



Yazılım İsterlerinin Çözümlemesi

Yazılım Yaşam Çevrimi

- “ Yazılım Yaşam çevrimin herhangi bir yazılım geliştirme uzmanı ya da deneyimli bir bilişim yöneticisi tarafından kolayca birkaç farklı biçimde ayrıntılandırılması mümkündür. ”
- “ Hangi aşamalardan geçilirse geçilsin, nasıl bir geliştirme yöntemi uygulanırsa uygulansın, kullanıcının **başlıca üç noktada bu çevrime yaşamsal katkısı** vardır: ”

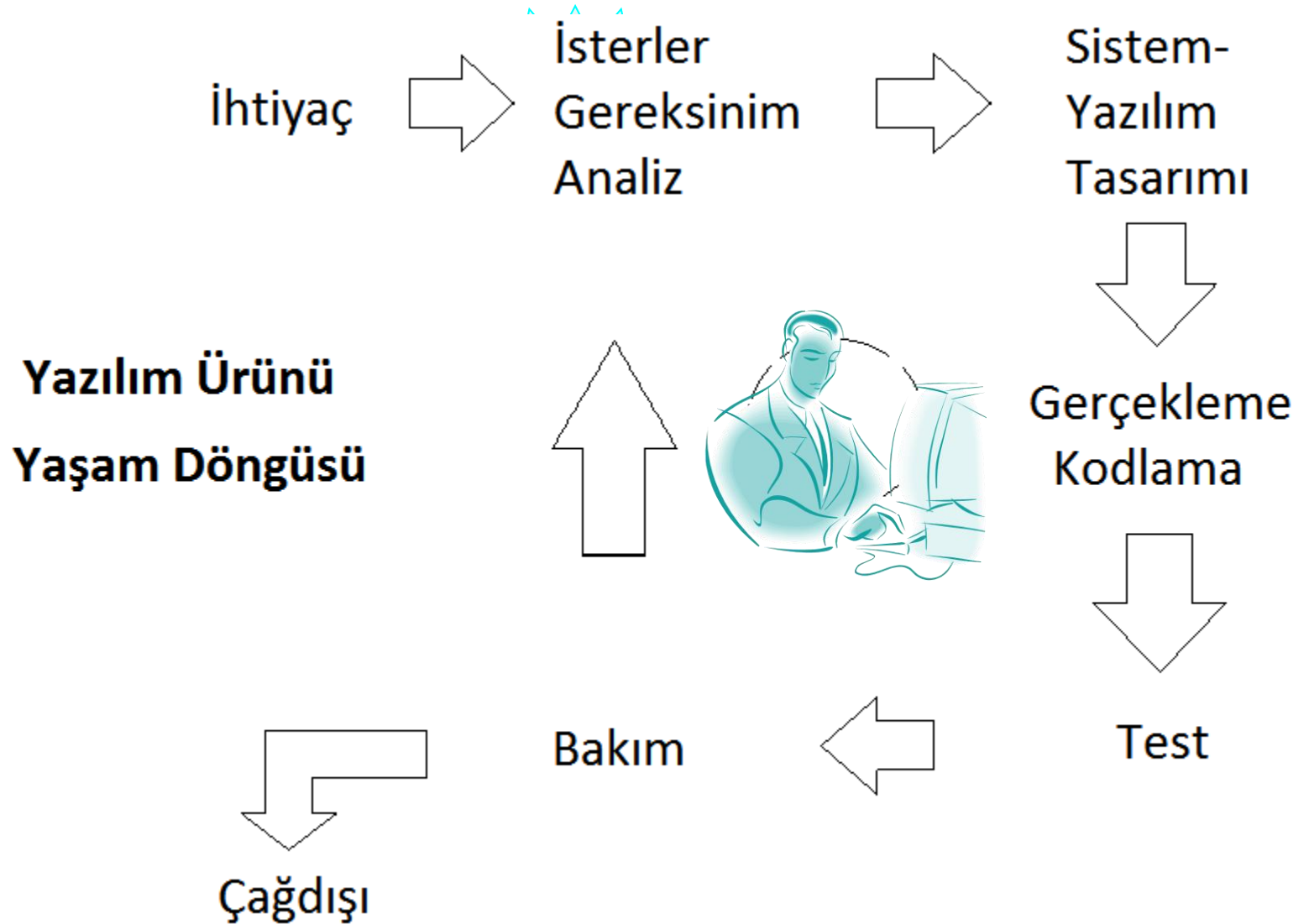
1. **İsterlerin belirtilmesi,**

2. **Yazılımın kabulü,**

3. **Alınan sistemin kullanım sonrası değerlendirmesi.**



Yazılım İsterlerinin Çözümlemesi



Sistem geliştirme aşaması ve İşlemleri

I. Tanımlama Aşaması

Ön araştırmayla İlgili İşlemler;

1. Proje sözleşmesi
2. Hedeflerin ve yönlerin belirlenmesi
3. Öneriler, seçenekler ve beklenen yararların/sonuçların belirtildiği bir rapor
4. Ön sistem incelemesi İçin zamanlama ve maliyet tasarısı
5. {3} ve (4)'ü kullanıcılara ve yönetime sunarak bir sonraki adıma geçme yetkisi

Ön sistem İncelemesiyle İlgili aşamalar:

6. Şimdiki sistemin maliyeti ve İlgili belgelerin toplanıp incelenmesi
7. Giriş/çıkış gerekleri belirlenmesi
8. Üst düzey iş akışı çıkarılması
9. Yazılım paketleri için ön liste hazırlanması
10. Maliyet ve getiri öngörüsü
11. Bir sonraki adım için zamanlama ve maliyet
12. (11)'i kullanıcı ve yönetime sunarak bir sonraki adıma geçiş

Sistem çözümleme ile ilgili işlemler

13. Toplanan verilerin doğrulanması
14. Geliştirme, sinama ve uygulama planı
