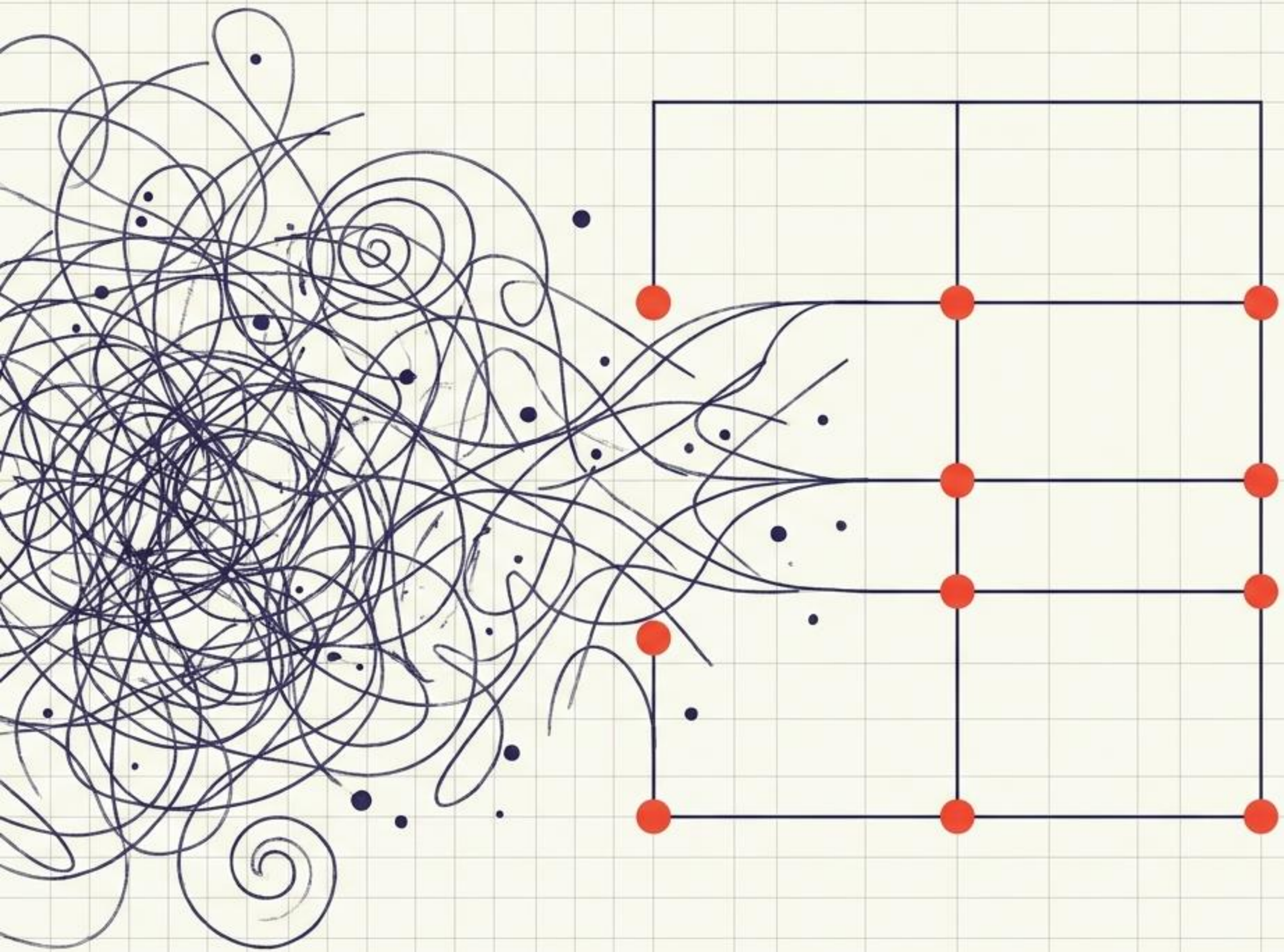


Yazılım Proje Yönetimi

Hafta 3: **Paydaş ve İletişim Yönetimi**

(Yazılımın Görünmez Mimarisi)

Projeler Kötü Kod Yüzünden Değil, Yönetilmeyen İnsanlar Yüzünden Başarısız Olur

DIAGNOSTIC DATA

%30 - %37

Yazılım projelerinin tam başarı oranı (Standish CHAOS Raporu).

