

6. Hafta

YMÜ225

Yazılım Gereksinim Analizi

Dr. Feyza Altunbey Özbay

İçerik

- Tahmin Teknikleri
- Algoritmik Maliyet Modelleme
- COCOMO Maliyet Modeli
- COCOMO81
- COCOMO II

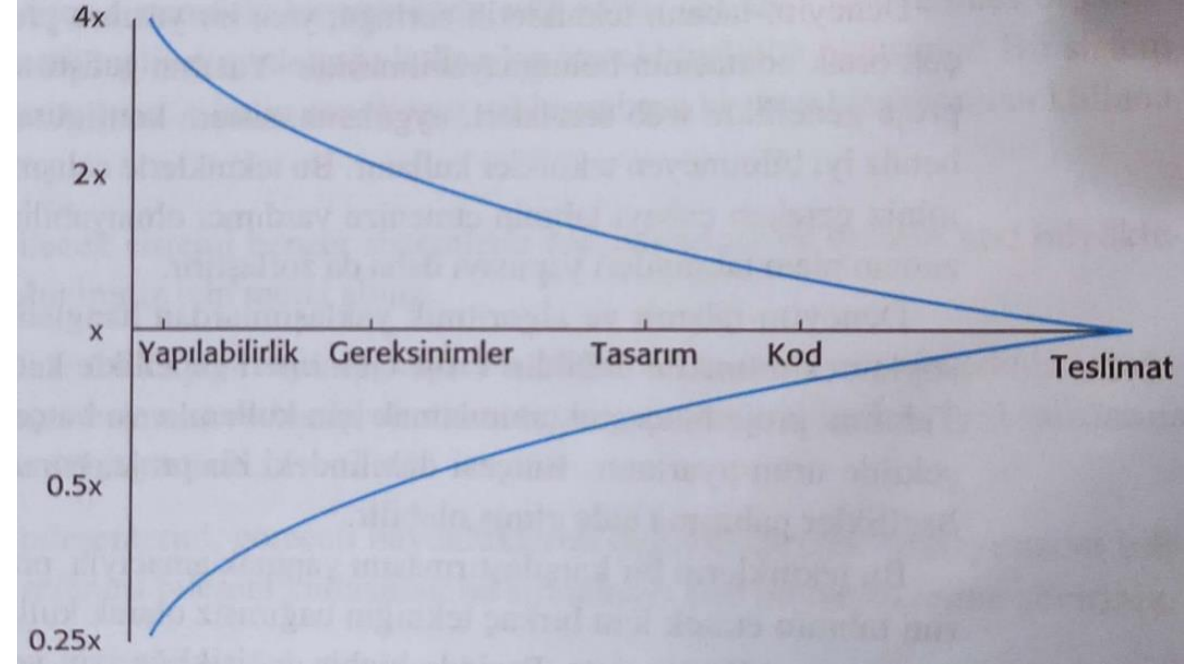
Tahmin Teknikleri

Proje zaman planlarını tahmin edilmesi zordur.

Tamamlanmamış kullanıcı gereksinimleri temel alınarak tahminler yapılmaya çalışılır.

Bir takım belirsizlikler:

- Yazılımın iyi bilinmeyen platformlarda çalışması ya da yeni geliştirme teknolojisi kullanılması gerekebilir.
- Projede yer alacak kişiler ve becerileri bilinmeyebilir.



Tahmin belirsizliği

Bütün belirsizliklere rağmen kurumlar yazılım çaba ve maliyet tahmini yapmaya gerek duyar. Tahmin için iki tür teknik vardır:

- Deneyim-tabanlı teknikler
- Algoritmik maliyet modelleme

- Deneyim-tabanlı teknikler: Gelecekteki çaba gereksinimlerinin tahmini, yöneticinin geçmiş projelerdeki deneyimi ve uygulama alanındaki deneyimine dayanır.
- Algoritmik maliyet modelleme: Bu yaklaşımda büyüklük, süreç özellikleri ve ilgili diğer personelin deneyimi gibi ürün niteliklerinin tahminlerine dayanarak proje çabasının hesaplanması için formüle dayalı bir yaklaşım kullanır.

