

altsisteme aktarırlar; altsistemden gelen verileri de gerekli dönüşümden geçi-
rerek uygun bir şekilde ana sisteme aktarırlar, yani altsistemi tümleştirirler
(integration). Altsistem arayüz yazılım birimleri, seçilen yönteme göre, her alt-
sistem için ayrı bir bilgisayar üzerinde koşabileceğgi gibi, bir bilgisayar üzerinde
bir ya da daha fazla yazılım birimi ile birden fazla altsistemin tümleştirilmesi
de olasıdır.

2.7. Özет

Sistemler doğal ve insan yapısı olarak iki gruba ayrılabilir. Bilgisayar tabanlı sistemler
de kullanım yerlerine ve taşıdıkları özelliklere göre gruplandırılırlar. Bilgisayar
sistemleri yapıları, donanım özellikleri ve üzerlerinde yüklü yazılımlar açısından ana
sistem, hedef sistem ve test sistemi olarak üçe ayrılabilir. Bir bilgisayarlı sistemi
oluşturan ana bileşenler, donanım, altyapı yazılımı ve uygulama yazılımıdır. Bilgi
işleme şekli ve mantığı, sistem mimarisini etkilediği için uygulama alanına uygun bir
mimari seçilmelidir. Esnek yapı ve genişleme yeteneği kazandırmak amacıyla yazılım
mimarileri genel, kullanıcı-sunucu mimarileri, iki ve üç katmanlı mimari, dağıtık
mimari olarak incelenmeli ve uygulama alanına göre seçim yapılmalıdır. Dağıtık
mimarilerde çeşitli yazılım birimi türleri kullanılabilmektedir.

2.8. Sorular

- 2.8.1)** Sistem nedir, tanımlayınız.
- 2.8.2)** Sistemleri gruplandırınız. Her grubu daha da alt grplara ayırarak örnekler
veriniz.
- 2.8.3)** Bilgi sistemini tanımlayınız.
- 2.8.4)** Bilgisayar tabanlı sistemleri gruplandırınız ve herbirine örnekler veriniz.
- 2.8.5)** Gerçek zamanlı sistemleri tepki sürelerine göre gruplandırınız. Herbiri için
bir örnek vererek gerçek zamanlı işlevleri açıklayınız.
- 2.8.6)** Veri tabanı yönetim sistemlerinin ana özelliklerini anlatınız.
- 2.8.7)** Otomasyon sistemleri hakkında bilgi veriniz.
- 2.8.8)** Kontrol sistemlerini gruplandırınız ve örnekler veriniz.
- 2.8.9)** Bir havaalanında bulunan hava trafik kontrol sistemini modelleyiniz.
Herbir bileşenin görevlerini sıralayınız. Böyle bir sistemin sahip olması gereken
önemli özellikleri anlatınız.
- 2.8.10)** Gömülü sistemler neden diğer sistemlerden farklıdır? Örnekler vererek
açıklayınız.
- 2.8.11)** Donanım ve yazılım özellikleri açısından sistem yapılarını gruplandırınız.

- 2.8.12)** Bir bilgisayarlı sistemi oluşturan ana bileşenler nelerdir?
- 2.8.13)** Ara katman kullanımının yarar ve sakıncalarını anlatınız.
- 2.8.14)** Uygulama yazılımlarını kullanım alanlarına göre sınıflandırarak anlatınız ve birer örnek veriniz.
- 2.8.15)** Merkezi ve dağıtık bilgi işleme mantığını, yapıları ve özelliklerini bakımından karşılaştırınız.
- 2.8.16)** Uygulama yazılımı ve donanım düzeyleri açısından yazılım mimarilerini gruplandırınız.
- 2.8.17)** Kullanıcı-sunucu, iki ve üç katmanlı mimarileri karşılaştırınız. Herbiri için örnek veriniz.
- 2.8.18)** Dağıtık mimarinin yarar ve sakıncalarını anlatınız.
- 2.8.19)** Yazılım birimi türlerini anlatınız. Bu turlere başka eklemeler yapılabilir mi?
- 2.8.20)** Soru 9'da modellediğiniz kontrol sistemi için olabilecek yazılım birimlerini isimlendirip üzerinde koşacağı bilgisayarlara atayınız.

Sistem mühendisliği bir dizi üst düzey mühendislik etkinliklerini içerir. Bundan sonra da, sistemi oluşturan donanım öğeleri donanım mühendisliği, yazılım öğeleri de yazılım mühendisliği süreçleriyle geliştirilirler. Kullanıcı etkileşimli sistemlerde en önemli etmen olan insan-makine arayüzü için de insan mühendisliği çalışmaları yürütülür.

Her türlü sistemin geliştirilmesine bir çözümleme aşamasıyla başlanmalıdır. Yapılabilirlik araştırması ardından, kavramsal tanımlama ve sistem isterleri belirtimi hazırlanır. Sistem tasarımlı sırasında ana isterleri karşılayacak mimari ve sistem yapısı belirlenir, donanım ve yazılım öğeleri tanımlanır, uygulama alanının gereksinimlerine göre altsistem tümleştirme mimarisi seçilir. Bu amaçla uygulama alanı ve sistem isterlerine uygun tümleştirme yöntemi belirlenmelidir.

Donanım hazır olarak piyasadan temin edilebileceği gibi özel olarak da geliştirilebilir. Yazılım da seçilen bir yönteme göre geliştirilerek tümlestirmesi yapılır. Bundan sonra donanım ve yazılım tümlestirme, sistemin kullanım yerine kurulumu ve testi yapılır. Eğitim ve kabul testlerinden sonra sistem kullanıcıya teslim edilerek bir bakım süreci başlatılır. Sistem mühendisliğini daha etkin bir şekilde yürütebilmek ve geliştirmeye otomasyon sağlamak üzere çeşitli yardımcı araçlar bulunmaktadır.

