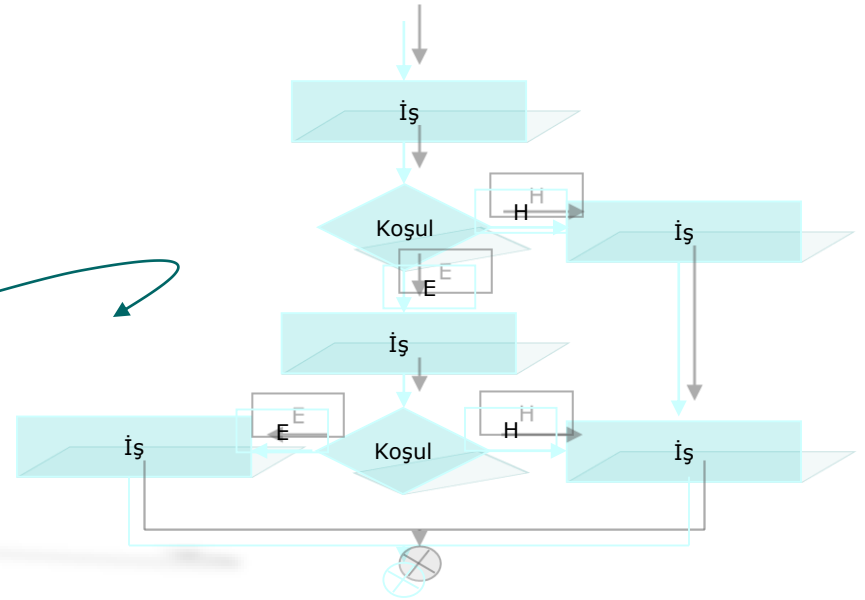


Sistem

Tasarımı



Tasarım , herhangi bir mühendislik sürecindeki ilk adımdır.

Genel olarak deneyim bilgi birikimiyle desteklenen çeşitli kurallarla yapılır.

Çeşitli geliştirme teknikleri , tanımlama ve tasarım yöntemleri bulunsa da Yazılım mühendisliği hala bir “**sanat**” niteliğindedir.

En önemli adımlarından birisi **Veri Tasarımı** dır.

Çözümleme sırasında toplanan bilgilerin kullanılacak veri yapılarına dönüştürülmesini içerir.

Mimari Tasarım ;

Yazılım birimlerinin yapısal parçalarını , birbirleriyle ilişkilerini tanımlar.

Yordamsal Tasarım ;

Yazılılımı oluşturan yapısal birimler yordam ve fonksiyonlar haline dönüştülür.

