedi gün

Gün	İş
1	İşi seç. Beş ölçütten geçsin. En sıkıcı ve en sık tekrarlayanı seç — heyecan verici olanı değil.
2	İşi elle yap ve her adımı yaz . İstisnaları ayrı bir listeye not et.
3	Üç katmanı belirle: tetik ne, iş ne, çıkış nereye. Kâğıda çiz.
4	Kur — ama çıkışı kendine yönlendirerek . Gerçek hedefe henüz yazma.
5	Hata senaryosunu ekle: dört sorunun cevabını kur. Bildirimi test et — bilerek bozup bildirimin geldiğini gör.
6	Gerçek çıkışa bağla. İlk hafta çıktıları elle kontrol et.
7	Tek sayfalık belgeyi yaz. Sahibini belirle (muhtemelen sen). Yaşam sinyali bildirimini kur.

Beşinci gün, çoğu kişinin atladığı ve otomasyonun ölüp ölmeyeceğini belirleyen gün. Bildirimi **bilerek bozarak test etmek**, kurulumun tamamından daha değerli — çünkü test edilmemiş bildirim, çalışmayan bildirimdir.

13 Otomasyon portföyü: birden çoğa geçince

Tek otomasyon yönetilebilir. Beş otomasyon bir portföy ve farklı kurallar gerektiriyor.

Portföy kayıt tablosu

Akış	Sahibi	Sıklık	Risk	Son kontrol	Durum

Portföy kuralları

Kural	Neden
Her akışın tek sahibi	İki sahip, sıfır sahip demek
Bildirimler tek yere	Farklı yerlere düşen bildirimler kaybolur
Ayda bir portföy taraması	Ölü akışları bulmanın tek yolu
Aynı işi iki akış yapmasın	Portföy büyürken en sık oluşan çakışma
Bir akış diğerini tetikliyorsa yazılı olsun	Zincirleme arızaların kaynağı
Yeni akış eklemekten önce ölü akış sil	Portföy sınırsız büyüyemez

ZİNCİRLEME BAĞIMLILIK

Portföy büyüdükçe akışlar birbirini tetiklemeye başlıyor ve bu bağımlılıklar genelde **yazılı hiçbir yerde durmuyor**. Sonuç: bir akış bozulduğunda üç akış birden çalışmıyor ve sebebi bulmak saatler alıyor. Kural basit: bir akış başka bir akışın çıktısını kullanıyorsa, ikisinin devir belgesinde de yazar.

Aylık tarama — beş dakika

1. Hangi akış bu ay hiç çalışmadı? (Ölü olabilir ya da bozuk.)
2. Hangi akıştan bildirim geldi, kapatıldı mı?
3. Hangi akışta elle düzeltme arttı?
4. Sahibi değişen veya ayrılan var mı?
5. Silinebilecek akış var mı?

14 Yedi günün sonunda

Çıktı	Ne işe yarar
Süreç çizimi	Karar noktalarını ve beklmeleri görünür kılar
Çalışan tek akış	Kurulum becerisinin kanıtı
Hata senaryosu	Dört sorunun cevabı yazılı
Test edilmiş bildirim	Bilerek bozup geldiğini gördün
Yaşam sinyali	Sessiz ölümü fark etmenin en ucuz yolu
Tek sayfalık devir belgesi	Sahipsizliğe karşı tek koruma

Son çıkarım: Bu listede "kaç otomasyon kurdum" yok. Bir otomasyonun değeri kurulmuş olmasında değil, **altı ay sonra hâlâ çalışıyor olmasında**. Ve bunu belirleyen şey kurulum kalitesi değil — beşinci gün yapılan iş: hata senaryosu, bildirim ve sahiplik.

Buradan sonrası

Otomasyon seçim puanlaması, hata senaryosu şablonu, model çağrısı desenleri ve çıktı doğrulama, bildirim metinleri, tek sayfalık devir belgesi, izleme paneli ve on hazır otomasyon tarifi 'nda.

t.me/aidanisman · aidanisman.pages.dev

Bu belgede araç isimleri örnek olarak geçmektedir; hiçbirisiyle ticari ilişkimiz yoktur. Ücretsiz katman sınırları ve fiyatlandırma modelleri sık değişmektedir — karar vermeden önce sağlayıcının güncel sayfasını kontrol edin. Belge eğitim amaçlıdır.