3.16. Sorular

- 3.16.1)** Bilgisayar sistemi mühendisliğini tanımlayınız.
- 3.16.2)** Yaşam çevrimi nedir? Sistem yaşam çevriminin ana evreleri nelerdir?
- 3.16.3)** Proje yaşam çevriminin aşamalarını anlatınız. Bu aşamaların örnek bir projede uygulanışını kısaca anlatınız.
- 3.16.4)** Sistem maliyetini tanımlayınız.
- 3.16.5)** Bir sistemin gerçekleştirilebilmesi için dikkate alınması gereken maliyet unsurlarını belirtiniz.
- 3.16.6)** Bilgisayar tabanlı sistem maliyetlerini geliştirme ve hazır alma durumları bakımından karşılaştırınız.
- 3.16.7)** Hazır olarak alınan bir sistem ile özel olarak geliştirilen bir sistemin maliyetini uzun vadede karşılaştırınız.
- 3.16.8)** Hangi maliyetler sistemin ilk geliştirme maliyetinden daha fazla olabilir?
- 3.16.9)** Teknik bir firma ya da kurumun sistem edinme veya geliştirme yapabilme yeteneğine göre stratejik hedeflerini anlatınız.
- 3.16.10)** Bir otomobil fabrikasında yeni model bir otomobilin üretilmesi için gerekli geliştirme sürecini anlatınız. Bu süreci bir mobil telefon üretimi ile karşılaştırınız.
- 3.16.11)** Bir proje yönetimini düzenlemek üzere hangi planlar hazırlanmalıdır?

- 3.16.12)** Sistem mühendisliği ne demektir, neleri içerir? Kisaca anlatınız.
- 3.16.13)** Donanım mühendisliği neleri içerir? Bir sistem tanımlayarak donanım mühendisliği işlerini anlatınız?
- 3.16.14)** Yazılım mühendisliğinin ilgi alanına hangi sistem bileşenleri girer?
- 3.16.15)** İnsan mühendisliği neden önemlidir? Örneklerle açıklayınız.
- 3.16.16)** İdeal bir sistem çözümleyicisinin alması gereken eğitimi ve sahip olması gereken özelliklerini tanımlayınız.
- 3.16.17)** Sistem çözümlemesinde yer alan ana etkinlikleri anlatınız.
- 3.16.18)** Sistem bileşenleri nelerdir?
- 3.16.19)** Bir otomobilin yol bilgisayarı için örnek bir çözümleme yapınız. Bu sistemin bileşenlerini belirleyiniz.
- 3.16.20)** Sistem isterleri hangi belgelerde, nasıl tanımlanır?
- 3.16.21)** Sistem tasarıımı ne gibi etmenlere dikkate edilerek yapılmalıdır?
- 3.16.22)** Tümleştirme ne demektir?
- 3.16.23)** Ne tür tümleştirme mimarileri kullanılır? Karşılaştırınız ve örnek vererek açıklayınız.
- 3.16.24)** Bir ana sistem alt sistemlerle nasıl iletişim kurar? Bir video kameradan bir bilgisayara sayısal ortamda film aktarmak ve kaydetmek için gerekli olan arayüzleri ve altyapı gereksinimini anlatınız.
- 3.16.25)** Sistemi oluşturan ana parçaları alt düzeylere ayırtarak tanımlayınız.
- 3.16.26)** Sistem mimari tasarıımı nedir? Bu aşamada neler yapılır?
- 3.16.27)** Sistem gerçekleştiriminin temel aşamalarını tanımlayınız.
- 3.16.28)** Yazılım geliştirme sürecinin ana etkinliklerini tanımlayınız.
- 3.16.29)** Sistem kurulumu hangi belgelerde anlatılır?
- 3.16.30)** Donanım kurulumu sırasında dikkat edilmesi gereken noktaları anlatınız.
- 3.16.31)** Kurulum testleri neden önemlidir? Sistem testinden farkı nedir?
- 3.16.32)** Sistem teslimi neleri içerir? Tek müşteriye teslim ile çok sayıda müşteriye teslim için yapılması gerekenleri karşılaştırınız.
- 3.16.33)** Sistem ne zaman bakım aşamasına geçer? Donanım ve yazılım bakımlarını karşılaştırınız; benzeyen ve benzemeyen yanlarını belirtiniz.
- 3.16.34)** Bitirme ödevi amacıyla geliştirilen bir yazılım projesi için hazırlanması gereken bir belge kümesi oluşturunuz.

Yazılım geliştirme amaçlı olarak bilgisayarlardan yararlanmak üzere yönelik, çözümleme, tasarım, bakım ve destek araçları bulunmaktadır. Son zamanlarda artan araç kullanım eğilimi ile tasarımdan teste kadar tüm süreçlerde otomasyon sağlayabilecek araçlar gelişmektedir.