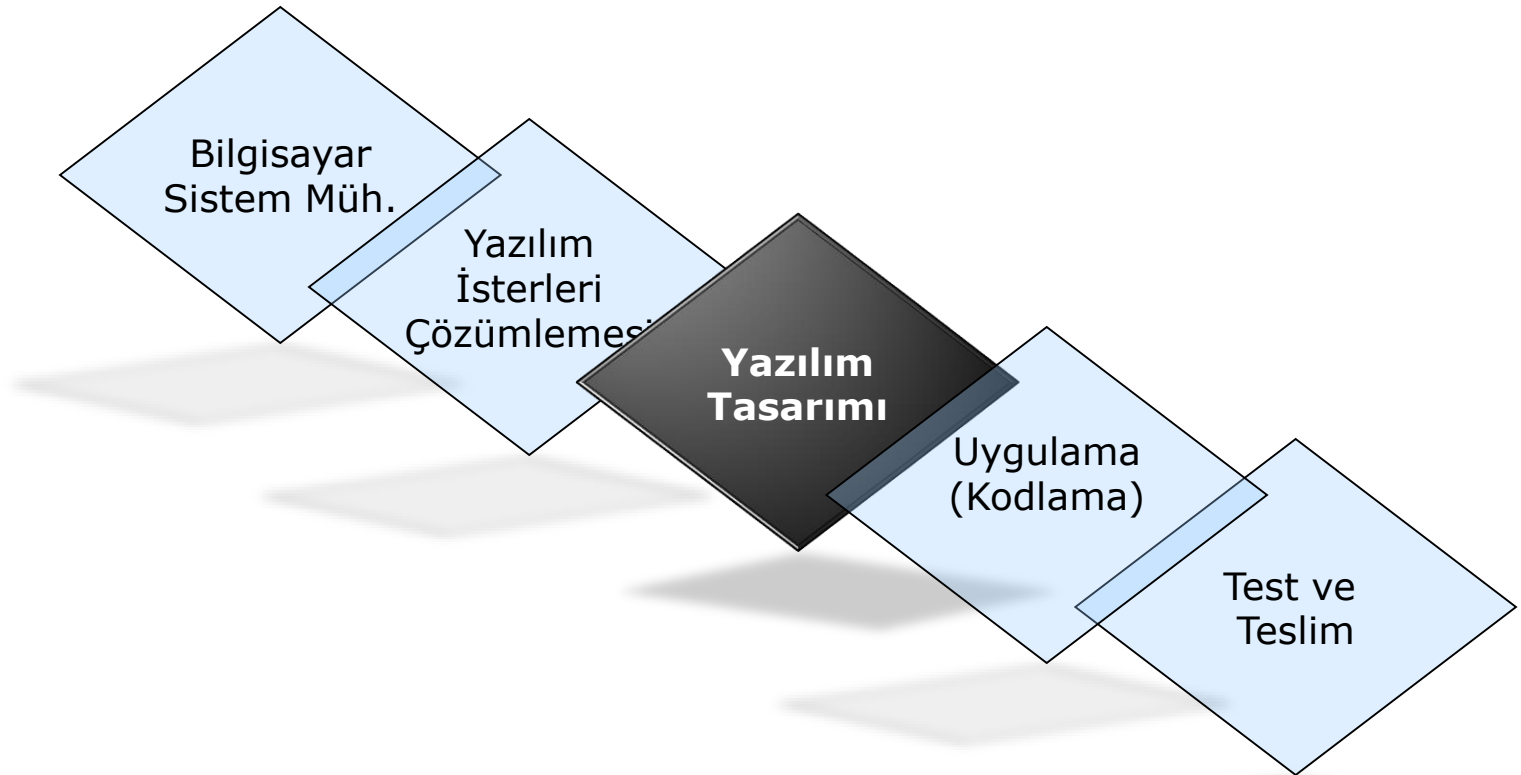
Arayüz Tasarımı ;

İnsan – Makine etkileşiminin şeklini alt sistemlerle olan arayüzlerin ayrıntılarını içerir.

Tüm detaylar belgede toplanır , değerlendirilir sonra da kodlama aşamasına geçilir.

Tasarım , yazılım testine kadar her şeyi etkilediğinden nitelik unsurunun öne çıktığı ilk aşama özelliğini taşımaktadır.

Yazılım geliştirme süreci içerisinde tasarım aşamasının yeri



Tasarımın ilk amacı Basitlik Olmalıdır.

“Sistem öyle tasarlanmalıdır ki, bir dizi değişiklik yapılsa bile sistem tasarımı hala basit kalabilmelidir.”

Yazılımın tasarımında gözden geçirilmesi gereken temel ilkelere en önemlileri şunlardır ;

1. **Soyutlama** : Denetimi ve anlaşılabilirliği arttırmak üzere en az ayrıntı ve işlem yapmaktır.
2. **Bilgi Gizleme** : Modüllerin iç yapılarını diğerlerinden gizlemek, bu şekilde karmaşıklığı engellemek ve soyutlamayı arttırmaktır.
3. **Kapsama** : Tüm isterlerin eksiksiz olarak karşılanması amacıyla yordam ve verilerin denetim altına alınması.

Tasarım Nitelikleri :