2,5 Kat

Etkili paydaş katılımı sağlayan projelerin orijinal hedeflerine ulaşma olasılığı (PMI Verisi).

%77

İletişim planlamasına yatırım yapan kurumların projelerini zamanında bitirme oranı.

SYSTEM ALERT

⚠ ⚠

Teknik altyapı ne kadar kusursuz olursa olsun; **yanlış hizalanmış beklentiler** ve **kaotik iletişim**, projenin **çökmesine** (crash) neden olan asıl hatalardır (bug).

İnsanları yönetmek, kodu yönetmekten daha karmaşık bir algoritma gerektirir.

⚠ ⚠

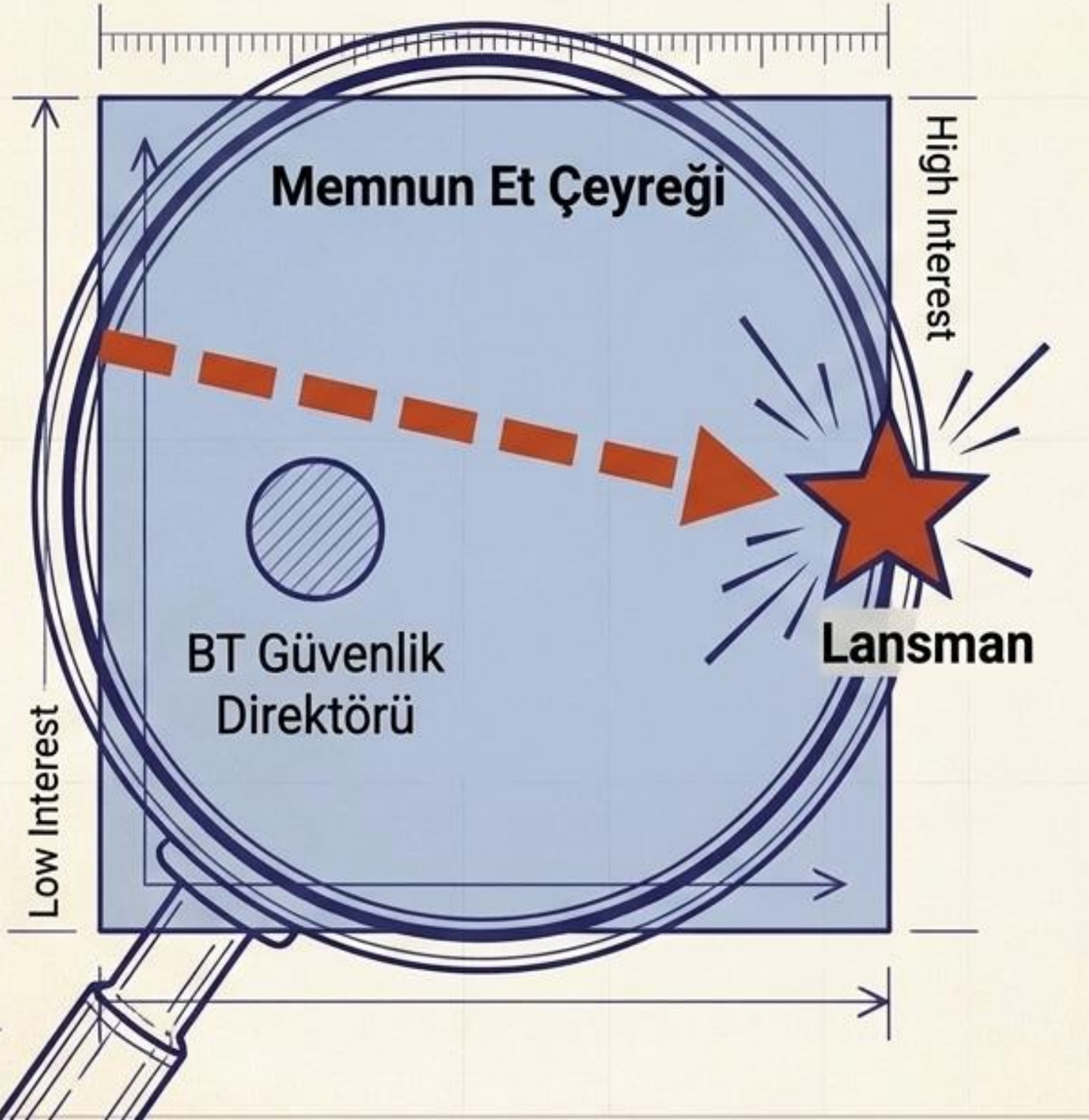
Paydaş Rollerindeki İnce Çizgi: Kim, Neyi, Neden İster?