Algoritmik Maliyet Modelleme

- Algoritmik maliyet modelleme;
- Proje büyüklüğünün tahminine
- Geliştirilen yazılımın türüne
- Diğer takım, süreç ve ürün faktörlerine dayanarak proje maliyeti tahmin etmek için matematiksel formül kullanır.

$$\text{Çaba} = A \times \text{Büyükölük}^B \times M$$

A: yerel kurumsal uygulamalarda ve geliştirilen yazılım türüne bağı olan sabir bir faktör

Büyükölük: fonksiyon ve uygulama noktası olarak ifade edilen, yazılım veya bir fonksiyonellik tahmininin kod büyüklüğünün bir değeriendirmesi

B: Yazılım karmaşıklığını temsil eder ve 1 ile 1,5 arasında değeri alır.

M: Yazılım için güvenilirlik gereksinimleri ve geliştirme takımının deneyimi gibi süreç, ürün ve geliştirme özelliklerini dikkate alan bir faktördür.

Algoritmik maliyet modelleri, tamamlanmış projelerin maliyetlerini ve özelliklerini analiz ederek ve daha sonra oluşan fiili maliyetlere en iyi uyan formülü bularak geliştirilmiştir.

Algoritmik maliyet modeli

Teslim edilen sistemdeki kaynak kodu satır sayısı (SLOC: Source lines of code) çoğu algoritmik maliyet modelinde kullanılan temel büyüklük ölçüsüdür. Kod satır sayısını tahmin etmek için aşağıdaki yaklaşımların bir kombinasyonu kullanılabilir:

- Geliştirilecek sistem benzer sistemlerin kod büyüklükleri ile karşılaştırılır ve tahmin için kullanılır.
- Sistemdeki fonksiyon veya uygulama noktalarının sayısını tahmin edilir.
- Sistem bileşenleri, göreceli büyüklükler değerlendirilerek sıralanır. Bilinen bir referans bileşeni kullanarak bu sıralamayı

Algoritmik maliyet modeli

Tüm algoritmik maliyet modelleri iki önemli problemden olumsuz etkilenir:

- Sadece tanımlamaların hazır olduğu projenin erken aşamalarında büyüklüğü doğru tahmin etmek imkansızdır.
- B ve M'ye katkıda bulunan karmaşıklık ve süreç faktörleri tahminleri öznelidir.

COCOMO

- 1981 yılında, Dr. Barry Boehm tarafından tanıtılmıştır.
- İsminin kısaltması ve yılı nedeniyle bu modele "COCOMO 81" de denilmektedir.
- Yazılım projesinde olması gereken çabanın, program büyüklüğünün bir üstel değerine bağlı olması prensibi ve endüstriden toplanan bilgiler ışığı altında geliştirilmiş bir kestirim metodu olan COCOMO (*Constructive Costing Model*) deneysel, gözlemsel bir modeldir.

$$\text{Çaba} = A \times \text{Büyükölük}^B \times M$$

COCOMO 81, birçok yazılım projesinden veri toplanarak, ardından gözlemlerle uyuyacak en iyi formölü keşfetmek için bu verilerin analiz edilerek çıkarılmış bir modeldir.

COCOMO

COCOMO'nun seçilme nedenleri;

- COCOMO'da kullanılan projeler çeşitlilik göstermektedir. Farklı alanlarda yapılmış projelerdir.
- COCOMO oldukça yaygın olarak kullanılmaktadır ve değerlendirilmektedir.
- COCOMO'nun ele aldığı projeler iyi ele alınmış projelerdir. İyi seviyede dokümante edilmiştir.
- Model ayrıntılı olarak açıktır. COCOMO modeli hem ticari hem de ticari olmayan kurumlarca desteklenmektedir.
- 1981'de ilk defa ortaya çıkmasından (Boehm B.W., 1981), Ada yazılım geliştirme için iyileştirilmesine, en yeni versiyonunun 1995'de yayınlanmasına kadar uzun bir soy ağacı mevcuttur. Böylece sürekli iyileştirme içinde olmuştur. Bu zamanda yaşanan gelişmelere paralel olarak sürekli güncellenmiştir.

