

Yazılım Süreçlerinin Yedi Esas Özelliđi

Hakan Erdođmuş

Kalemun Research Inc.

UYMS • 8 Ekim, 2009

Yazılım Geliştirme Yaklaşımlarına İki Tarihsel Bakış

- **Mühendislik ve araştırmaya dayalı:** Yazılım “katı”dır; yazılım geliştirme bilimsel, planlama odaklı, kontrol edilebilir bir süreçtir.

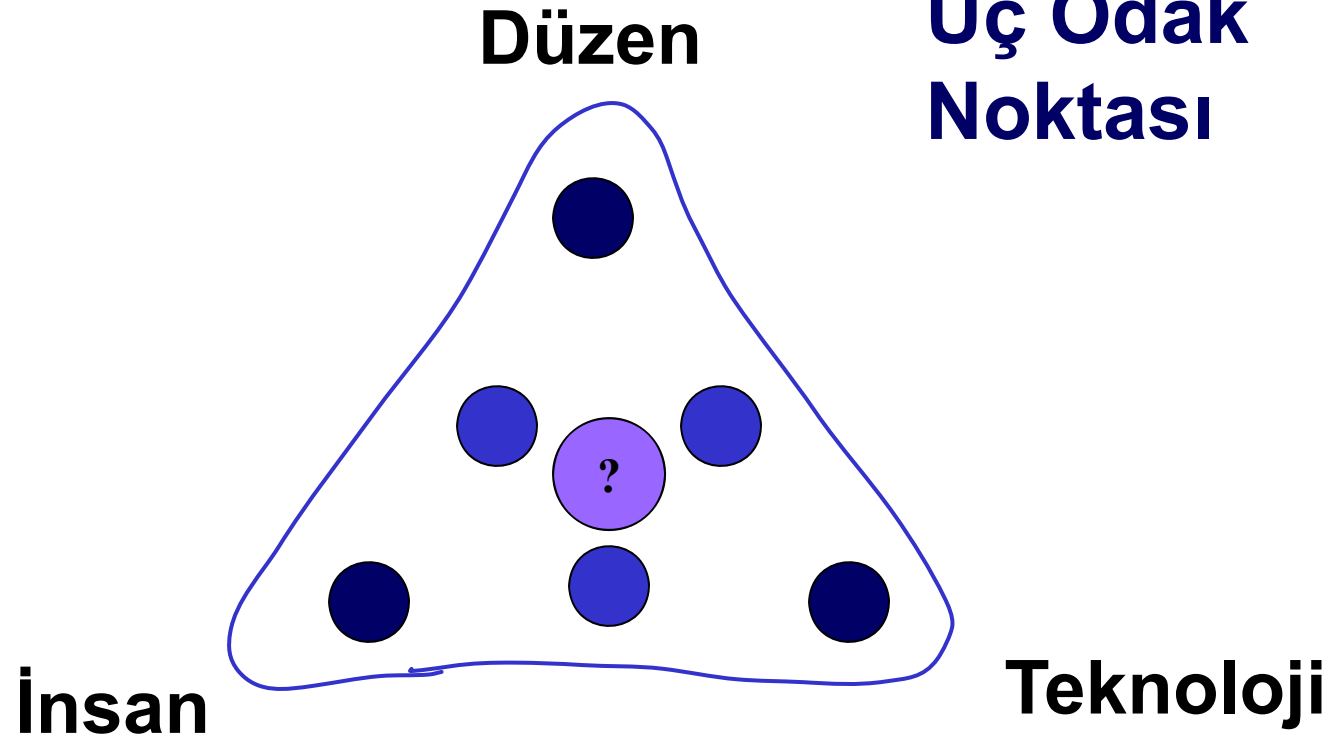