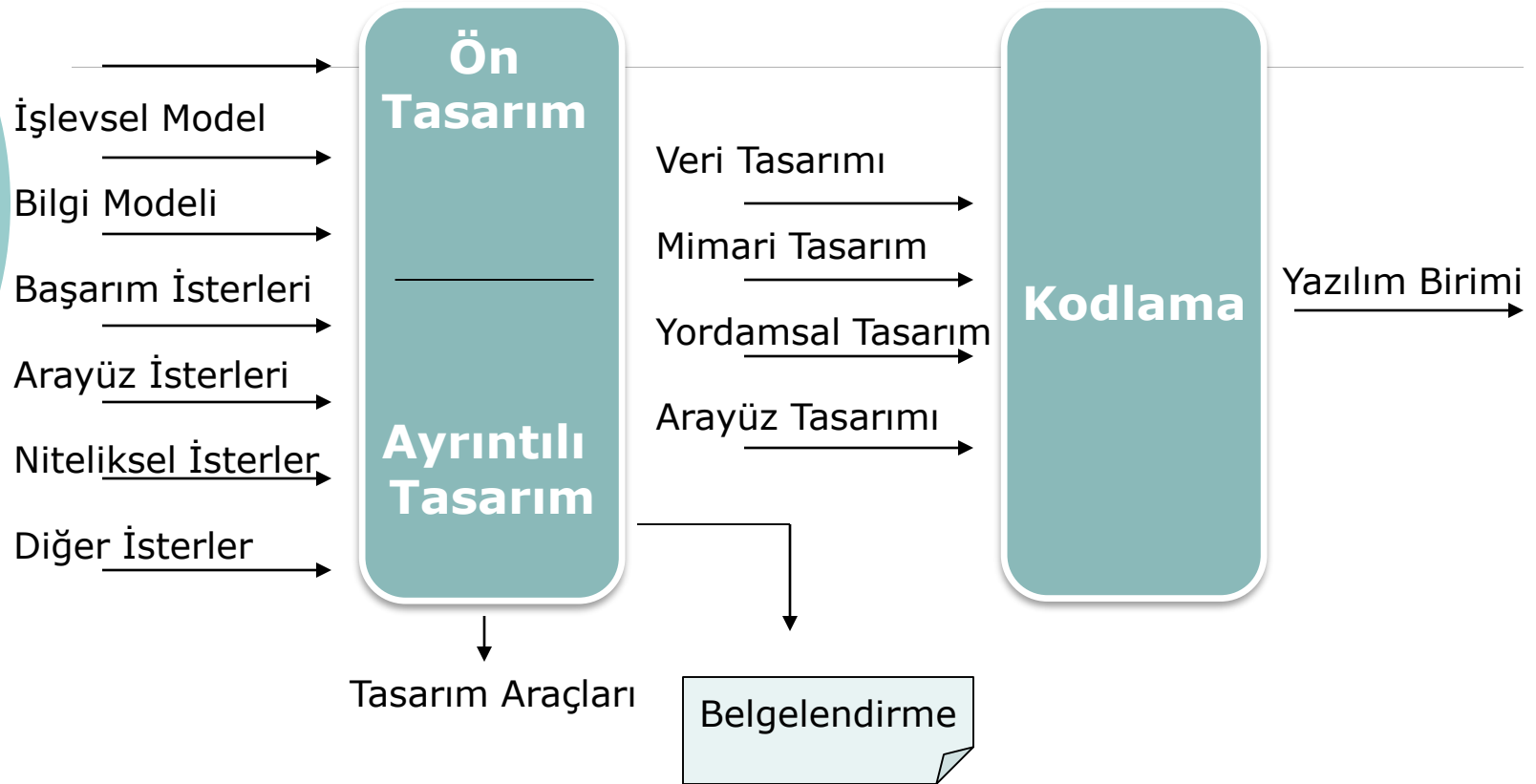
1. İsterler ile izlenebilirliği olmalıdır.
2. Geliştirilen birimin kodu ve testleri ile izlenebilirliği olmalıdır.
3. Programlama dilinden mümkün olduğunca bağımsız olmalıdır.
4. İşlevselliği , başarımı ve güvenilirliği yüksek bir ürün oluşturulmalıdır.
5. Yürütme sırasında oluşabilecek hataların ilgili iş sürecini aksatmayacak şekilde kotarılması sağlanmalıdır.
6. Öğrenmesi ve kullanımı kolay bir ürünü hedeflemelidir.
7. Tekrar kullanılabilir olmalıdır.
8. Bir ürün ailesine temel oluşturabilmelidir.
9. Kolay anlaşılmalıdır.
10. Gerektiğinde kolaylıkla değiştirilebilmelidir.
11. Kurumsal tasarım standartlarına uygun olmalıdır.
12. Diğer tasarımlarla birleştirilebilmesi mümkün olmalıdır.

Yazılım tasarımı sürecinde ve tanımlamalarında rehber olarak bazı standartlar kullanılabilir.