15. (9)'da hazırlanan paketler ön listesiyle işlevsel gereklerin (7,8) karşılaştırılması
16. Bir sonraki adım için sistem tasarım planı hazırlanması
17. Kullanıcı ve yöneticilerden bir sonraki aşama için onay

II. Geliştirme Aşaması Sistem Tasarımıyla ilgili İşler

18. Veri tabanı tasarımı ve değerlendirmesi
19. Donanım gerekleri belirlenmesi
20. Sınama verilerinin saptanması
21. Her bir işlevin mantık tasarımı
22. Giriş/çıkış gerekleri kesinleştirilmesi
23. Uygulama/geçiş planı ve kullanıcı işlemlerinin belirlenmesi
24. Programlama planı
25. Kullanıcı ve yöneticilere (24)'ün sunulduğu ve bir sonraki adıma geçiş

Programlama ile ilgili işler

- 26. Sistem gerekler tanımının gözden geçirilmesi ve Programcıların görevlendirilmesi
- 27. Veri tabanı tasarımı (18) ve program tanımının (21, 22, 24, 26) gözden geçirmesi
- 28. Tasarım üzerinden gidilmesi (Yapısal gözden geçirme)
- 29. Sınama verilerinin hazırlanması
- 30. Program yazımı ve düzeltimi
- 31. Program ve işletim belgeleri hazırlanması
- 32. Tüm sistem sinaması
- 33. Uygulama için yönetim onayı

III. Uygulama Aşaması Sınama ve yerleştirmeye İlgili işler.

- 34. Geçiş ve sınama planları
- 35. Kullanıcı eğitimi
- 36. Kullanım ve işletim el kitaplarının elden geçirilmesi
- 37. Kullanım ortamında tüm sistemin sınanması
- 38. Sonuçların değerlendirilmesi ve geçişin gerçekleştirilmesi
- 39. Kullanıcı kabulü

Proje değerlendirme ile ilgili İşler

- 40. 1–6 ay içinde kullanıcı değerlendirmesi
- 41. Sistem geliştirme süreciyle ilgili öneriler geliştirilmesi.