4.7. Sorular

- 4.7.1) Yazılımın günümüzdeki anlamını ve içeriğini dikkate alarak tanımı yapınız.
- 4.7.2) Yazılımın karmaşaklılığı ne demektir? Teknik ve yönetsel karmaşaklılığını vererek açıklayınız.
- 4.7.3) Yazılımın yöneticiler ve uygulayıcılar tarafından nasıl algılanmakta olduğunu anlatınız.
- 4.7.4) Yazılım ve donanım karşılaştırıldığında ortaya çıkan önemli farklılıklar örneklerle açıklayınız.
- 4.7.5) Yazılımları geliştirme ve üretme şekline göre sınıflandırınız.
- 4.7.6) Yazılım etkinliği ne demektir? Etkin yazılım nasıl geliştirilmeli?
- 4.7.7) Yazılım mühendisi nedir? Bilgisayar mühendisliği ile yazılım mühendisliğini karşılaştırınız.
- 4.7.8) Yazılım yaşam çevrimi verileri ne demektir? Açıklayınız.
- 4.7.9) Yazılım geliştirme sürecinde yer alan temel aşamalarдан geliştirme evresini anlatınız.
- 4.7.10) Klasik çevrim ile yazılım geliştirme yöntemini anlatınız.
- 4.7.11) Çağlayan modelinin özelliklerini, yararlarını ve aksak yönlerini anlatınız.
- 4.7.12) "V" modelini anlatınız, yarar ve sakıncalarını tartışınız.
- 4.7.13) Spiral geliştirme modelinde kullanılan aşamaları örnek bir proje ile açıklayınız.
- 4.7.14) Prototipleme ile spiral modeli karşılaştırınız. Bir Internet sitesi uygulaması için hangi yöntemi tercih edersiniz?
- 4.7.15) Evrimsel prototipleme ile artımlı geliştirme yöntemlerini karşılaştırınız.
- 4.7.16) Yeni geliştirme tekniklerini özetleyiniz. Herbirini klasik çevrim ile karşılaştırınız. Başka geliştirme teknikleri üzerinde tartışınız.
- 4.7.17) Yazılım mühendisliği araçlarını gruplayarak anlatınız.
- 4.7.18) Yazılım geliştirmede araç kullanımı neden önemlidir?

4.7.19) Yazılım geliştirmeye araçları nasıl seçilmelidir?

4.7.20) Çok sayıda altsistemi tımlaştıracak bir sistem geliştireceksiniz. Altsistem için bir grafik kullanıcı arayüzü ve bir de veri akışını denetleyen yazılım birimi kullanılacaktır. Bu sistem için ne gibi araçlar kullanılabılır? Üretimi için nasıl bir araç geliştirilebilir?

4.7.21) Kod üretici bir araç kullanmakla var olan örnek kodu kullanarak yazılım birimi oluşturmak arasında ne gibi fark vardır?

4.7.22) Grafiksel kullanıcı arayüz yazılımı geliştirmek için ne tür bir kullanabilir?

- **Ben** interler üzerinde **yazılım** geliştirmek istedilerde, **sonra da onları anlatıp, onların görüş bireliği olmaması** önce istedilerde, **bir interin tasarımcı, kodlayıcı gerçekleştirmesinde farklılıklar yaratabilir**. Bir interin tasarımcı, kodlayıcı gerçekleştirmesinde yorumlanmasının bayanızlığa daveti, test edici tarafından farklı şekilde kullanıcı için son derece zorluğu. Bazi interler gerçekleştirmeye zorluğu. Ancak önemli ya da az önemli, bazları ise fazla önemli olmamayı bilirler. Interlerin gerçekleştirmeye zorluğu ise kullanıcıların inter varyansı mak zorundadır. Interlerin varyansı bir interi yenine getirebilmek için anlatılmasının gereken bir durumdu. Bazen basit bir interin varyansı mak çok fazla emek harcaması gerekebilir. Eğer harcamamak gereken bir varyant bulmak için zamanı geçiren kişiye göre, varyant bulmak çok zaman alır.

Cıktılar

- *İsterlerin yazılım geliştirenlər üçün yeni olmayı: Sistemi geliştiren her zaman insanlardır. Orun yeteri kadar bilgili ve deneyimli olması hayatı için gereklidir. Geliştiricilər bənzər isterlərle daha önce Karşılaşmışlardır ve teknik Əzimləri hazırlamışlarla fəzla sorun yoktur. Ancak, uygulama alanındaki isterlər geliştirici üçün çox yeni işe ortaya çıxan deneyim eksikliyi proje başparmağına nəzarət etməyi tələbatlıdır.*

İsterlerin sayı ve kapasının çok fazla olması: İsterlerin sayısının ve kapasının çok fazla olduğu büyük ölçekli projeleri (ister sayısının binlerle ifade edildiği) gerçekleştirmek her zaman büyük risk taşıır. İşin karmaşaklısı, sistemin hâşerimi, artan personel hayatı, geliştirme etkinliklerinin planlanması ve zamanlama zorunluluğuyla birlikte işin karmaşaklılığı ortaya çıkarır.

510 Özet

Yazılım geliştirme sürecinin başlangıcında yazılım ısterlerinin çözümlemesi getirilen bu aşamada, müşterinin yazılımdan bekledikleri belirlenir, gereksinimler açıkly kavuçturulur, yazılım ısterleri modellenir ve tanımlanarak sonraki aşamalar için temel oluşturur. Sistem İsterleri Belirtimi kullanılarak yazılım açısından problem anlayılması ve çözümlemesi sağlanır, Yazılım İsterleri Belirtimi kullanılarak özetlenerek geçirilir.

Yazılım isterleri uygulama alanı, kullanıcı ve işlevsel isterler olarak toplanır, nitelik etmenleri ile birleştirilir, tanımlanır, sınıflandırılır ve belgelendirilir. Çözümlemek sistematik bir şekilde yapılır. Yaygın olarak kullanılan çözümleme yöntemleri yapsal çözümleme, nesneye yönelik çözümleme, veri yapularma yönelik çözümleme ve problem çözümleri, isterleri Belirtimi bu aşamada ortaya çıkan en önemli belgesidir.

6.11. Sorular

- 5.11.1)** Yazılım isterleri çözümlemesi ne demektir?
 - 5.11.2)** Çözümleme ayamasında yapılan çalıymaları anlatınız.
 - 5.11.3)** Isterler neden değişirler?

İsterlerin çok sık değiştiği alanlarda nasıl bir geliştirme yöntemi izlenmelidir?

5.11.4) İster belirlenmede hangi düzeyler kullanılır? Bu düzeylerin sonucunda hangi belge oluşturulur?

5.11.5) İsterler belirleniken izlenebilecek yöntemleri anlatınız.

5.11.6) Test edilemeyecek isteklere örnek veriniz.