Yap, ilerlet, değiştir

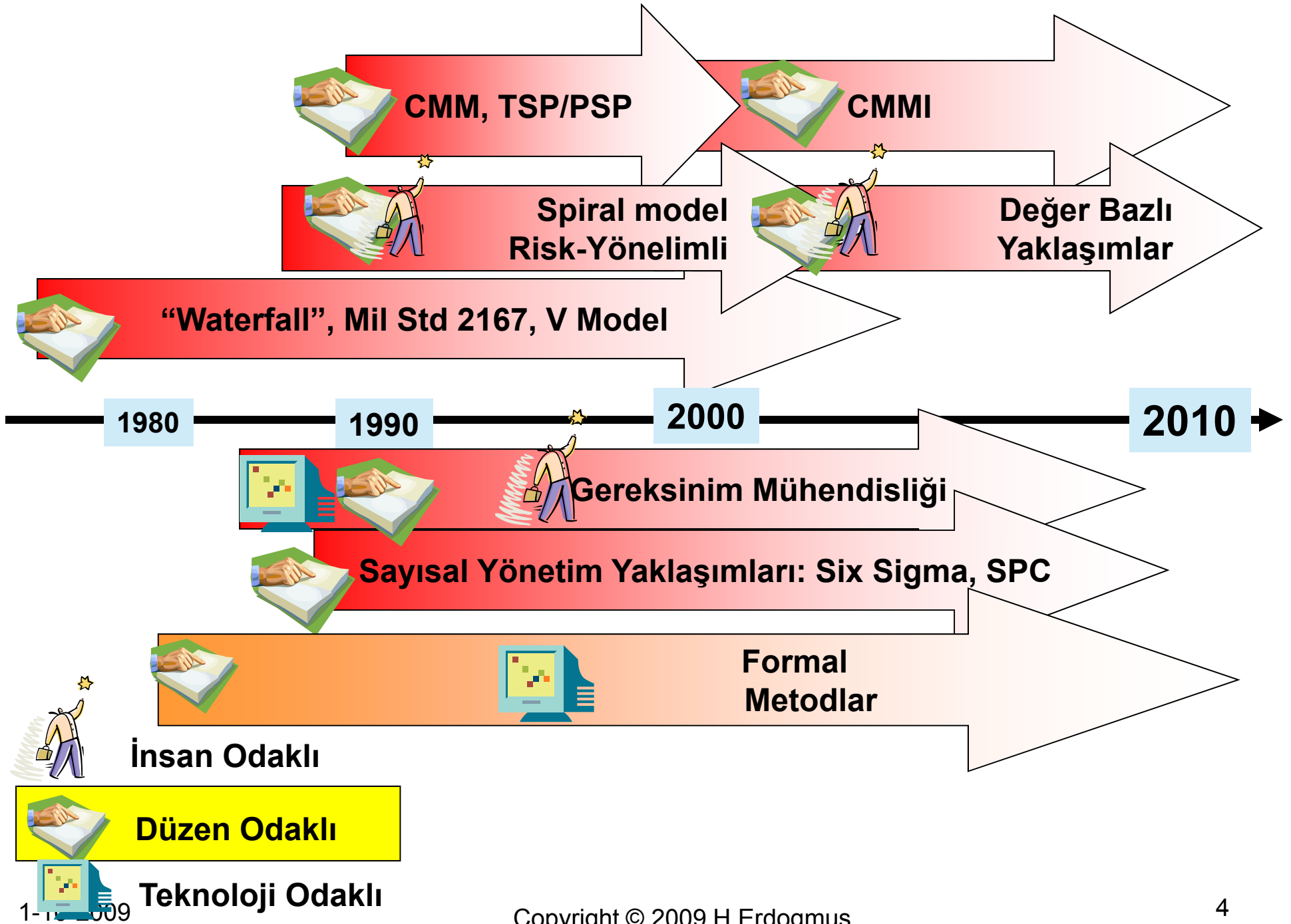


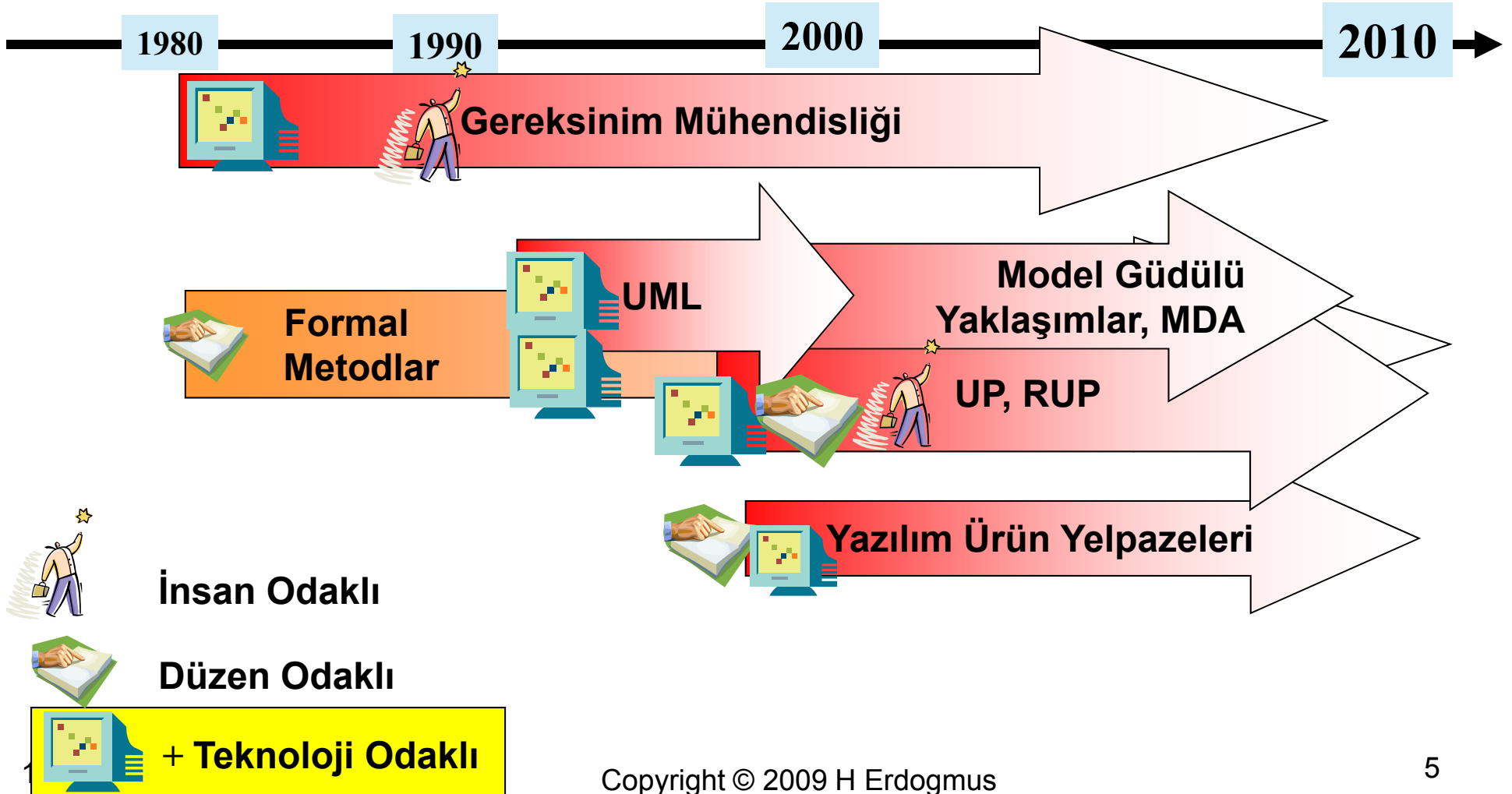
Planla, uygula, kontrol et

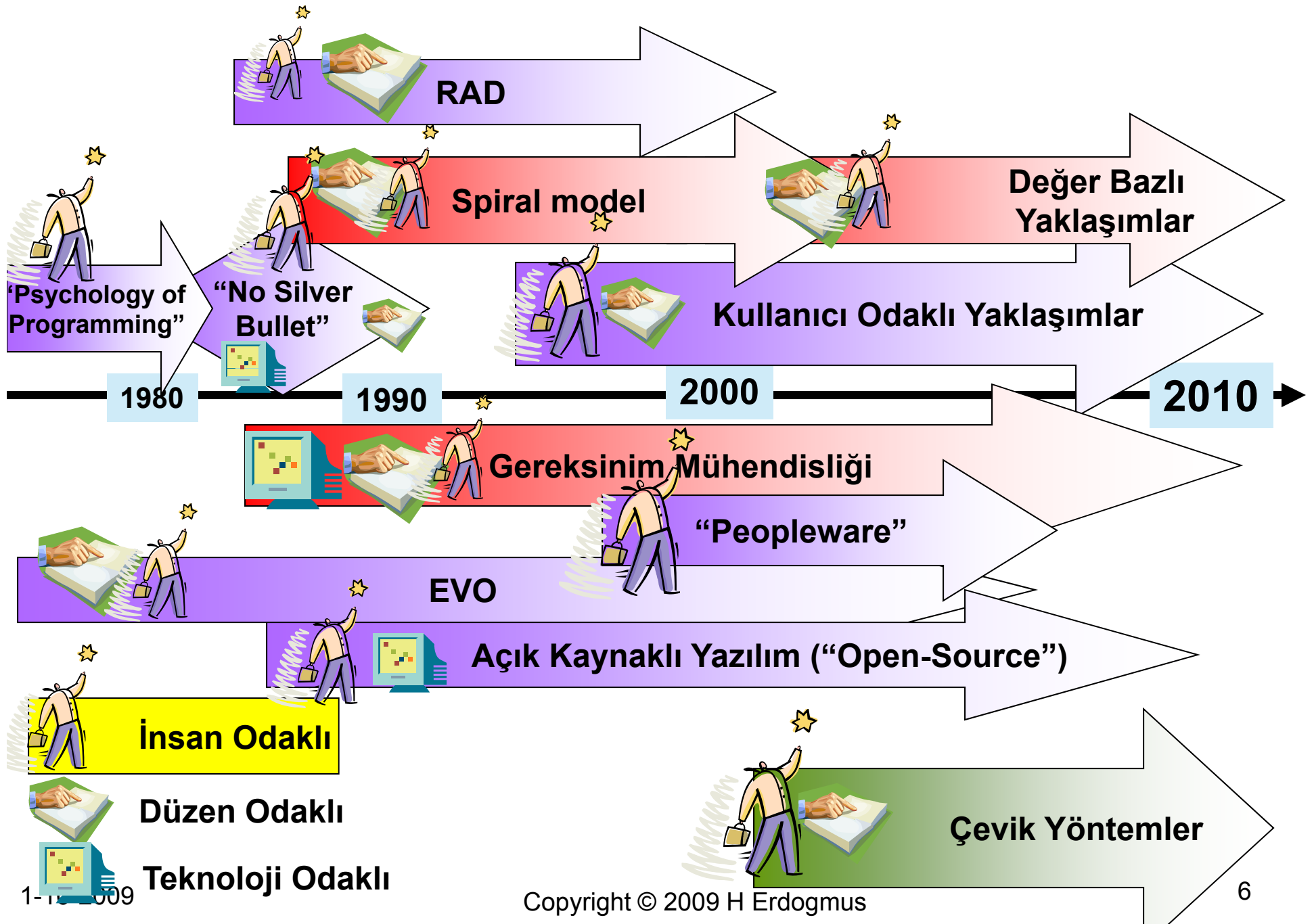
- **Tabandan destekli ve beceriye dayalı:** Yazılım “esnek”tir; yazılım geliştirme organik, yetenek gerektiren, teknoloji tarafından yönlendirilen bir süreçtir.

Yazılım Geliştirme Süreçlerinin Üç Odak Noktası







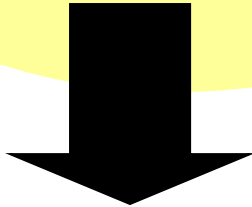


Bundan sonra: Melez Yöntemler Dönemi

Hiç biri bütün değil, hiç biri kesin değil!



Akımların yumuşamasıyla,
mevcut sürtüşmeler
yavaş yavaş
birbirini tamamlayan ve
destekleyen fikirlere dönüşüyor!