“ Yazılım yaşam çevriminin hangi aşamalara ayrıldığı ve bunların gerçekleştirilme sırası, yazılım geliştirme yaklaşımına bağlıdır. ”

Geleneksel yaklaşım

1. Kullanıcı isterlerinin işin başında yeterli açıklık ve doğrulukla tanımlanabileceği,
2. Bu isterlerin tanımında kullanıcı ile geliştirici arasında anlaşmaya varılabileceği, bunun için yeterli zaman ve diğer kaynakların ayrılabilceği,
3. İsterler tanımlama ve tasarım aşamaları yeterli nitelikte hazırlık oluşturarak sonuçlanmadan programlamaya geçilmeyeceği için, geliştirme projesinin başlamasından uzun bir süre sonraya kadar kullanıcıya hiçbir ürün verilmemesinin kabul edilebileceği durumlarda olumlu sonuç verecektir.

Yazılım Projesi Gerekçesi :

1. Gereken yazılımın kısa tanımı.
2. Gereksinim nedenleri.
3. Söz konusu işin bugün nasıl görüldüğü.
4. İstenen yazılımın sağlayacağı hizmetler.
5. Bu yazılımdan yararlanacak kuruluş birimleri ve işlevleri.
6. Beklenen proje maliyeti ve öngörülen gerçekleşme süresi.

“ Yazılım projesinin gerekçesi oluşturulurken, bilişim aşamalarının gelişmesi de bilinir ve proje tanımında yol gösterici olursa, hem çalışma daha güvenilir hedeflere yönelecek, hem de yönetim katmanlarının desteği daha kolay sağlanacaktır. ”

İsterler Anlatımı

- Tüm özel terimler tanımlanmalı,
- Önce genel, sonra artan düzeyde ayrıntıya yer verilmeli,
- Çizelge ve resimler kullanılmalı,
- Tüm isterler çizelgelerle ilintilendirilmeli,
- İsterler somut olarak ölçülebilecek en üst (ayrıntısız) düzeyde tanımlanmalı,
- Belgenin bölümleri arasında tutarlılık sağlanmalı; bütünlük içinse:
- Kullanıcı gerekleri ve istenen başarımlar düzeyleri belirtilmeli, (bunları sağlayacak yazılımın nasıl kurulacağı değil,)
- İşlevsel isterler (ne sağlanacağı),
- Veri tabanı (hangi bilgilere dayanılacağı),
- Başarım isterleri-ölçütler (Kullanıcı sayıları, veri hacim ve sıklıkları) belirtilmeli,
- Kullanılacak donanım ve destek yazılımı ile ilgili kısıtlayıcı ya da belirleyici özellikler belirlenmeli,
- geliştirmeye ilgili kısıtlar (zaman, öncelik sırası, yazılım mülkiyeti, v.b.) varsa belirtilmelidir.

İsterler Anlatımı

“İsterler belgesinin bütünlüğü, örneğin, yazılımın olabilecek tüm veri girişlerinde, görülecek tepkisini tanımlamak anlamına gelir.

Yalnızca doğru ve tutarlı verilerle ne yapılacağı anlatılıp hatalı, eksik ya da çelişkili verilere nasıl tepki gösterileceğinin belirlenmemesi, bu açık noktanın tasarım aşamasında belki de tümüyle kullanıcının beklentisine uymayan biçimde karşılanmasına yol açabilir. ”

İsterler İçeriği :

Yazılım isterlerinin belirtilmesinde üç yöntem kullanılabilir ;

1. **Girdi/çıkıtı belirtimi,**
2. **Belirleyici örnekler,**
3. **Modeller.**

Girdi/çıkıtı belirtimi:

Olası girdilerin üreteceği tepkileri ya da istenen çıktıları almak için verilecek girdileri belirtmeye dayanan bu yöntem, tüm olası girdileri göz önüne almanın güçlüğü nedeniyle bazı durumlarda kullanılamayabilir.

Belirleyici örnekler:

Tüm girdiler ele alınmamakla birlikte bazı durumlarda iyi seçilmiş bazı örneklerle sistemin nasıl çalışmasının gerekliliği anlaşılmayan nokta bırakmayacak biçimde açıklanabilir.

Modeller:

Soyut (örneğin matematiksel ya da çizgesel) modellerin oluşturulabildiği durumlarda, diğer yöntemlerden çok daha kesin belirtilmelerin elde edilmesine yarayabilir.