			
Sponsor (Bütçe ve Yetki)	**Müşteri / İş Sahibi (Hedef ve Teslimat)**	**Son Kullanıcı (Deneyim ve Fonksiyon)**	**Proje Ekibi (Teknik Mimari ve Üretim)**
• **Odak:** Finansal geri dönüş ve stratejik uyum. • **Ana Soru:** Bu yatırımın getirisi (ROI) nedir?	• **Odak:** İş problemlerinin çözümü ve pazar avantajı. • **Ana Soru:** Bu yazılım temel iş sorunumuzu çözüyor mu?	• **Odak:** Günlük iş akışı ve kullanılabilirlik. • **Ana Soru:** İşimi kolaylaştırıyor mu, arayüz anlaşılır mı?	• **Odak:** Kod kalitesi, mimari bütünlük ve netlik. • **Ana Soru:** Gereksinimler net ve uygulanabilir mi?

İnsanları Sınıflandırma Algoritması: Güç–İlgi Matrisi (Mendelow)



En Tehlikeli Çeyrek ve Sessiz Katiller



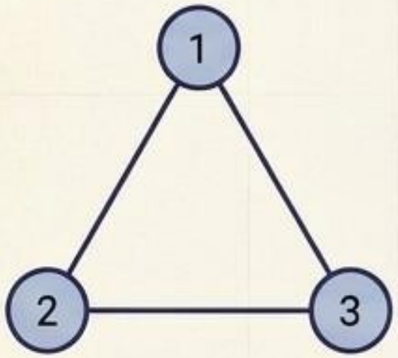
- **Sistem Zafiyeti:**
- Projeler genellikle Memnun Et çeyreğinde sessizce ölür. Yüksek güçlü bir paydaş başlangıçta ilgisizdir, ancak lansmana bir hafta kala projenin kendi alanına dokunduğunu fark edip projeyi durdurabilir.

- **Çatışma Yönetimi Protokolleri:**
  1. Sınıflandırma matrisini gizli tutun (Bu politik bir haritadır).
  2. Gayri resmi etkileyicileri (informal influencers) haritaya dahil edin.
  3. Gizli çıkar çatışmalarını erken yüzeye çıkarın; geç gelen sürprizler her zaman çok daha pahalıdır.