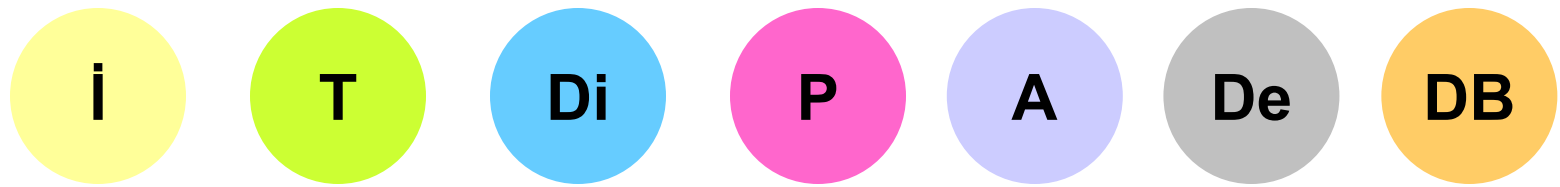


• ~~Süreç doktrininin~~ yerine

- Entegrasyon
 - Denge
 - Amaç
 - Ortam

Faydacı Süreç

- James Bach'ın yazılım süreçlerine *faydacı* bakışı (1995)...
- **Süreç:** bir grup problemi çözmek için kullanılan kalıplar ve etkinlikler sistemi
- Tom DeMarco (“Software Engineering: An Idea Whose Time Has Come and Gone?”, 2009):
~~Süreç – Yönetim + Ölçüm + Planlama + Kontrol~~



İnsan, insan, insan

[HireAttitudeSample.mp4](#)

***Kanada'daki yazılım müdürleri işe
alacakları mühendislerde ne tür
özellikler arıyorlar?***

(Janice Singer'ın izniyle)

- **“Özgeçmişlerde teknik beceriler her zaman aynıdır”**
- **“Onların davranış biçimlerine ve takım ile sağlayabileceklerini düşündüğüm uyuma daha çok önem veririm”**
- **“Daha az yetenekli, belki daha az akıllı, ama uyumlu bir çalışanı, daha çok yetenekli ama uyumsuz bir çalışana tercih ederim”**
- **“Sonuçta önemli olan kişiyle nasıl iletişim kurduğunuz ve o kişinin de diğerleriyle nasıl iletişim kurduğudur”**
- **“Onların iletişim, mümkünse yazılı iletişim, yeteneklerini ve takımla nasıl bir uyum içinde olacaklarını değerlendiririm”**
- **“Biz küçük bir ekibiz, dolayısıyla sosyal etkileşimimiz son derece önemlidir”**

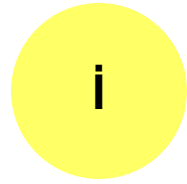
İnsan Merkezilik



Takım etkinliđi olarak yazılım geliřtirme:
İřlerinden gurur duyan motive olmuř, yaratıcı,
yetenekli bireyler

İnsan Merkezilik

- **Teşvikleme: etkili ve modern yazılım geliştirme ve yönetim tekniklerinin firmadaki teşviklerle uyuşması**
- **Uzlaştırma: farklı proje katılımcılarının ihtiyaçlarını dengelenmesi**
 - Aşağıdaki noktaların derinlemesine anlaşılması
 - Yazılım geliştirmecilerini ve yazılım müdürlerini neyin motive ettiği
 - Yazılımcı olmayan profesyonellerin yazılım geliştirmeyi nasıl gördükleri
 - Müşterilerin ne istediği ve kullanıcıların yazılımları nasıl kullandıkları
- **Sosyal beceriler: mesleki ve teknik becerilerle aynı bazda tutulması**
- **Etkin liderlik**
- **İşbirliği => Yoğun işbirliği**
 - En az sürtüşme ile çalışma
 - Birbirine güven, sosyal etkileşim
 - Müşteri ve kullanıcı ile yakın iletişim

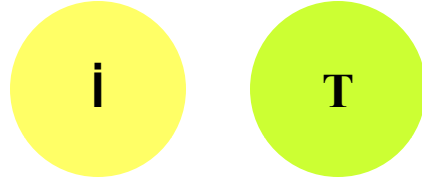


**Ancak *insan merkezilik* tek başına
başarıyı garanti edebilir mi?**

Teknik yönelim

Yazılım geliřtirmenin temellerinde ve yazılım teknolojisinde etkinlik

- Uygun teknolojinin kullanılması (programlama dilleri, programlama örneklemleri, tasarım yöntemleri, ...)
- Teknolojinin etkili bir şekilde uygulamasını destekleyen teknik çözümler ve teknik becerilere değer verilmesi
- Altyapıya önem vermek (araçlar, platform, mimari, geliştirme ortamları, uygulama çerçeveleri - “frameworks”)
- İşlenen en düşük soyutlama seviyesini (kaynak kod?) merkezi ve kesin eser olarak kabul etmek
- Otomasyondan mümkün olduğu kadar faydalanmak (örneğin derleme etkinlikleri)



Teknik yönlendirme tabii ki insan merkezliđi güçlendirir ancak her sürdürülebilir, ciddi, ölçeklendirilebilir iş hem teknik hem de yönetsel düzen ve sistemli bir yaklaşım gerektirmez mi?