Yazılım İsterler Belgesi İçindekiler Önerisi:

1. Giriş

- 1 Amaç
- 2 Kapsam
- 3 Tanım ve kısaltmalar
- 4 Göndermeler
- 5 Özet

2. Genel belirleme

- 1 Ürünün yeri
- 2 Ürünün işlevleri
- 3 Kullanıcı özellikleri
- 4 Genel kısıtlar
- 5 Varsayım ve bağımlılıklar

3. Özel isterler Ekler

- 1 ...

Yazılım İsterleri Belirlemede Biçimsel Yöntemler

- Biçimsel olmayan kullanıcı isterlerinin sistem belirtimine dönüştürülmesini kolaylaştırmalı,
- Belirtimin tutarlılık, bütünlük gibi niteliklerinin denetlenmesine olanak sağlamalı,
- Ortaya çıkan sistem belirtiminin gerçekten kullanıcı isterlerine karşı gelip gelmediğini görebilmek için kullanıcının biçimsel olmayan diline dönüştürülebilmeli.

Yazılım İsterlerinin Çözümlemesi

1. İşlevsel isterlerin anlaşılabilir, kesin ve kullanıcı açısından anlamlı biçimde ortaya konulabilmesi,
2. Belirtimin içsel tutarlılığı, kullanıcı örneklerini destekleyebilmesi, işlem sonuçlarının doğruluğu, v.b. özelliklerin doğrulanabilmesi,
3. Kullanıcının kaynak ve başarımlarına uyan bir gerçekleştirimin oluşturulabilmesi, işlevsel isterleri karşılayan, işlevsel-dışı olanları da kısıtlamayan bir belirtimin ortaya konulabilmesi,
4. Belirtimin oluşturulması, doğrulanması ve gerçekleştirilmesinin, kullanışlı biçimde yapılabilmesi,
5. Belirtim yönteminin kullanımının, ekonomik olarak olurlu olması.

Yazılım Niteliği Ölçütleri

Başarıma ilişkin etmenler:

Verimlilik : Kaynak kullanımı.

Güvenilirlik : Sonuçlara ne ölçüde güvenildiği.

Tutarlılık : Yazılım birimlerinin ne ölçüde çelişkisiz sonuç verdiği.

Kalımlılık : Uygunsuz koşullarda çalışabilme yeteneği.

Kullanılabilirlik : Kullanım ve öğrenim kolaylığı.



Tasarıma ilişkin etmenler:

Doğruluk : İsterlere uygunluk düzeyi.

Bakım kolaylığı : Onarım olanağı.

Doğrulanabilirlik: Sonuçların doğruluğunun ne ölçüde denetlenebildiği.

Genişletilebilirlik: Yeteneklerin artırılabilirle, yazılımın geliştirilebilme düzeyi.

Uyarlanabilirliğe ilişkin etmenler:

- Esneklik** : Değiştirilebilme, yeni uygulamaları destekleme ölçüsü. Uyumluluk: Çalışabilme, uyum düzeyi.
- Taşınabilirlik:** Farklı donanımlar ve destek yazılımları üzerinde çalışabilme düzeyi

“Bu etmenler aynı zamanda özellikle satın alınacak yazılımın değerlendirilmesinde kullanılmaya uygun nitelik ölçütlerininide oluşturmaktadırlar. ”

Nitelik etmenleri:

Yazılımın çalışmasına ilişkin etmenler

- Doğruluk** : isterlere uygunluk.
- Güvenilirlik** : Sonuçların doğruluk ve duyarlılığı.
- Verimlilik** : Yazılımın gerektirdiği bilgisayar kaynakları, zaman ve program uzunluğu.
- Korunmuşluk**: Yetkisiz kullanıma ve zarara karşı korunma düzeyi.
- Kullanılabilirlik** : Öğrenim, işletim ve genel olarak kullanıma yatkınlık düzeyi.

Yazılımın gelişmesine ilişkin etmenler

Bakım kolaylığı: Görülen bir hatanın nedeninin bulunması ve giderilmesi için gereken emek (10. Bölümde bu konu daha ayrıntılı olarak incelenecektir.)

Esneklik: Çalışan bir programda değişiklik yapmak için gereken emek.

Sınanabilirilik: Yazılımın istenen işlevi gerçekleştirdiğini sınamak için gereken emek

Yazılımın uyumluluğuna ilişkin etmenler

Taşınabilirlik: Yazılımın bir yazılım/donanım ortamından diğerine aktarılması için gereken emek.