COCOMO 81

- COCOMO modelinin ilk versiyonu (COCOMO 81), maliyet tahmini analizinin ayrıntılarını yansıtan üç seviyeli bir modeldir.
 - Birinci seviye : Başlangıç ve kaba bir tahmin sağlanmakta
 - İkinci seviye : Proje ve süreç çarpanlarını kullanarak bunu değiştirir
 - Ayrıntılı seviye : Projenin farklı aşamaları için tahminler üretir
- Yapılacak hesapların kapsamları açısından üç değişik modelden oluşur. Basit model, orta ve detaylı modeller.
- Ayrıca bu modellerde kullanılacak problemler, 'organik, yarık ayırık ve gömülü sınıflar altında toplanmıştır:

COCOMO 81

- **Organik (Ayrık) problemler** için küçük boyuttaki programcı takımları, bildikleri ortamlarda iyi anlaşılmış uygulamaları geliştirirler. İletişim problemleri azdır ve elemanlar hızlıca işlerini halledebilecek durumdadırlar.
- **Yarı ayrık problemlerde** ise ekipte tecrübeli ve tecrübesiz elemanlar bulunabilir. İlgili sistemler konusunda deneyimleri sınırlı olabilir ve geliştirilen sistemin her kısmını bilmeyebilirler.
- **Gömülü problemler**, Geliştirilecek yazılım, sistemin donanım, kurallar, işletim süreçleri veya yazılım gibi diğer bileşenleri ile çok kuvvetli bağlantılar oluşturur. Gereksinim değişiklikleri ile problemleri halletmek oldukça zordur. İhtiyaç belirtiminin geçerlilik incelenmesi çok maliyetlidir. Yazılımın geliştirilmesinde görev alanların tüm kısımlara hakim olma olasılığı iyice azalmıştır.

COCOMO 81

- Problem sınıflarına göre basit model için aşağıdaki formüller kullanılır.

Problem	Çaba	Süre
Organik	$\text{Çaba} = 2.4 (\text{KLOC})^{1.05}$	$\text{Süre} = 2.5 (\text{Çaba})^{0.38}$
Yarı ayrık	$\text{Çaba} = 3 (\text{KLOC})^{1.12}$	$\text{Süre} = 2.5 (\text{Çaba})^{0.35}$
Gömülü	$\text{Çaba} = 3.6 (\text{KLOC})^{1.20}$	$\text{Süre} = 2.5 (\text{Çaba})^{0.32}$

- Basit COCOMO modeli, küçük, orta boy projeler için hızlı kestirim yapmak amacıyla kullanılır.

KLOC: kilo lines of code (bin satırlık kod)

COCOMO 81

- Orta detaydaki modelde ise, sistemin güvenilirlik, veri tabanı büyüklüğü, işletme ve kayıt sınırlandırmaları, personel özellikleri ve kullanılan yazılım araçları gibi diğer özelliklerinin hesaba katılması amaçlanmıştır. Orta seviyedeki projeler için COCOMO ölçütleri:

Ürün	Donanım	İnsan	Proje
Güvenilirlik	Hız	Analist yeteneği	Yazılım aracı kullanımı
Veri tabanı büyüklüğü	Bellek yeterliliği	Uygulama Deneyimi	Zamanlandırma
Ürün Karmaşıklığı	Sanal Makine Değişkenliği	Geliştirme Ortamı Deneyimi	Yeni Programlama Teknikleri
	Kullanılabilme Süresi	Yazılım Geliştirici Yeteneği	
		Programlama Dili Deneyimi	

COCOMO 81

- Belirli bir takım özelliğın, proje açısından etkileri ayrı ayrı ağırlıklandırılarak katsayılar ortaya çıkarılır. Çizelgede yazılım geliştirme çaba çarpanları gösterilmiştir.
- Bu faktörler, ilgili özellik için düşük (<1), nominal (1) veya yüksek (>1) olarak saptanırlar.
- Katsayılar birbiri ile çarpıldığında “Çaba Ayarlama Katsayısı” (ÇAK) (*Effort Adjustment Factor*) bulunur.
- Bu katsayı 0.9 ile 1.4 arasında bir değer alır