Yönetsel olarak süreç iki aşamada ele alınabilir :

- **Ön tasarım** : isterlerin veri ve mimari tasarımına dönüştürülmesidir.
- **Ayrıntılı tasarım** : Veri ve mimari tasarımın ayrıntılı veri yapıları ile algoritmik gösterime dönüştürülmesidir.

Yazılım geliştirme sürecinin tasarım aşaması sırasında kullanılan veri akışı :



Veri Tasarımı :

Veri yapısı ve veri modeli iç içe geçmiş iki kavramdır.

Birisi verinin bellekte tutulması veya saklanmasıyla ilgilenirken diğeri veriler arasındaki ilişki ve bağıntılar konusıyla ilgilenir.

Veriler üzerinde işlem yapacak olan algoritmalar da bu veri modellerine göre tasarlanırlar.

İyi bir veri tasarımı için neler gereklidir ;

- Veri yapıları/veri modelleri üzerinde yapılacak işlemlerin tanımlanması.
- Veri sözlüğünün oluşturulması.
- Döngüsel bir yol izlenmesi.
- Veri yapıları yalnızca kendilerini kullanan modüllere görünür olmalıdır.
- Sık kullanılan veri modelleri kütüphane haline getirilmelidir.
- Programlama dili özellikleri kullanılarak kodlama yapılmalıdır.

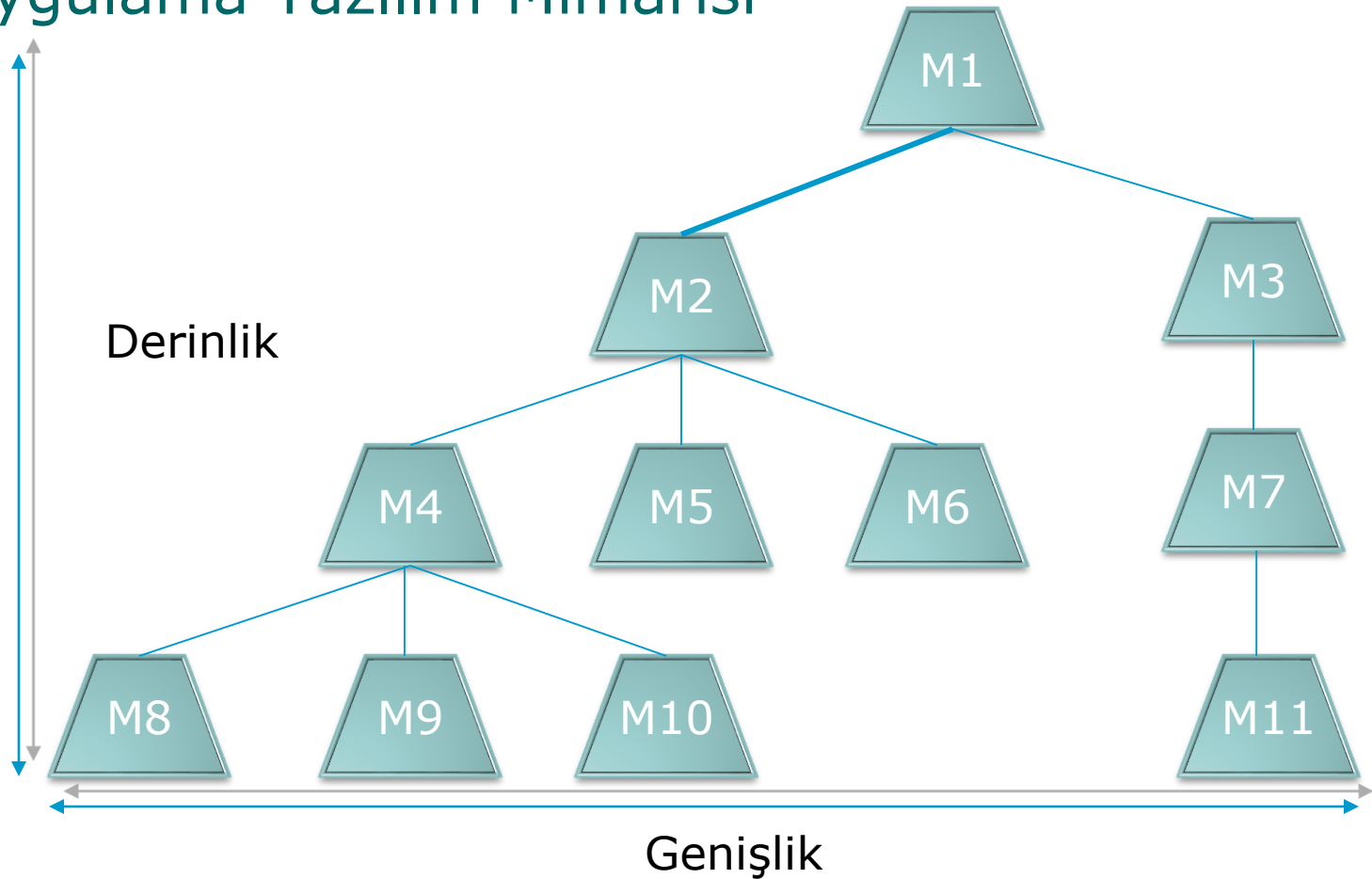
Mimari Tasarım ;