Desteleyicilik: Yazılımın ya da parçalarının yeni uygulamalarda kullanılabilirliği

Uyumluluk: Başka yazılım sistemleriyle işbirliği yapabilme düzeyi.

Yazılım İsterlerinin Çözümlemesi

Yazılım Niteliği Ölçütleri

Kurallara uyum : Standartlara ve kuruluş içindeki yazılım üretimi kurallarına uyum düzeyi.

Hatasızlık : işlemlerin ve denetimin hatasızlık duyarlılığı.

İletişim rahatlığı: Standart arabirimlerin, protokollerin ve iletişim yöntemlerinin kullanılma düzeyi.

Tamlık : İsterlerde belirtilen işlevin gerçekleştirilme düzeyi.

Karmaşıklık : Yazılımın yapısal karmaşıklığı. Çağırılan yordam sayısı, koşulların ve kararların, girdi çıktı birimlerinin fazlalığı v.b. ayrıntılara bağlı olarak yazılım karmaşıklığının tanımı için bkz. [McCabe, 1976]. Önerilen en basit karmaşıklık ölçüsü, yazılımın satır sayısı olarak uzunluğudur. Burada yine (0-10) aralığında bir değerlendirme söz konusudur.

Özlülük : Birçok araştırmacı, diğer ölçütlerin, özellikle de karmaşıklığın yazılım uzunluğuna bağlı olduğu görüşündedir. (Ör. [Li, 1987]) Özlülük, gerçekleştirilen işlev sayısının yazılım uzunluğuna oranı olarak tanımlanabilir.

Tutarlılık : Yazılımın tüm birimlerinde ve geliştirme sürecinin tümünde tutarlı bir tasarım ve belgeleme yönteminin kullanılması.

Veri rahatlığı : Tüm yazılımın standart ve uyumlu veri yapıları kullanması.

Hata hoşgörüsü: Hata durumunda işlemi sürdürme kolaylığı.

Yazılım İsterlerinin Çözümlemesi

Verimli çalışma :

Yazılımın başarımı; yapılan işin kısa zamanda ve az kaynakla gerçekleştirilmesi.

Genişletilebilirlik :

Yazılım mimarisinin, veri yapılarının ve süreç tasarımının genişletilebilme yeteneği.

Genellik :

Yazılımın kullanım alanının genişliği.

Donanımdan bağımsızlık:

Yazılımın işleyebilmesi ya da belli başarımlarına uyabilmesi için donanıma ne ölçüde bağımlı olduğu.

Denetimlilik :

Yazılımın kendi işleyişini denetleme, başarımlarını ve oluşan hataları izleme yeteneği.

Birimsellik :

Yazılım birimlerinin işlevsel tutarlılık ve birbirinden bağımsızlık düzeyi.

Yazılım İsterlerinin Çözümlemesi

İşletilebilirlik:

Yazılımın çalıştırılma kolaylığı.

Güvenlik:

Programları ve verileri koruyan düzeneklerin etkililik düzeyi.

Anlaşırılık:

Yazılımın, özellikle kaynak kodunun açıklayıcı belgeleme içerme düzeyi.

Basitlik:

Yazılımın kolay anlaşılabilme düzeyi.

Yazılımdan bağımsızlık:

Yazılımın standart olmayan yazılım desteklerine, işletim sistemi, v.b. yazılım sistemlerine bağımlılık

Açıklanabilirlik:

Yazılımın yapısal ya da işlevsel özelliklerinden her birinin, hangi isterlerden kaynaklandığının açıklığı.