Disiplin

**Ölçeklendirme ve tekrarlanabilirlik için özen gerekir....
ancak
disiplin \neq resmiyet veya bürokrasi**

- **Olgunluk seviyeleri (“maturity levels”)**
- **Körü körüne itaat**
- **Kontrol etme saplantısı**

değil

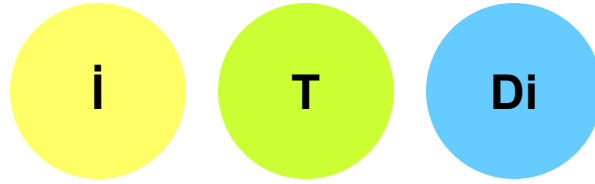
- **Sinerjistik etkinliklerin uyumlu ve sistemli bir şekilde uygulamaları**

Disiplin

- **İşin bölümlere ayrılması ve iş akışının yapılandırılması**
 - Bireysel
 - Ekip ve proje içinde
 - Firma içinde
- **Teknik etkinliklerin sistemli ve yeterli olarak uygulanması**
 - Program kodlama standartları
 - Gereksinim toplama teknikleri
 - Yazılım mimarisi, modelleme
 - İyi tasarım ilkelerine değer verip uygulamak
 - Program denetleme ve gözden geçirme, düzenli sınaama, regresyon testi
- **Yönetimsel etkinliklerin sistemli ve yeterli olarak uygulanması**
 - İnsan kaynaklarının, projelerin ve ürünlerin yönetimi
 - Planlama (yinelemeli ve artımlı – “iterative and incremental”)
 - Performans, verimlilik ve kalite takibi
 - Dokümantasyon

Disiplin

- **Tüm süreçler disiplini değişik şekillerde mecbur eder:**
 - CMMI ve RUP gibi süreç çerçeveleri
 - Extrem Programlama ve Scrum gibi çevik yöntemler



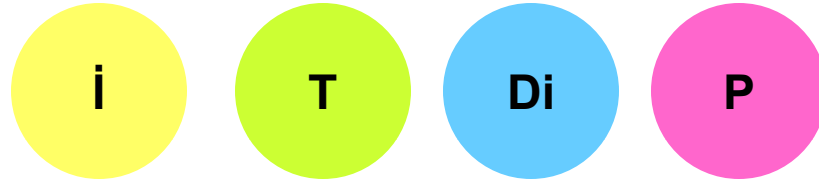
***Ancak disiplininin bir karřıt kuvvetle
dengelenmeye ihtiyaçı vardır...***

Pragmatiklik

Süreç etkinliklerinin sahip olması gereken özellikler

- Uygulanabilirlik
- Ortama uyum
- Ölçeklendirilebilirlik
- Firma kültürü ve firmanın risk alma tercihlerine uyumluluk
- Net bir amaca hizmet etme
 - Düzenleyici gereksinimleri tatmin etmek
 - Müşteri/kullanıcı ihtiyaçlarını karşılama
 - Geliştirme ekibinin ihtiyaçlarını karşılama
 - Diğer proje katılımcılarının ihtiyaçlarını uzlaştırma

Dokümantasyon == **P** ==> **sadece yeteri kadar ve faydalı dokümantasyon**



***Süreç seçimlerinde çok pragmatik olabilirsiniz,
ama seçimlerinizin beklentilerinizi karşıladığını
nasıl bileceksiniz?***

Öngörülen riskleri nasıl yönetebilirsiniz?

Ampirizm

- **Kararları veriye dayalı olarak destekleme: hem gözlemlerle hem de ölçümlerle**
- **Girdi ve çıktıları ampirik bilgi ile ilişkilendirme**
 - Etkinlik ve eylemlerin sonuçlarından haberdar olmak
 - Ne yaptığınızı bilmek:
 - Başarısız veya başarılı olduğunuzda
 - Gelişme gösterirken veya geriye giderken
 - Tekrar kullanılabilir ve paylaşılabilir bilgiyi yakalamak
 - İleriye dönük karar vermek için kanıtları toplamak ve gözden geçirmek
 - Kendi ortamınızdan
 - Dış kaynaklardan

“Verinin var olduğu ortamlarda yönetim olmadığı ortamlardan çok daha kolaydır.”

– Robert L. Glass

Ampirizm

Ampirizmin desteklediđi etkinlikler:

- Önkestirim
- Sözleşme görüşmeleri ve fiyatlandırma
- Proje planlaması ve yönetimi (kaynak sağlama, öncelik sırasını belirleme, gelişme takibi, piyasaya çıkış/yayımlama “release” planlaması)
- Kalite güvencesi
- Risk yönetimi
- *Genelde yüksek nitelikli karar alışı*

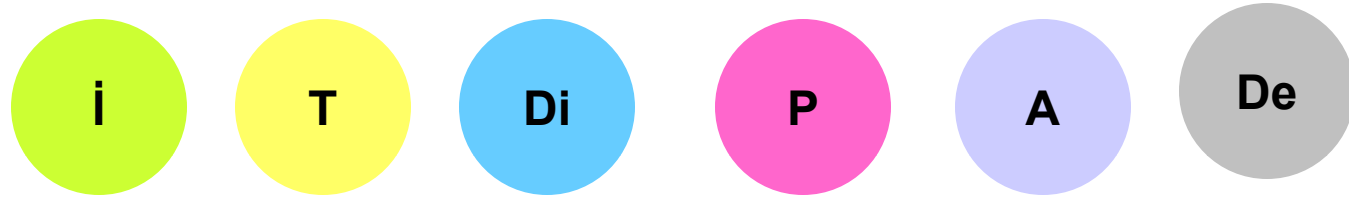


Ampirizm aynı zamanda neyin işe yarayıp yaramadığı hakkındaki teorilerimizi içgüdüsellüğün ötesinde daha tarafsız bir şekilde sınamanıza yardımcı olur. Böylece öğrenmeyi teşvik eder...