5.11.7) İsterleri tanımlamak için ne gibi teknikler kullanılır?

5.11.8) İsterler hangi aşamaya kadar izlenmeli?

5.11.9) Belirtim hazırlanırken dikkate alınması gereken sistem özellikleri nelerdir?

Bu özelliklere neden gerek duyulur?

5.11.10) Problem çözümlemeye kullanılan temel ilkeler nelerdir?

5.11.11) Başta gelen çözümleme yarar ve sakincalarıyla beraber anlatınız.

5.11.12) Yapısal çözümleme yönteminin yarar ve sakincalarıyla açılayınız.

5.11.13) Veri akış diyagramında yer alan temel simgeleri açıklayınız.

5.11.14) Bir akvaryumun otomatik yemleme, ısı kontrol ve su değişimine düzenğini tasarlamanız ve veri akış modelini çiziniz.

5.11.15) Durum geçiş diyagramı ne için kullanılır?

5.11.16) Bir çamaşır makinesi için olası durumları içeren bir durum geçiş diyagramı oluşturunuz.

5.11.17) Varlık ilişkisi diyagramı ne için kullanılır?

5.11.18) Bir üniversitedeki öğretim üyeleri, öğrencileri ve tez çalışmalarını öz nitelikleri ve aralarındaki ilişkileri tanımlayarak varlık ilişkisi diyagramı ile modelleyiniz.

5.11.19) Nesnelerin ana yapısını anlatınız. Bir nesnenin belirtimi neleri içermelidir?

5.11.20) Bir sözcük işlemci tasarlamanak için iş tanımı yazınız. Bu tanımın içindeki nesneleri tanımlayınız.

5.11.21) Çok büyük ve karmaşık sistemler için nesneye yönelik yaklaşım uygun mudur? Yarar ve zararlarını tartışınız.

5.11.22) Gerçek zamanlı sistemlerin çözümlenmesinde yapsal ve nesneye yönelik çözümleme yaklaşımının yarar ve sakincalarını karşılaştırınız.

- 5.11.25)** Veri yapılarına yönelik çözümlemeye örnek olarak basit bir sistem alınız ve çözümleme yapınız.
- 5.11.26)** Nesneye yönelik ile yapsal çözümlemeyi bir arada kullanmak mümkün olabilir mi? Tartışınız.
- 5.11.27)** Prototipleme ne zaman kullanılır? Ne tür örnekler oluşturulur?
- 5.11.28)** Yazılım İsterleri Belirtimi belgesinde neler bulunması gereklidir?
- 5.11.29)** Yazılım isterlerinin müsteri tarafından gözden geçirilmesindeki yanlış nedir?
- 5.11.30)** Çözümleme aşamasında ne gibi yardımcı araçlar kullanılabilir?
- 5.11.31)** İsterler çözümleme aşamasında karşılaşabilecek riskleri anlatınız.

6.15. Özet

Tasarımın en önemli üç ilkesi soyutlama, bilgi gizleme ve kapsamadır. Tasarım, ön tasarım ve ayrıntılı tasarım olmak üzere iki aşamada yapılabilir. Yazılım tasarımını, veri, mimari, yordamsal tasarım ve arayüz tasarımından oluşturur. En önemli tasarım yöntemlerinden ikisi nesneye yönelik tasarım ve veriye yönelik tasarımdır. Bunların yanında tasarım kalıpları önemli yarar sağlar. Etkileşimli sistemlerde kullanıcı arayüzü tasarımı uygun mimaride ve belirli niteliklere sahip olacak şekilde yapılır. Gerçek zamanlı sistemlerin yapısal ve dinamik özellikleri, donanım, işletim sistemi, ara kat tasarımları yapılır. Veri tabanı uygulamaları da uygun bir mimari, arayüz ve veri gereksinimleri dikkate alınarak tasarlanırlar. Internet uygulamaları için amaç, arayüz ve güvenlik tasarımda ön planda tutulur.

Tasarımda genel olarak modülerlik ve tekrar kullanım dikkate alınır, denetim ve işlev ayrı tutularak temel ilkeler gözetilir. Yazılım Tasarım Tanımlaması, tasarım aşaması sonunda ortaya çıkan en önemli belgedir.

6.16. Sorular

- 6.16.1) Yazılım tasarımını ne demektir, açıklayınız.
- 6.16.2) Tasarımda göz önünde bulundurulması gereken ilkeler nelerdir?
- 6.16.3) Yazılım tasarım sürecinin temel aşamalarını ve her aşamada yapılan işleri sıralayınız.
- 6.16.4) Yazılım mimarisi nedir, açıklayınız. Seçimi nasıl yapılır?
- 6.16.5) Yordamsal tasarım yöntemlerini anlatınız.
- 6.16.6) Bileşenler arası arayüz tasarımı nasıl yapılır?
- 6.16.7) Altsistemleri tümleştirmek üzere kullanabilecek bir yazılım mimarisini anlatınız. Benzer şekilde kendiniz bir mimari tasarlayınız.
- 6.16.8) Yazılım tasarım yöntemlerinden birkaçını ana özellikleriyle karşılaştırınız.
- 6.16.9) Veri akışına yönelik tasarım neden yapısal tasarım olarak da adlandırılır? Bu yöntemi veriye yönelik tasarımla karşılaştırınız.
- 6.16.10) Veri akışına yönelik tasarımın ana aşamalarını anlatınız.
- 6.16.11) Nesneye yönelik tasarımın ana fikrini ve temel özelliklerini açıklayınız.
- 6.16.12) Nesneye yönelik tasarım aşamalarını anlatınız.
- 6.16.13) Nesneye yönelik tasarımda programlama dilinin önemini açıklayınız.