Eğitim:

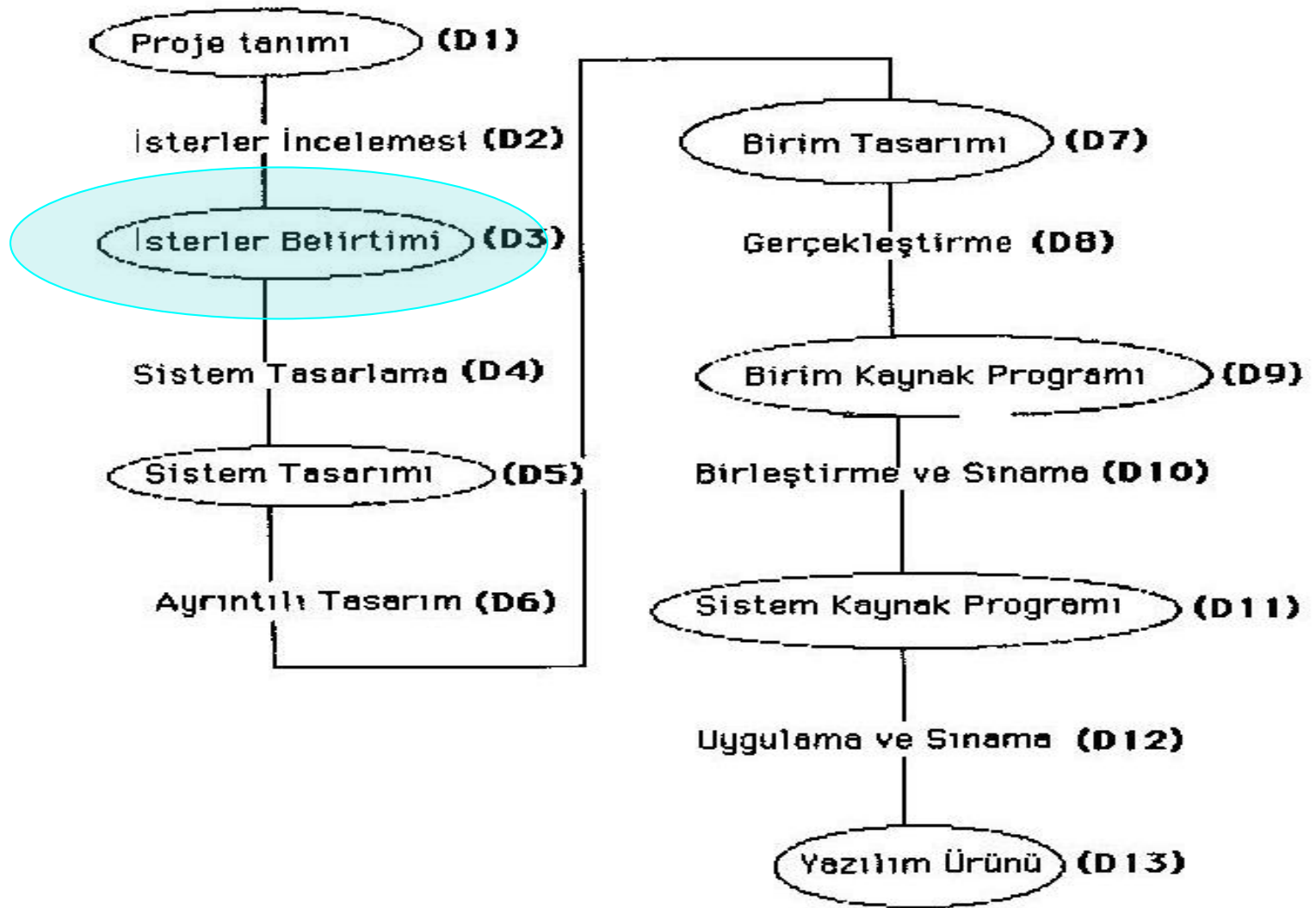
Yazılımın, yeni kullanıcılara ne ölçüde kolayca kullanım olanağı sunduğu ve eğitim gerektirdiği.

Yazılım Sürecinin Nitelik Üzerinde Etkisi

Yazılım niteliğine ilişkin olarak yapılan daha yeni çalışmalar, niteliğin, özellikle de güvenilirlik öğesinin, yalnızca son yazılım ürünü üzerinde belli bir zamanda yapılacak incelemeyle saptanamayacağı görüşüne ağırlık vermektedir.

Yalnızca son ürünün değil, yaşam çevriminin çeşitli aşamalarındaki ara ürünlerin de incelenmesi yazılım niteliğinin belirlenmesinde gerekli görülmektedir.

Yazılım İsterlerinin Çözümlemesi



Yazılım nitelik öğeleri dizisi.



VERİ AKIS DİYAGRAMI GELİŞTİRME