Uygulama yazılımı bir problemin çözümünü çeşitli parçalara bölerek sağlayabilir.

Parçaların yazılımdaki karşılığı **modüller** dir

Modüllerin sıradüzensel ilişkilerini gösteren yapıya **uygulama yazılım mimarisi** denir.

Uygulama Yazılım Mimarisi



- Uygulama alanının özellikleri :
Donanım özellikleri ...
- Uygulama yazılımının karmaşıklık derecesi :
Basit uygulamalar , tek program içinde , hertürlü arayüz ve bilgi işlemeyi kapsayacak şekilde geliştirilebilirler.Bölümlemek.
- Kullanıcı arayüzü kısıtlamaları :
Bilgi işleme birimleri ile kullanıcı arayüzünün farklı mimariye sahip işlemcilerde çalışması gereken durumlar olabilir.
- Taşınabilirlik :
Farklı işletim sistemi ve donanım özelliklerinde de çalışabilmesi gereklidir.

Yapısal Programlama Gösterimi :

Yazılım tarihinin en eski tasarım yöntemlerinden biri belirli yapıları kullanarak işlevleri metinsel bir şekilde anlatmaktır.

Tasarım dillerinin ortak özellikleri :

- Her türlü yapıyı destekleyebilen sabit bir anahtar sözcük listesi.
- Veri tipleri ve veri yapıları tanımlama yeteneği.
- Alt program tanımlama ve çağırma düzeneği.
- Bilgi işlemeyi serbest bir dille anlatabilme olanağı.
- Arayüz tanımlama yeteneği.
- Koşul ve çevrim yapıları.
- Giriş / Çıkış yapıları
- Zaman belirtimleri.

Grafiksel Gösterim :

Bazen bir resim bir çok satırdan oluşan bir anlatım yerine geçebilir.

Bu gerçekten hareketle grafiksel gösterim yöntemleri bulunmuş, bu yöntemleri kullanan yazılım araçları geliştirilmiştir.

Gösterim şekillerinin iyi bilinmemesi sonucu tasarımı yanlış anlaması, hatalı kodlamaya neden olabilir.

“Grafiksel gösterimlerin iyi öğrenilmesi ve anlaşılması gereklidir.”

Yapısal çözümleme ve tasarım :

Yapısal çözümleme ve tasarımda veri akış diyagramları ve durum geçiş diyagramları kullanılır.

UML :

Nesneye yönelik çözümleme ve tasarımın hem metinsel hem de grafiksel olarak yapılabilmesine yardımcı olan uluslar arası çevrelerce kabul edilmiş, standart ve yaygın bir tanımlama dilidir.

Akış diyagramları :

Çeşitli tasarım yöntemlerinde kullanılabilecek görsel anlatımları ve diyagramları ikiye ayırmak mümkündür.