- 6.16.14)** Nesneye yönelik tasarım ile veri akışına yönelik tasarımını bir arada kullanmak mümkün olabilir mi? Tartışınız.
- 6.16.15)** Tasarım kalıpları nedir, ne amaçla kullanılır?
- 6.16.16)** Tasarım kalıplarına örnekler vererek açıklayınız.
- 6.16.17)** Etkileşimli yazılımın yapısını anlatınız.
- 6.16.18)** Kullanıcı arayüz yazılımı mimarisini açıklayınız.
- 6.16.19)** Kullanıcı arayüz tasarımında dikkate alınması gerekliliği önemli özellikler nelerdir?
- 6.16.20)** Gerçek zamanlı sistemlerin yapısal özellikleri nelerdir?
- 6.16.21)** Gerçek zamanlı sistemlerin tasarımında süreç yönetiminin önemi nedir?
- 6.16.22)** Veri tabanı yönetim sistemi mimarisini ana bileşenleriyle anlatınız.
- 6.16.23)** Merkezi ve dağıtık veri tabanı sistemlerini karşılaştırınız.
- 6.16.24)** Veri tabanı yönetim sistemi tasarım sürecinde yer alan önemli adımları anlatınız.
- 6.16.25)** İnternet yazılımlarının ana hedefi ne olmalıdır?
- 6.16.26)** Modüler tasarımın yararlarını ve tasarım ilkelerini anlatınız.
- 6.16.27)** Bir tasarımı en iyi anlatan belgelendirme nasıl olabilir?
- 6.16.28)** Yazılım tasarımı sırasında karşılaşılabilecek önemli riskler nelerdir?

bağlayıcı ile bağlanarak yürütülebilir kod elde edilir. Yorumlayıcı kaynak kodun satırlarını birer birer okuyarak işlemci komutları haline dönüştürür ve bu komutları uygular. Yazılım geliştirmek için ilk sahip olunması gereken, uygun bir geliştirme ortamıdır. Dağıtık mimariler için özel yürütme ortamlarına gereksinim vardır. İyi bir yazılım elde edebilmek için kodlama sırasında birçok kurala uyulması gereklidir. Bu kuralların ilke olarak benimsenmesi daha nitelikli yazılım geliştirilmesine yardımcı olur. Kodlama sırasında kaynak kod dosyaları içine açıklayıcı bilgiler yerleştirerek belgeleme yapılır. Gerekirse başka açıklayıcı bilgiler ayrı belge dosyaları halinde düzenleşim sisteminde saklanmalıdır.

7.9. Sorular

- 7.9.1) Yazılım gerçekleştirmi neden önemlidir? Çok iyi tasarılanmış bir yazılımın gerçekleştirim aşaması hangi nedenlerle sorun kaynağı olabilir?
- 7.9.2) Programlama dillerinin gelişimini özetleyiniz.
- 7.9.3) Özel amaçlı dillerle genel amaçlı dilleri karşılaştırınız.
- 7.9.4) Programlama dillerini karşılaştırmak üzere kullanılan genel özellikleri anlatınız.
- 7.9.5) Nesneye yönelik programlama dillerinin ortak özelliklerini açıklayınız.
- 7.9.6) Gerçek zamanlı diller neden diğer dillerden ayrı tutulurlar?
- 7.9.7) Milisaniye düzeyinde zamana bağlı olarak bir donanımdan veri okuyan bir sistem için çeşitli dillerde bir döngü kodu yazınız.
- 7.9.8) Bir geliştirme projesinde programlama dili seçimi nasıl yapılır?
- 7.9.9) Derleyicilerin ana işlevleri nelerdir? Yorumlayıcılardan farkı nedir?
- 7.9.10) İyi bir derleyicide ne gibi özellikler aranmalıdır?
- 7.9.11) Bir geliştirme ortamında bulunması gerekenleri anlatınız.
- 7.9.12) Geliştirme ortamı ile test ortamının aynı bilgisayarda kurulu olması neleri etkileyebilir? Olumlu ve olumsuz yanları için örnekler veriniz.
- 7.9.13) Yürütme ortamında ne gibi mimariler kullanılmaktadır?
- 7.9.14) Birden fazla bilgisayar üzerinde çalışan bir uygulama geliştirmek için nasıl bir altyapıya gereksinim duyulur?
- 7.9.15) Kodlamada kullanılacak dili (İngilizce ya da Türkçe) yarar ve sakıncalarıyla tartışınız.
- 7.9.16) Kaynak kod içinde yapılan belgelemeyi anlatınız.

- 7.9.17) Kodlama sırasında kullanılan veri bildirimini için önemli ~~kuralları~~ açıklayınız.
- 7.9.18) Kodlamada etkinlik neden önemlidir? Nasıl sağlanır?
- 7.9.19) Aykırı durumların kotarılması nasıl yapılır? Neden önemlidir?
- 7.9.20) Bir yazılımın hiç çökmemesi ne demektir?
- 7.9.21) Kodlamanın niteliksel özelliklerini sıralayınız.
- 7.9.22) Modüler yapıda kaynak kod dosyası nasıl oluşturulur?
- 7.9.23) Program yazmak ile yazılım geliştirmek arasında ne fark vardır?
- 7.9.24) Kodlamada kullanmak üzere seçtiğiniz bir dil için kod yazım ~~kılavuzu~~ hazırlayınız.
- 7.9.25) Sizin için uygun olabilecek bir geliştirme ortamı dizin yapısı tasarlayınız.
- 7.9.26) Kodlama sonrası ne gibi belgelendirme yapılabilir?
- 7.9.27) Tasarımın tamamen koda dönüştürülememesi durumunda ne gibi riskler oluşur?
- 7.9.28) Kod gözden geçirtmesinde nelere dikkat etmelisiniz?