Ölçüt	Açıklama	Değerlendirme					
		Çok Az	Az	Normal	Yüksek	Çok Yüksek	Aşırı Yüksek
Product							
RELY	Yazılımın Güvenilirliği	0.75	0.88	1.00	1.15	1.40	-
DATA	Veritabanı Hacmi	-	0.94	1.00	1.08	1.16	-
CPLX	Ürün Karmaşıklığı	0.70	0.85	1.00	1.15	1.30	1.65
Bilgisayar							
TIME	Çalışma zamanı kısıtı	-	-	1.00	1.11	1.30	1.66
STOR	Ana depolama kısıtı	-	-	1.00	1.06	1.21	1.56
VIRT	Sanal makine uçuculuğu	-	0.87	1.00	1.15	1.30	-
TURN	Bilgisayar değişim (dönme) zamanı	-	0.87	1.00	1.07	1.15	-
Personel							
ACAP	Analistin yetenekleri	1.46	1.19	1.00	0.86	0.71	-
AEXP	Uygulama Tecrübesi	1.29	1.13	1.00	0.91	0.82	-
PCAP	Programcı kapasitesi	1.42	1.17	1.00	0.86	0.70	-
VEXP	Sanal Makine Tecrübesi	1.21	1.10	1.00	0.90	-	-
LEXP	Dil Tecrübesi	1.14	1.07	1.00	0.95	-	-
Project							
MODP	Modern programlama deneyimi	1.24	1.10	1.00	0.91	0.82	-
TOOL	Yazılım Araçları	1.24	1.10	1.00	0.91	0.83	-
SCED	Geliştirme Zaman Planı	1.23	1.08	1.00	1.04	1.10	-

COCOMO 81

Orta seviyedeki projeler için COCOMO ölçütleri:

Problem	Çaba
Organik	$\text{Çaba} = 3.2 (\text{KLOC})^{1.05} \times \text{ÇAK}$
Yarı ayrık	$\text{Çaba} = 3.0 (\text{KLOC})^{1.12} \times \text{ÇAK}$
Gömülü	$\text{Çaba} = 2.8 (\text{KLOC})^{1.20} \times \text{ÇAK}$