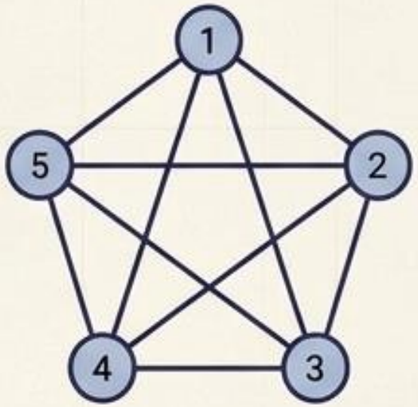
İletişimin Matematiği: Neden Plan Yapmak Zorundayız?

$$\text{İletişim Kanalı Sayısı} = N(N-1) / 2$$

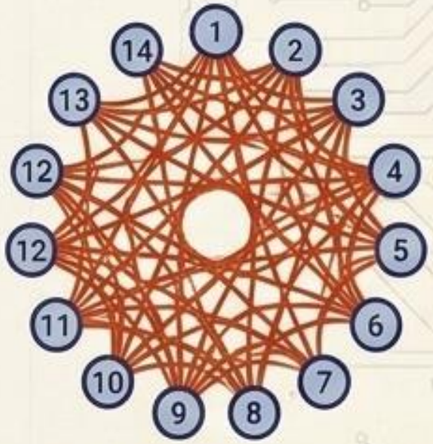
(N = Paydaş Sayısı)



N=3 (3 Kanal)



N=5 (10 Kanal)