Deneyleme

Alternatif çözüm ve varsayımların güvenilir ve az maliyetle sınanabileceği bir ortamın sağlanması

- ***“Spikes” (teknik bir çözümün bir örnekle olurluk denemesi)***
- ***Prototipleme***
- ***Simülasyon (Benzetim)***
- ***Çoklu yayımlamak (“Split runs”)***
- ***Erken kullanılabilirlik değerlendirmesi***



Sistemik deneyleme aslında deęer arttırıcı bir etkinliktir...

Değer Nedir?

- **Bir amaca hangi oranda yaklaşıldığı**
- **Kârlılık**

Değer Bazlılık

Yazılım projeleri kaynak ve zaman açısından kısıtlı şartlar altında yürütülür

- **Süreç etkinliklerinin karlılık ve proje başarısı üzerinde etkilerini anlamak ve açıkça ifade etmek:**
 - Karlılığı/sonuçları nasıl destekliyorlar ve arttırıyorlar
 - Karlılığı/sonuçları arttırmak için birbirlerini nasıl destekliyorlar?
- **Başarıyı değer bazlılık anlamında ölçmek**
- **Değer bazlılığı ve teşvikleri aynı konuma getirmek**

Değer Bazlılık

Maliyet etkinliği ile değer yaratma...

+ **Geliştirme verimliliğini arttır**

– **Üretim maliyetini indir**

– **Düşük kalite maliyetini azalt (ya da kaliteyi arttır)**

⇒ **Net verimliliği arttır**

*“When the Pursuit of Quality
Destroys Value” – John Favaro*

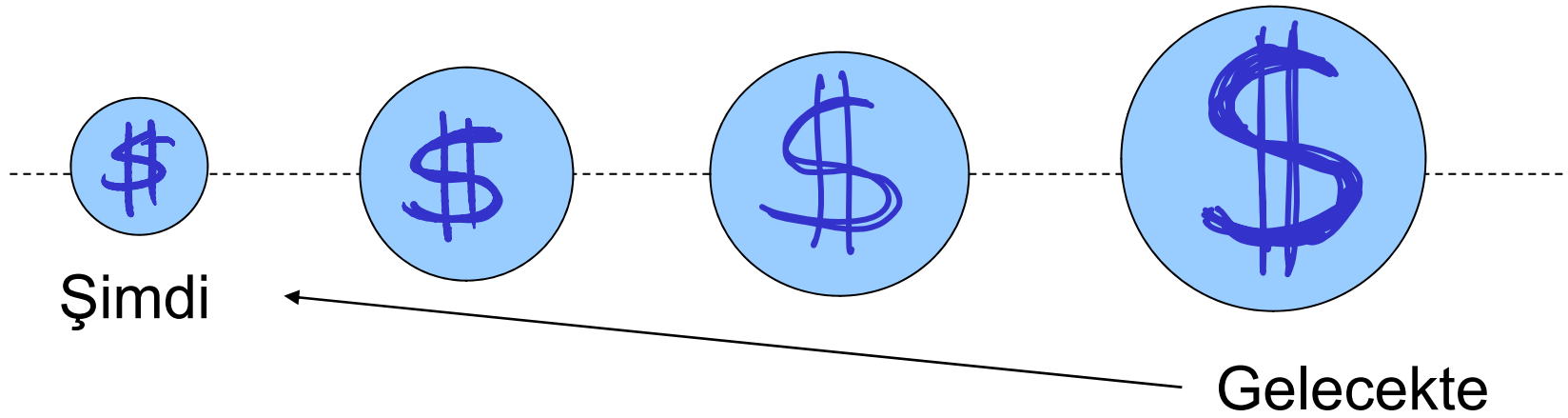
Üretim Çıktısı

Toplam Maliyet = Düşük Kalite Maliyeti + Üretim Maliyeti

Değişik aşama ve aktiviteler arasında kaynak tahsisinin optimizasyonu

- Örneğin yeni işlev geliştirme ve sınaama etkinlikleri arasında

Zamanın Deęeri

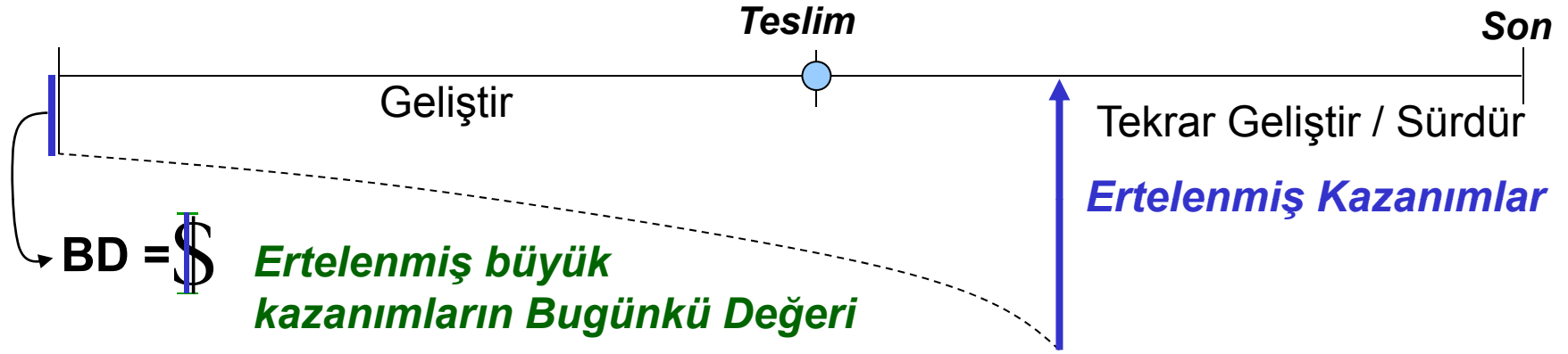


Erken kazan!