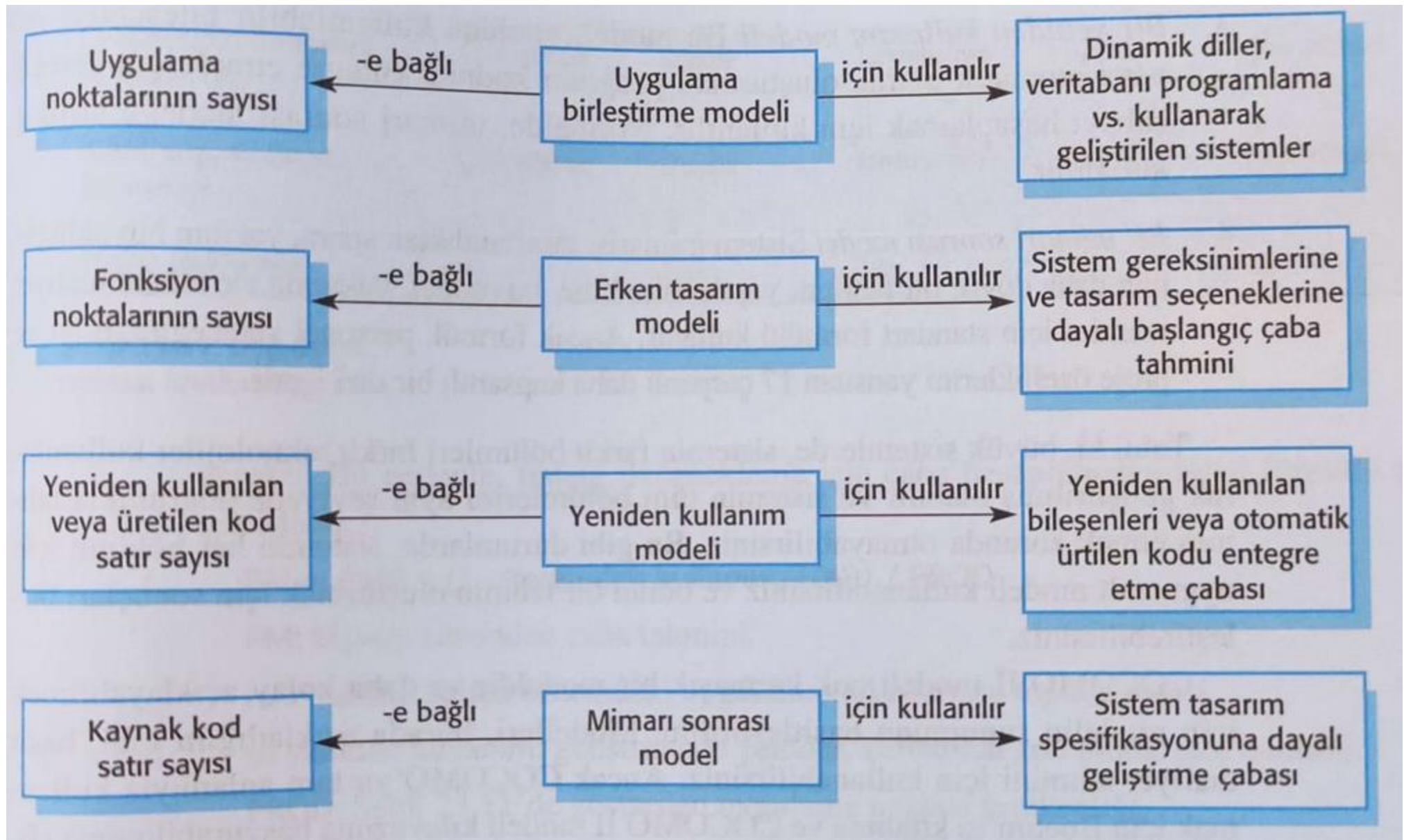
- Elde edilen ÇAK değeri tablodaki formülleri kullanılarak çaba hesaplaması ile sonuçlandırılır.
- Süre hesaplaması ise Basit COCOMO modelinde olduğu gibi yapılır.

COCOMO 81

- Detaylı COCOMO modeli ise projenin evrelerine bağlı olarak süreç içinde değişiklikleri hesaba katarak arada bir kestirim hesaplamasını önerir (Fenton, N.E. ve Pfleeger S., 1998).
- Bu modelde zamana bağlılık temel değişikliktir.
- Projenin farklı evrelerinde çaba yoğunluğu ve yapılacak işin karmaşıklığı değişecektir.
- Proje başlangıcındaki az iş gücü ile gereksinimlerin ilk çabaları düşük düzeylerle temsil edilirken geliştirmenin analiz, tasarım ve uygulama gibi evrelerinde yükseklerdedir. Daha sonra geliştirme sonrası faaliyet azalır. İleride bakım ve onarım yine bazı geçici yükselmeler yapabilir.

COCOMO II

- COCOMO 81, yazılımın şelale sürecine göre geliştirildiği düşünülerek geliştirilmiş bir modeldir. Bu başlangıç versiyonunun önerilmesinden sonra yazılım geliştirmede büyük değişiklikler meydana geldi.
- Yazılım, yeniden kullanılabilir bileşenlerin birleştirilmesi ve bunların bir betik bir dil aracılığıyla geliştirilmesi yoluyla yaratılabilir.
- Prototip oluşturarak ve artırımsal geliştirme çoğunlukla kullanılan bir süreç modelleridir. Birçok durumda var olan alt sistemler, kütüphaneler kullanılır. Yeni yazılım oluşturmak için var olan yazılım üzerine yeniden mühendislik yapılabilir.
- Bu değişiklikleri hesaba katmak için COCOMO II prototip oluşturarak, bileşenleri birleştirerek geliştirme, dördüncü nesil programlama dillerini kullanma gibi farklı yaklaşımlar kabul etmiştir.