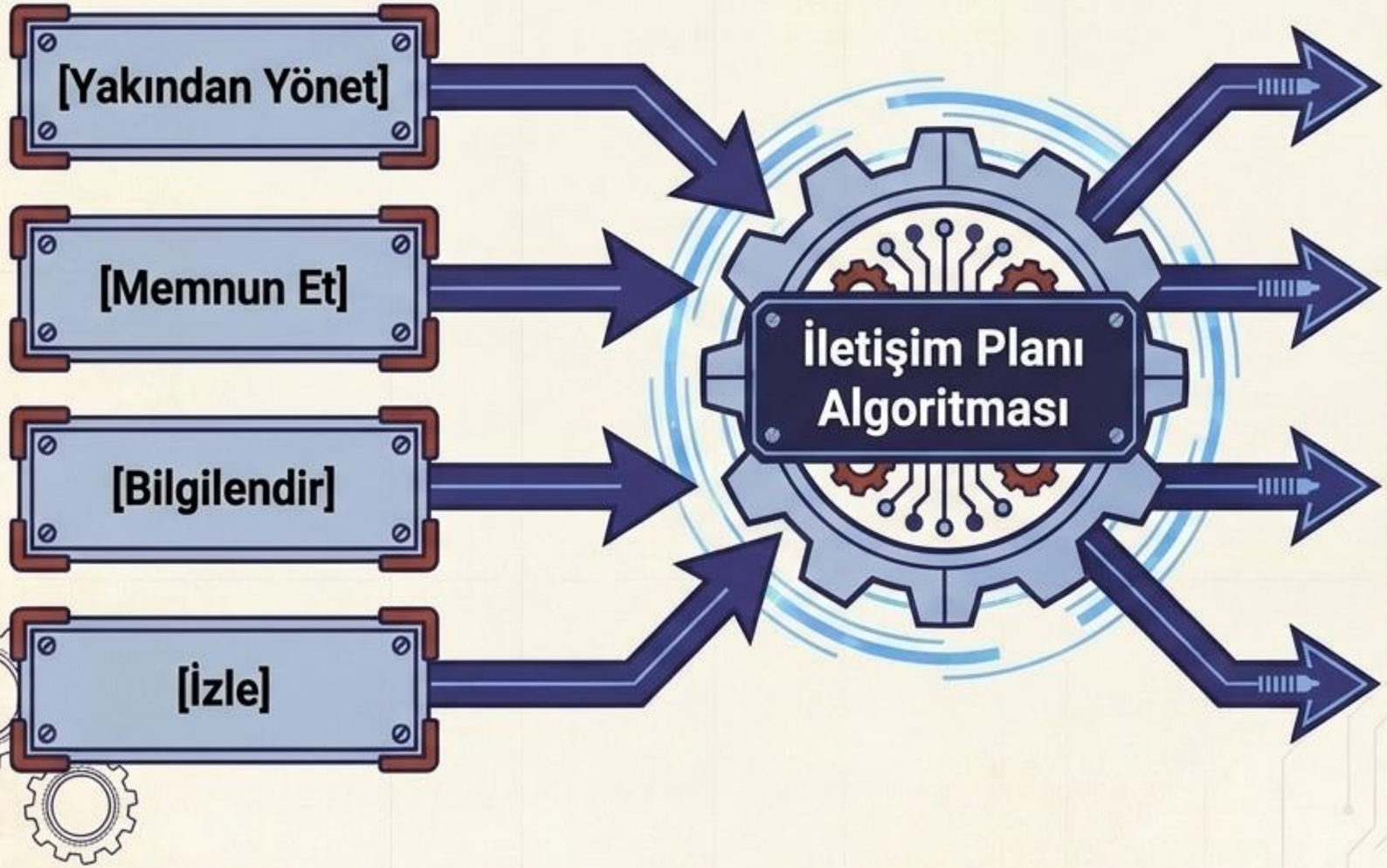


N=14 (91 Kanal)

Gerçeklik: Ekibe katılan her yeni kişi, iletişim karmaşıklığını doğrusal (lineer) değil, katlanarak (eksponansiyel) artırır.
Sonuç: Merkezileştirilmiş bir plan yoksa, bilgi akışı kısa sürede gürültüye (noise) dönüşür ve kritik kararlar kaybolur.

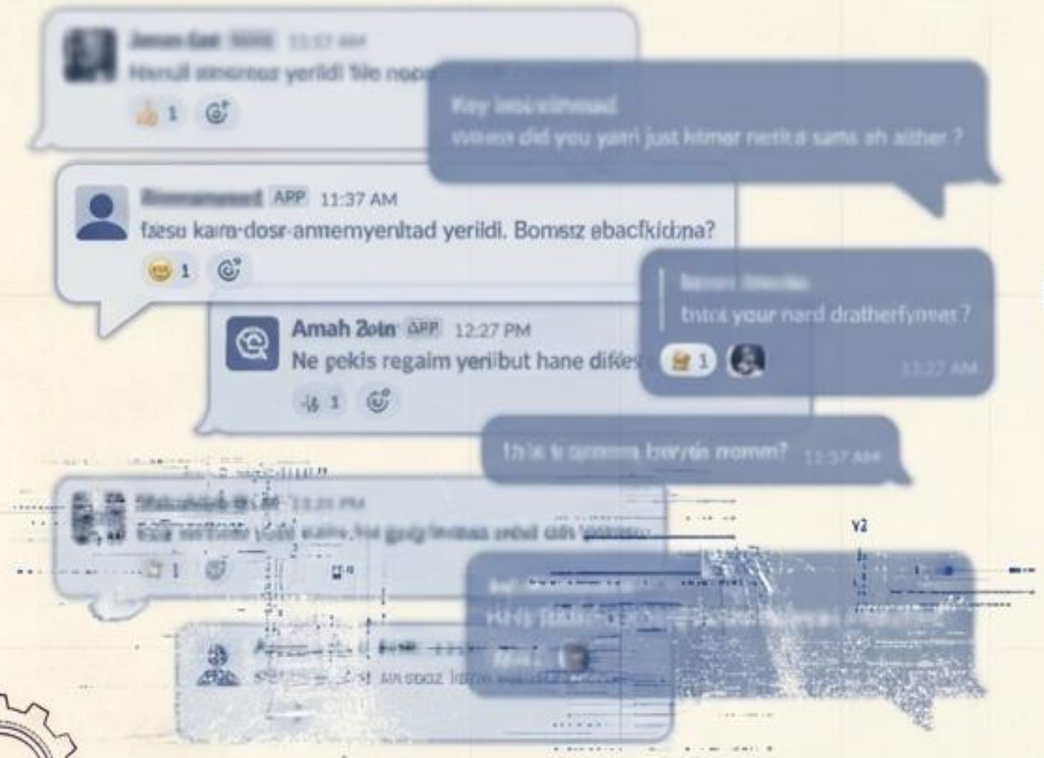
Kaosu Evcilleştirmek: Matrisi Plana Çevirmek

İletişim planı; **kimin, neyi, ne zaman ve nasıl bileceğini standartlaştıran algoritmanızdır.** Yanlış kanal, mesajın (ve dolayısıyla projenin) kaybolması demektir.