8.8. Sorular

- 8.8.1) Yazılım testinin önemini ve gerçekleştirmeye süreci içindeki yerini anlatınız.
- 8.8.2) Yazılım testi hangi amaçla yapılır?
- 8.8.3) Testler nasıl yapılır? Kısaca anlatınız.
- 8.8.4) En yaygın test yöntemlerini açıklayınız.
- 8.8.5) Bir tren garına yerleştirilecek olan demiryolu trafik kontrol sisteminin nasıl test edilebileceğini anlatınız.
- 8.8.6) Otomatik test gerçekleştirebilen yardımcı araçlar nelerdir?
- 8.8.7) "V Modeli"nin ne gibi yararları vardır? Sakıncaları nasıl giderilebilir?
- 8.8.8) Birim testi nedir? Nasıl yapılır?
- 8.8.9) Tümleştirme nedir? Kaç çeşittir?
- 8.8.10) Yeterlilik testi ne demektir? Nasıl yapılır? Neden önemlidir?
- 8.8.11) Doğrulama ve geçerleme arasındaki farkı anlatınız.
- 8.8.12) Doğrulama sürecinde yapılan işleri açıklayınız ve bir örnek veriniz.
- 8.8.13) Geçerleme işlemini anlatınız ve bir örnek veriniz.
- 8.8.14) Sistem testi nedir? Çeşitlerini örnekler vererek açıklayınız.
- 8.8.15) Kabul testi nedir? Çeşitlerini örneklerle açıklayınız.
- 8.8.16) Sistem kullanım yerine monte edilirken ne tür testler yapılır?
- 8.8.17) Küçük bir sistem tasarlayarak üzerinde rasgele yapılabilecek testleri anlatınız.
- 8.8.18) Testler için kabul kılavuzlarını anlatınız.
- 8.8.19) Profesyonel bir geliştirme projesinde kabule esas testlerin neye göre yapılacağı ne zaman belirlenmelidir?
- 8.8.20) Gerçek zamanlı bir sisteme örnek vererek nasıl test edilebileceğini anlatınız.
- 8.8.21) Neden bir test yönetimi gereklidir? Testler kimler tarafından yapılmalıdır?
- 8.8.22) Hata ayıklama nedir? Ne zaman başlar, nasıl bir süreçle yapılır?
- 8.8.23) Başlıca hata ayıklama yöntemleri nelerdir?
- 8.8.24) Testlerde ne gibi belgeler kullanılır? Anlatınız.
- 8.8.25) Test aşamasında karşılaşılabilecek önemli riskleri belirtiniz.

- Yazılımın bakımı:*
- *Geliştirme ve test ortamının bulunması:* Yeni bir yazılım veya sistem için uygun bir geliştirme ve test ortamının oluşturulması gereklidir. Bunu sağlamak için ayrı bir yatırım gerekebilir.
 - *Personel devamlılığı:* Yazılımı geliştiren kişilerin daha sonra ayrılması durumunda deneyim sahibi personel kalmayabilir. Eğer yeterli belgelendirme yoksa bir değişiklik için çok zaman harcanabilir.
 - *Personel deneyimi:* Yeni personelin bakım aşamasındaki bir yazılıma gelen değişikliği uygulaması için önemli bir öğrenme süreci gerekebilir.
 - *Eski teknolojinin getirebileceği zorluklar:* Bakım aşamasında üzerinde değişiklik yapılması gereken yazılımın kullandığı teknolojinin eski olması durumunda (örneğin Assembly ile geliştirilmiş bir yazılıma yeni bir işlev eklenmesi), yeni isterlerin uygulanması tahmin edildiğinden daha güç olabilir.
 - *Müşteri hoşnutsuzluğu:* Üzerinde değişiklik yapılan yazılımın çeşitli kısımlarının etkilenmesi nedeniyle müşteri hoşnutsuzluğu ortaya çıkabilir.
 - *Eski yazılıma değişiklik uygulamanın güçlüğü:* Eski bir yazılıma değişiklik uygulamaya çalışmak yeni baştan geliştirmekten daha pahalıya mal olabilir.
 - *Eski yazılımı yeniden geliştirme arzusu:* Eski bir yazılımda değişiklik yapmak tansa yeni baştan geliştirmek için arzu duyulabilir. Dikkatle değerlendirilmeli takdirde, gereğinden fazla maliyet doğurur.

9.6. Özet

Bilgisayar tabanlı sistemlerin geliştirilip kullanıcıya tesliminden sonra bakım aşaması başlar. Yazılım bakımı düzeltici, uyarlayıcı, iyileştirici ve önleyici bakım olarak uygulanır. Bakım amacıyla bir Değişiklik Kontrol Kurulu oluşturulur. Herhangi bir değişiklik, isterler çözümlemesi, tasarım, gerçekleştirim, test ve teslim aşamalarından geçilerek uygulanır. Değişiklik istekleri kullanıcıdan veya geliştiriciden gelebilir. Yazılım, bakım kolaylığı sağlayacak şekilde geliştirilmeli, bakım sırasında niteliksel özellikler korunmalı ve niceliksel ölçümler yapılmalıdır.

Bakımdan kodlamanın, verilerin, belgelendirmenin ve başarının etkilenmesi olasıdır. Belgelendirilmemiş yazılımlar için bakım gereksinimi oluştduğunda, aynı kod üzerinde bakım yapılabilir, kod başka ortama taşınabilir, tersine mühendislik veya yeniden yapılanma uygulanabilir.

9.7. Sorular

9.7.1) Yazılım bakımı ne demektir?

9.7.2) Yazılım bakım çeşitleri nelerdir? Bunları bir örnek üzerinde açıklayınız.