COCOMO Maliyet Modelleri

COCOMO II

- COCOMO II'de tanımlanan seviyeler:
- **Erken prototip oluşturma seviyesi:** Büyüklük tahminleri nesne noktalarını temel alır ve gerekli çabayı tahminlemek için basit büyüklük/üretkenlik formülü kullanılır.
- **Erken tasarım seviyesi:** Bu seviye, sistem ihtiyaçlarını bazı başlangıç tasarımıyla tamamlamaya karşılık gelir.
- **Mimari sonrası seviyesi:** Sistem mimarisi tamamlandığında, yazılım büyüklüğü hakkında mantıklı, doğru tahmin yapılabilir. Bu seviyedeki tahmin, personelin yeteneklerini, ürün ve proje özelliklerini yansıtan daha kapsamlı çarpan kümesi kullanır.

Erken prototip oluřturma seviyesi

- Erken prototip oluřturma seviyesi, prototip oluřturularak gerekleřtirilen projeler ve var olan bileřenlerin birleřtirilmesiyle oluřturulan projeler iin gerekli abayı tahminlemek amacıyla COCOMO'nun iinde mevcuttur.
- Programcı retkenlięi; geliřtiricinin deneyimine ve yeteneęine ve geliřtirmeyi desteklemek iin kullanılan bilgisayar destekli yazılım geliřtirme aralarının yetenekleri ile iliřkilidir.

Yazılım Geliřtiricinin deneyimi ve yeteneęi	Bilgisayar Destekli yazılım mhendislięi aracının olgunluęu ve yeteneęi	retkenlik
ok Az	ok Az	4
Az	Az	7
Normal	Normal	13
Yksek	Yksek	25
ok Yksek	ok Yksek	50

Erken prototip oluşturma seviyesi

Bu seviyede, yeniden kullanım oldukça yaygındır. Bu yüzden plan tahmininde kullanılan işlev puanları beklenen yeniden kullanım yüzdesini hesaba katarak ayarlanmalıdır.

$$\text{Çaba} = (\text{İşlev puan sayısı} \cdot (1 - \% \text{Yeniden Kullanım} / 100)) / \text{üretkenlik}$$

Erken tasarım seviyesi

Erken tasarım seviyesinde üretilen tahminler, algoritmik modeller için standart formülü temel alırlar.

$$\text{Çaba} = A \times \text{Büyüklik}^B \times M$$

- Boehm, A katsayısının bu seviyede yapılan tahminler için 2.5 olmasını önermiştir. B üssü, projenin büyüklüğü arttıkça, artan gerekli çabayı yansıtır. M çarpanı, ürün güvenilirlik ve karmaşıklığı, gerekli olan yeniden kullanım, platform zorluğu (PDIF), personel yeteneği (PERS), personel deneyimi (PREX), plan (SCED) ve destek araçları (FCIL)'nı içeren proje ve süreç sürücülerini içeren basitleştirilmiş kümeyi temel alır.

Mimari sonrası seviyesi

- Mimari sonrası seviyesinde üretilen tahminler, erken tasarım tahminlerinde kullanılan aynı temel formülü temel alırlar.
- Mimari sonrası modelindeki başlangıç tahminlerini ayarlamak için kullanılan özellikler dört sınıfa karşılık gelirler:

Ürün özellikleri

Ölçüt	Açıklama
RELY	Sistem güvenilirliği
DATA	Veritabanının büyüklüğü
CPLX	Modüllerinin karmaşıklığı
DOCU	Gerekli belgelenmenin büyüklüğü
RUSE	Yeniden kullanılabilir bileşenlerin yüzdesi

Bilgisayar özellikleri

Ölçüt	Açıklama
TIME	Çalışma zamanı kısıtları
STOR	Bellek kısıtları
PVOL	Platformun uçuculuğu

Mimari sonrası seviyesi

Personel Özellikleri

Ölçüt	Açıklama
PCAP	Programcı yeteneği
ACAP	Proje analistinin yeteneği
PCON	Personel devamlılığı
AEXP	Proje sahasında analist deneyimi
PEXP	Proje sahasında programcı deneyimi
LTEX	Dil ve araç deneyimi

Proje özellikleri

Ölçüt	Açıklama
TOOL	Yazılım araçlarının kullanımı
SITE	Çok siteli çalışma büyüklüğü ve site haberleşmelerinin kalitesi
SCED	Geliştirme planı sıkıştırma