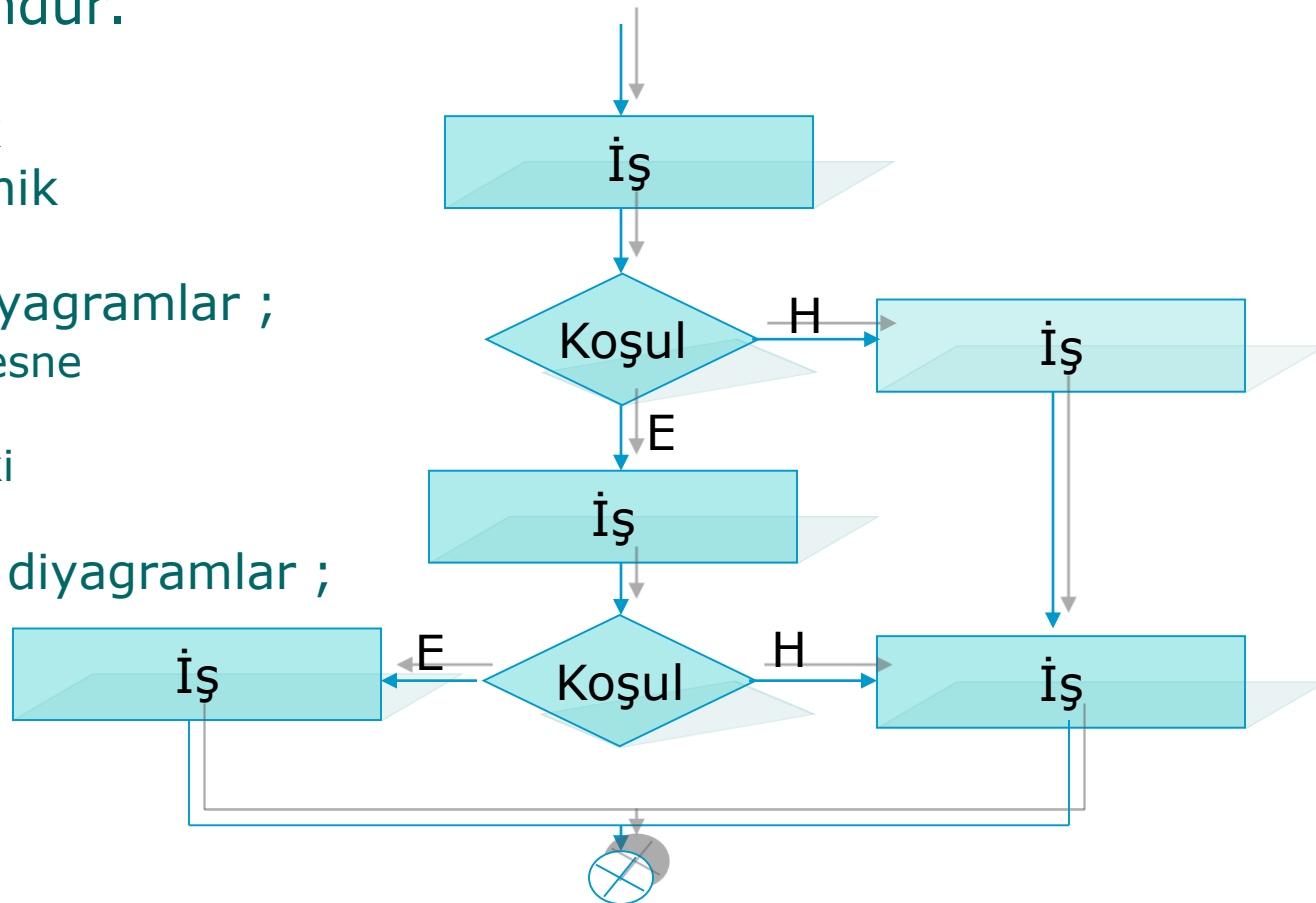
1. Statik
2. Dinamik

Statik akış diyagramları ;

Sınıf ve nesne
Bileşen
Varlık-iliski
Yapı ...

Dinamik akış diyagramları ;

Veri akış
Etkileşim
Durum
Akış ...



- <http://e-bergi.com/2008/Ekim/Cevik-Modelleme-ve-Cevik-Yazilim-Gelistirme>
- Sistem Analizi ve Tasarımı Prof.Dr. Oya Kalıpsız
- Yazılım Mühendisliği Dr.M.Erhan Sarıdoğan
- BT HABER dergisi, Sayı 259, 2000.
- DELPHI UNLEASHED, SAMS PUBLISHING, Charles Calvert, 1997.
- www.mehmetduran.com
- <http://jamshidhashimi.com/2010/08/23/agilecevik-modelleme-ve-cevik-yazilim-gelistirme>
- http://en.wikipedia.org/wiki/Agile_Modeling
- http://en.wikipedia.org/wiki/Agile_software_development
- <http://www.minepla.net/2008/10/agilecevik-yazylym-gelithirme>