- 9.7.3) Bakım örgütlenmesi nasıl oluşturulmalıdır?
- 9.7.4) Bakım sürecinde yapılması gereken işleri tanımlayınız.
- 9.7.5) Bakım sürecinin normal süreçten farklılıklarını nelerdir?
- 9.7.6) Bakım aşamasında değişiklikler nasıl önerilir?
- 9.7.7) Bakım kolaylığı sağlayan en önemli etmenler nelerdir?
- 9.7.8) Bakım aşamasının ana sorunlarını anlatınız.
- 9.7.9) Bir Internet sitesi yazılımının bakımı neleri içerebilir? Örnekle anlatınız.
- 9.7.10) Bakım projeleri için maliyet kestirimi nasıl yapılabilir?
- 9.7.11) Bir şirketin finans yazılımını geliştiren firma ile nasıl bir bakım anlaşması yapılabilir? Bu anlaşmanın içeriğini anlatınız.
- 9.7.12) Teslim edilen bir yazılımın bakım sürecinin maliyeti nasıl hesaplanabilir?
- 9.7.13) Belgesi bulunmayan yazılımların bakımı nasıl yapılır?
- 9.7.14) Kod taşıma ne demektir? Ne zaman uygulanır? Yarar ve sakincaları nelerdir?
- 9.7.15) Tersine mühendisliğin amacı nedir? Yeniden mühendislikle karşılaşırız.
- 9.7.16) Bakım aşamasında karşılaşılabilecek önemli riskleri belirtiniz.
- 9.7.17) Eski bir yazılım üzerinde değişiklik yapıp yeni işlevler eklemek yerine onu baştan geliştirmek istediğinizde karşınıza ne gibi sorunlar çıkabilir?

Gözden geçirmeler, yazılım geliştirme sürecinin çeşitli evrelerinde yazılım ürününün yazarından başka kişiler tarafından incelenmesi şeklinde uygulanarak kusurların ortaya çıkarılmasını, bunların uygun şekilde düzeltilmesini ve ürünün daha da iyileştirilmesini sağlarlar. Bu amaçla her geliştirme aşaması için gözden geçirme ve inceleme kontrol listeleri oluşturulur.

Resmi nitelik güvence yöntemleri uygulanarak doğruluğu kanıtlanan yazılımın güvenilirliği ve yararlanılabilirliği de ölçülebilir. Eğer yazılımin geliştiriciden başka bir embla aktarımı gerekiyorsa lisans ve sahiplik durumları dikkate alınarak belirli bir planlama yapılır. Ürün ve süreç niteliklerini sayısal olarak ölçebilmek için çeşitli metrikler ortaya konmuştur. Yazılım metriklerinin amacı, bir yazılımın yaşam çevrimi boyunca yazılım projesinin ve ona ait nitelik gereksinimlerinin karşılanıp karşılanmadığını değerlendirmektir.

10.10. Sorular

10.10.1) Yazılım niteliği ne demektir?

10.10.2) Yazılım nitelik etmenleri nelerdir? Gruplayarak anlatınız.

10.10.3) Yazılım niteliğini ölçmek üzere kullanılabilecek metrikler nelerdir?

10.10.4) Nitelik güvence nedir? Neden gereklidir?

10.10.5) Nitelik güvence etkinliklerini tanımlayınız.

10.10.6) Süreç modeli ne demektir? Nasıl seçilir?

10.10.7) Nitelik sistem standartlarına birkaç örnek veriniz.

10.10.8) Gözden geçirmeler nasıl uygulanır?

10.10.9) Resmi nitelik güvence yöntemlerinden başlıcaları nelerdir? Ana özelliklerini anlatınız.

10.10.10) Yazılım güvenilirliği nedir? Nasıl belirlenir? Neden önemlidir?

10.10.11) Yazılım emniyetini açıklayınız. Güvenilirlikle emniyet arasındaki farkı belirtiniz.

10.10.12) Sistem aktarımı ne demektir? Ne zaman yapılır? Neleri kapsar?

10.10.13) Yazılım sözleşmeleri kaç türüdür? Örnekler vererek açıklayınız.

10.10.14) Yazılım lisansı ne demektir? Çeşitleri nelerdir?

10.10.15) Nitelik güvence yönetimi nasıl uygulanır? Bu amaçla, bir yazılım evi için örnek bir örgüt yapısı oluşturunuz.

10.10.16) Yazılım geliştirmede ürün ve süreç niteliği ne demektir?

- 10.10.17)** Yazılım ürün metrikleri ne için kullanılır?
- 10.10.18)** Metriklerle doğrulama ve değerlendirme nasıl yapılır?
- 10.10.19)** Bir Internet sitesi oluşturmada kullanılabilen metrikleri tanımlayınız.
- 10.10.20)** Bir proje yöneticisi ölçme sürecinin önemini personeline nasıl anlatmalıdır?
- 10.10.21)** Yazılımın başarım ve etkinliğini artırmak için nasıl bir yol izlenebilir?

hakki kazandıktan sonra bu dosyaları kendi bilgisayarlarına indirirler. Yine Internet üzerinden veya telefonla ücret ödemesi yaptıktan sonra alacakları etkinleştirme kodlarını kurma sırasında girerek yazılımlarını kullanmaya başlayabilirler. Sonraki sürüm yükseltmeleri veya yamalar da bu sitelerden indirilerek kullanılır.

11.6. Özet

Düzenleşim yönetimi, bir ürünün işlevsel ve fiziksel özelliklerinin denetimini ve gözlemlenebilirliğini sağlamak amacıyla ürüne yaşam çevrimi süresince uygulanan bir yönetim disiplinidir. Yazılım geliştirme sırasında üretilen her türlü kaynak kod, yürütülebilir kod ve belge birer düzenleşim öğesi olarak tanımlanabilir. Düzenleşim yönetimi, genel işlevi kapsamında, tüm proje gereksinimlerini karşılayacak şekilde, her türlü yazılım ve malzeme düzenleşiminin en alt ayrıntı düzeyine kadar tanımlandığından ve kaydedildiğinden emin olarak, düzenleşim yönetimi kapsamına giren ögelere getirilecek değişikliklerin denetim altında tutulmasını ve kaydedilmesini sağlar. Ürün tanımlama ve sürüm denetimi bir ürün gelişme yolu tanımlar. Değişiklik denetimi, bir kurul denetimi altında ve belirli bir süreçe göre yapılır.