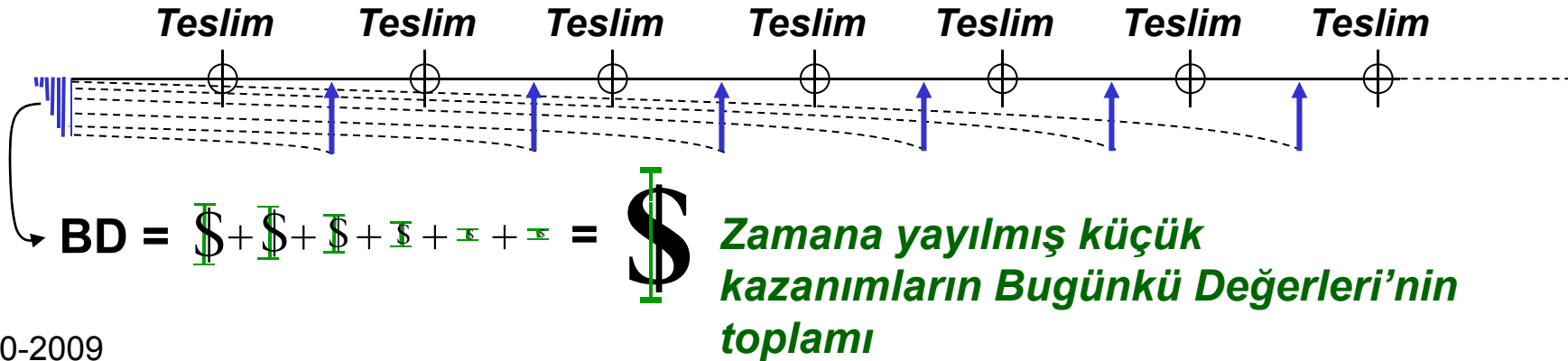
Geç harca!

Zamanın Değeri Artımlı Teslimatı Tercih Eder

Seyrek: Ertelenmiş kazanımlar, hepsi bir anda veya büyük kısımlar halinde



Artımlı: Erken kazanımlar, küçük kısımlar halinde



Değer Bazlılık

Değer yaratma stratejileri

- **Kazanımların bugünkü değerini arttır**
 - **i** Son kullanıcı veya müşterilerin neyi önemsediklerini anlayarak
 - Erken ve sık teslimatı garanti ederek
 - Etkin önceliklendirme: yüksek değerli işlemlere öncelik tanıyarak
- **Özellikle maliyet ve kazançların belirsiz olduğu durumlarda yatırımları mümkün olduğunca ertele**
- **Di A** Yeterli derecede değerlendirme ve planlamayla öngörülebilir riskleri izle ve azalt

Değer Bazlılık

Diğer Değer Yaratma Stratejileri

- **De Belirsizliğin yüksek olduğu durumlarda esnekliği ve öğrenme kapasitesini arttır**
 - **Öğrenme kapasitesini arttırmak için geri besleme devirlerini kısalt**
 - *Esneklik yalnızca belirsizlik durumunda değerlidir!*



Zamanında Karar Verme

Yeni bir işlev ilave etmeyi düşünüyorsunuz...

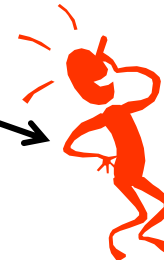
Bugünkü maliyeti
10000 \$

Kazancın 15000 \$
olacağını tahmin
ediyorsunuz... Bir sene
içinde belli olacak!

Belirsizlik

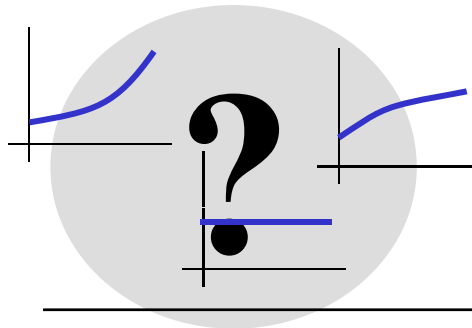


İyi Senaryo: Müşteri
memnundur; kazanç
elde edilmiştir



Kötü senaryo: Müşteri
açısından bir şey
değişmemiştir; zaman
ve para kaybettiniz

Uygulama Maliyeti



Zaman

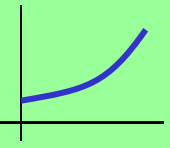
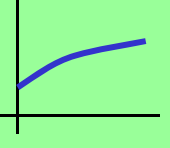
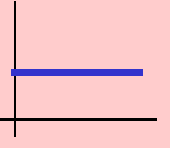
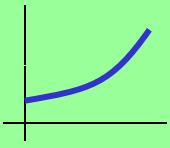
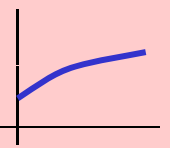
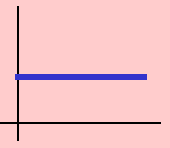
... opsiyon bazlı değer analizi
yoluyla