Versiyon Kontrolü Olarak Karar Kayıtları (Decision Log)

Sohbet (Chat), tek gerçeklik kaynağı (Source of Truth) olamaz.



Karar Kaydı Anatomisi:

1. Ne karar verildi?
2. Kim onayladı? (Accountable)
3. Gerekçesi (Rationale) neydi?
4. Tarih ve Etki Alanı nedir?

Etkili toplantılar sadece karmaşıklık, çatışma çözümü ve kritik kararlar içindir. Durum güncellemeleri yazılı olmalıdır.

Unutmayın: İnsan hafızası silinir, personeller değişir. Belgelenmemiş karar, hiç alınmamış karardır.

Dağıtık Ekiplerin Ağ Topolojisi: Uzaktan Çalışma



Eşzamansız (Asynchronous) Varsayılanı:

Zaman dilimlerine saygı gösterin. Yazılımcıların kesintisiz odaklanma (deep work) süresini toplantılarla bölmeyin.

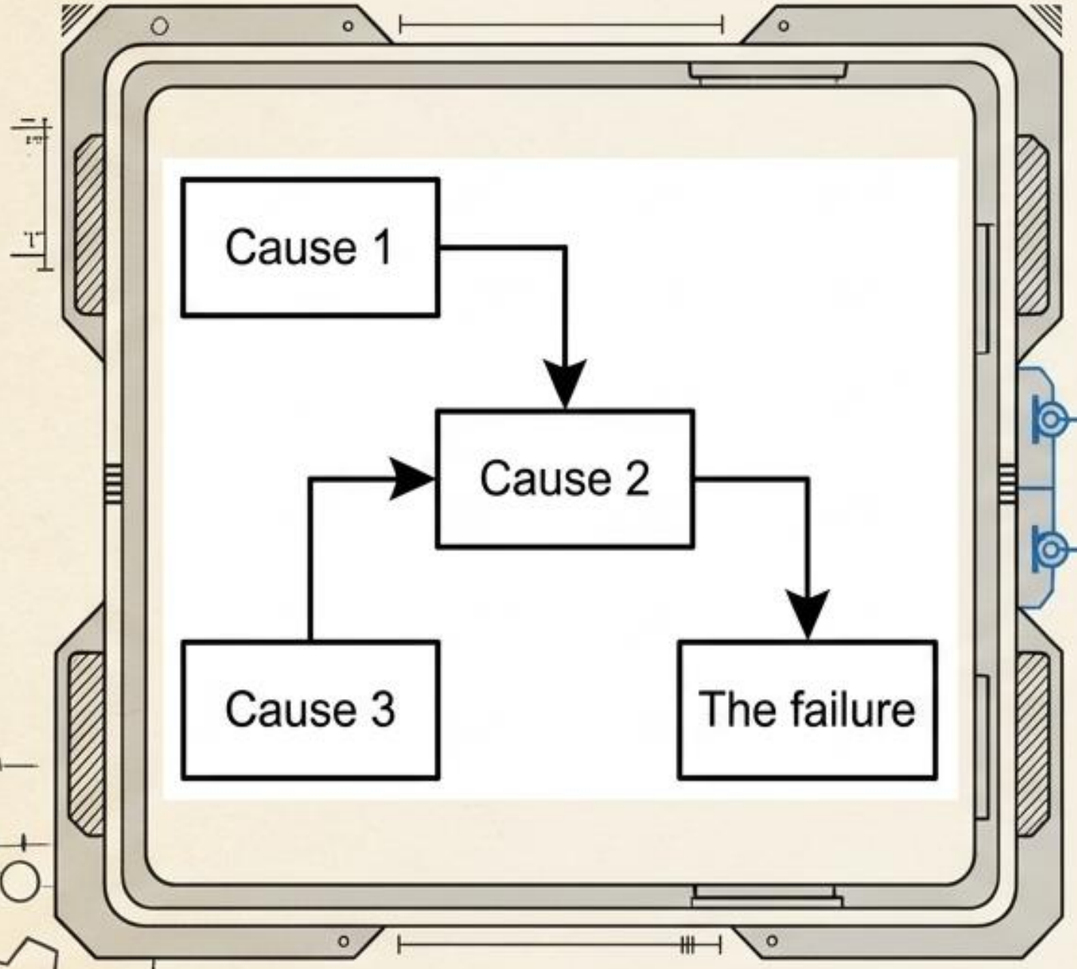
Devir Teslim (Handoff) Standartları:

İş bir sonraki zaman dilimine veya ekibe geçerken; mevcut durum, açık sorular ve sonraki adımlar eksiksiz belgelenmelidir.

Aciliyet Protokolleri:

- Anında **(Senkron):** Sunucu çökmesi, canlı sistem hatası.
- **Aynı Gün (Asenkron):** Bağımlılık engelleyiciler (blockers).
- **Ertesi Gün (Asenkron):** Rutin durum güncellemeleri, tasarım revizyonları.

Hata Ayıklama (Debugging): Başarısızlığın Kök Neden Analizi



Yazılım projelerindeki başarısızlıkların Kök Neden Analizi (RCA) şeması.

Başarısızlık (The failure) rastgele oluşmaz; **İnsan** (Sosyal etkileşim, motivasyon) ve **Görev** (Gereksinimler, yönetim) katmanlarındaki hataların reaksiyonudur.

Çıkarımlar:

- Teknik Olmayan Hatalar: Araştırmalar, başarısızlıkların doğrudan zayıf iletişim, belirsiz görev öncelikleri ve ekip içi uyumsuzluktan kaynaklandığını kanıtlamaktadır.
- İnsan değişkenini doğru yönetemeyen hiçbir metodoloji projeyi kurtaramaz.

Kaynaklar

- **BA Coach (2025)**: *Stakeholder Analysis: Mendelow's Matrix* – Mendelow Güç-İlgi Matrisi'nin 4 çeyreği (*Key Players, Keep Satisfied, Keep Informed, Minimal Effort*) ve paydaş haritalama stratejileri1more_horiz.
- **ProjectManagement.com (2026)**: *Stakeholder Analysis using the Power Interest Grid* – Farklı paydaş gruplarının (*Sponsor, Müşteri, Düzenleyici Kurumlar, Ekip*) güç ve ilgi düzeylerine göre yönetilmesi2.
- **Rework (2024)**: *Stakeholder Analysis: The Power-Interest Grid Explained* – Paydaş analizinin proje başarısızlıklarını önlemedeki rolü, PMI istatistikleri ve güç-ilgi ızgarası uygulamaları1more_horiz.
- **Productive.io (2026)**: *Communication Plan for Project Management* – İletişim planı bileşenleri, $N(N-1)/2$ iletişim kanalı karmaşıklık hesabı ve Karar Kayıtları (*Decision Logs / MoM*) süreçleri39.
- **Workhint Blog (2025)**: *Distributed Team Communication Plan For Managers* – Dağıtık ve uzaktan çalışan ekiplerde eşzamanlı (*synchronous*) ve eşzamansız (*asynchronous*) iletişim matrisleri ve el değiştirme (*handoff*) standartları3more_horiz.
- **The Digital Project Manager (2025)**: *Project Planning For Remote Teams* – Uzaktan çalışan yazılım ekiplerinde toplantı yönetimi, beklenti hizalaması ve şeffaf bilgi akışı stratejileri3.