... uygulama maliyeti ve
belirsizliğe dayalı

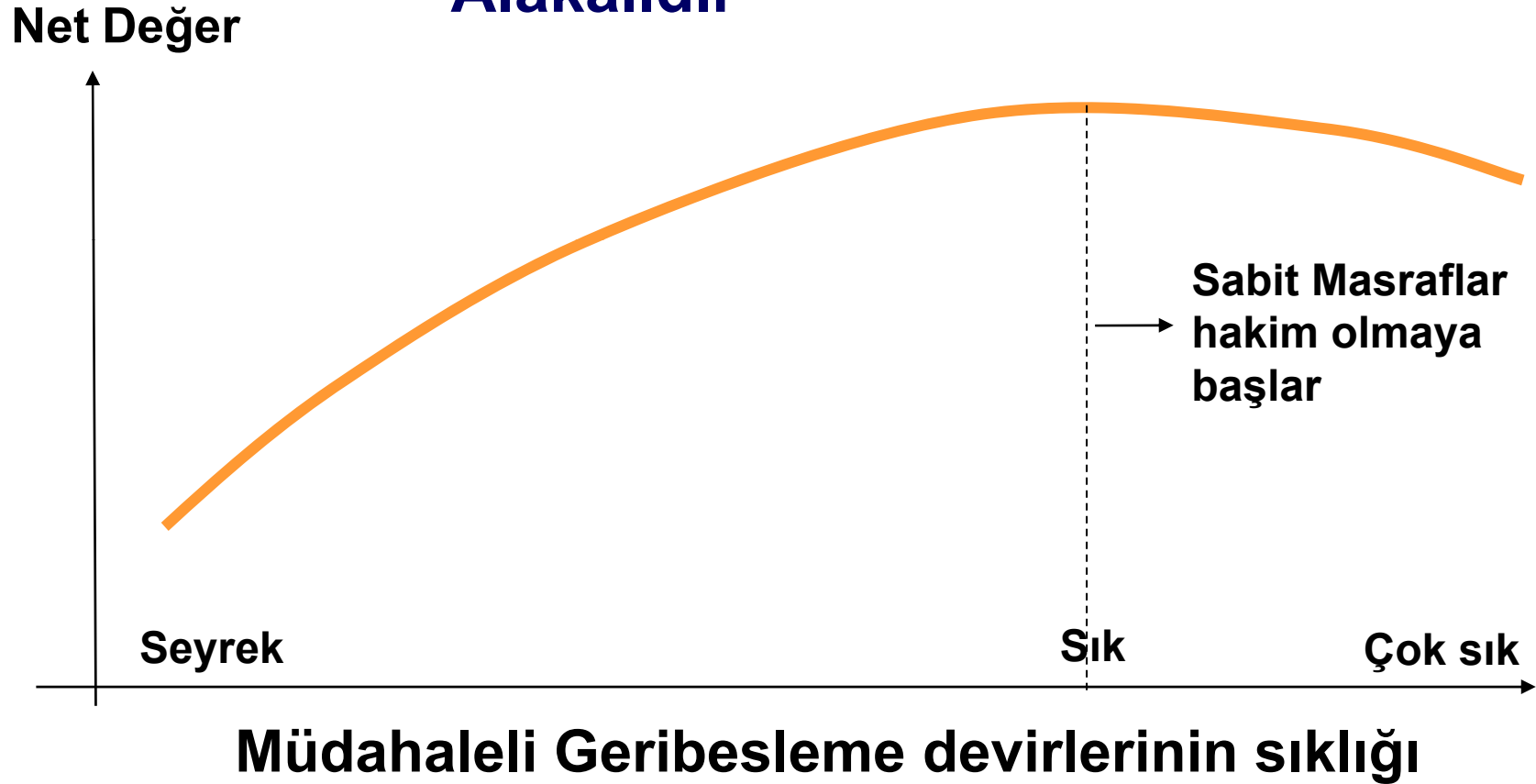
Zamanında Karar Vermenin Genel Kuralları

Uygulama Maliyeti

Belirsizlik / Zaman Süreci

	Geleneksel	Yavaş Artan	Sabit
Alçak / Kısa	Erken 	Erken 	Geç 
Yüksek / Uzun	Erken 	Geç 	Geç 

Belirsizlik Durumunda Öğrenme Kapasitesinin Önemi Müdahaleli Geribesleme Devirlerinin Etkisi ile Alakalıdır



Geribesleme devirlerini (yinelemelerin süresini) kısaltın!

DB?

Geleneksel Başarı Ölçütleri Her Zaman Geçerli midir?

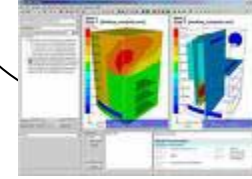
Bütçeyi Aşmadan



Zamanında



Tam İşlevsel



- İstatistikî proje başarı oranları bu kriterlere göre düşük: $< 60\% \Rightarrow$ yani bir yazılım krizi mi söz konusu?
- Burada hedef: *tekrarlanabilirlik ve öngörülebilirlik*; acaba bu doğru mu?

Şöyle bir proje başarılı mıdır?

Bütçeyi %100 aşmış, %100'ü gecikmiş,

Sadece öngörülen işlevlerin sadece %10'u planlandığı biçimde

teslim edilmiş ve %20'si önemli bir şekilde değiştirilmiş,

ancak

ürün 30000 çalışan tarafından verimliliği %20 arttırarak

her gün kullanılıyor, hem de asgari destekle

***Bu yedi temel unsur
hem birbirlerini destekleyici
hem birbirlerini dengeleyici;
her biri deęişik geliştirme süreçleri için
deęişik ölçülerde geçerli...***

**Teknik
Yönelim**

**İnsan
Merkezcilik**

**Deęer
Bazlılık**

Disiplin

Pragmatiklik

Ampirizm

Deneyleme