Yazılım geliştirme süreci boyunca herhangi bir düzenleşim öğesine sabitleme uygunabilir. İsterlerin, tasarımin, kaynak kodun sabitlenmesi birer aşama noktasıdır. Geliştirilen son ürünün, ara ve test sürümlerinin çoğaltıması, kurulum için gerekli dosyaların eklenmesi, sürümün belgelendirilmesi, kullanıcılara teslimi düzenleşim yönetimi denetiminde yapılır.

11.7. Sorular

- 11.7.1) Düzenleşim yönetimi nedir? Neden gereklidir? Kullanılmadığı durumlarda ne gibi sorunlar oluşabilir?
- 11.7.2) Düzenleşim öğesi nedir? Bir sistem yapısı ile örnek veriniz.
- 11.7.3) Düzenleşim yönetim sisteminin genel özellikleri neler olmalıdır?
- 11.7.4) Bir veri tabanı yönetim sistemi kullanarak basit bir düzenleşim yönetim sistemi tasarlaymentınız.
- 11.7.5) Düzenleşim yönetimi sürecinin temel aşamalarını anlatınız.
- 11.7.6) Ürün gelişme yolu nedir? Anlatınız.
- 11.7.7) Düzenleşim denetimi nasıl yapılır? Ana etkinlikleri nelerdir?
- 11.7.8) Değişiklik uygulanması nasıl bir süreçe göre yapılır?
- 11.7.9) Değişiklik denetim sürecinin temel adımlarını tanımlayınız.
- 11.7.10) İşletim sistemi yeteneklerini kullanarak basit bir düzenleşim sistemi tasarlaymentınız.

- 11.7.11) Düzenleşim denetimi ile geliştirme nasıl yapılır?
- 11.7.12) Sabitleme nedir? Nelere uygulanır?
- 11.7.13) Sabitlemenin neden gerekli olduğunu kendi ifadelerinizle anlatınız.
- 11.7.14) Ürün çoğaltma nasıl yapılır? Temel adımlarını anlatınız.
- 11.7.15) Üç kişilik bir yazılım ekibi ile bir tek müşteriye küçük bir proje yapmactasınız. Ne tür sabitlemeler uygulayacağınızı ve bunları nasıl denetim altında tutacağınızı anlatınız.
- 11.7.16) Açık kaynaklardan araştırma yaparak ticari düzenleme yönetim araçlarını araştırınız.

MDA hakkında diğer bilgiler ve *MDA Kanvası* (Haziran 2003) ve OMG resmi Internet sitesinde (<http://www.omg.org>) bulunabilir.

12.5. Özет

Belirli bir işi yapmak üzere birbiriyle bütünsel olarak kullanılan yöntemler kümesi olarak tanımlanan yöntembilimler, hedefledikleri işin başarıyla, zamanında ve en ucuz biçimde yapılmasını sağlamak amacıyla geliştirilmişlerdir. Askeri sistemler ve buna bağlı olarak askeri sistem yazılım geliştirme gereksinimleri birçok alanda teknolojik gelişmelerin yönlendirici, belirleyici, yenileyici ve itici unsurunu oluşturmuş, çeşitli standartların gelişmesine öncülük etmişlerdir. En son standart olan IEEE/EIA 12207 satın alıcılar, satıcılar, geliştiriciler, kullanıcılar ve bakımcılar için süreçler tanımlamakta, yönetim, mühendislik ve veri gereksinimlerini içermektedir.

Yazılım geliştirme sürecinde gereken adımlar, aşama noktaları ve hazırlanacak belgeler seçilecek bir standarda uygun olarak bir geliştirme planı içinde tanımlanır. Projenin büyüklüğüne göre, standartların yeniden biçimlendirilmesi olasıdır.

Model-güdümlü mimari, yazılım geliştirme süreçlerini modele dayandırarak taşınabilir sistem geliştirmeyi hedeflemektedir. Bu amaçla geliştirilmiş yardımcı araçlar bulunmaktadır.

12.6. Sorular

- 12.6.1) Yazılım geliştirme standartlarının oluşmasına askeri sistemler nasıl öncülük etmişlerdir?
- 12.6.2) IEEE/EIA 12207'nin temel süreçleri nelerdir?
- 12.6.3) Standart bir yazılım geliştirme sürecinde yer alan temel aşamaları ve aşama noktalarını tanımlayınız.
- 12.6.4) Yazılım geliştirme ortamı ne zaman ve nasıl kurulur? İdeal bir ortamda neler bulunmalıdır?
- 12.6.5) Yazılım isterleri çözümlemesi aşaması hangi adımlardan oluşur?

- 12.6.6) Yazılım tasarım aşaması hangi adımlardan oluşur?
- 12.6.7) Yazılım gerçekleştirmi neleri içerir?
- 12.6.8) Yazılım testi nasıl yapılır?
- 12.6.9) Geliştirilen yazılım kullanıcıya teslim edilirken neler yapılır, ne tür belgeler hazırlanır?
- 12.6.10) Tek bir bilgisayar üzerinde çalışabilen ve ürün fiyatlarını içeren bir satış kasası yönetim yazılımını bir müşteri için özel olarak geliştirecek ve tüm kaynak kodlarıyla beraber müşteriye teslim edeceksiniz. Hazırlamanız gereken belgeleri tanımlayınız.
- 12.6.11) Küçük bir yazılım geliştirme için gerekli bilgileri toplayan tek bir belge şablonu oluşturunuz.
- 12.6.12) Bir Internet sitesi geliştirme projesi için nasıl bir belgelendirme kullanılabilir?
- 12.6.13) Hiç belge hazırlanmadığı takdirde bir yazılımın kullanımı ve bakımı sırasında karşılaşılabilecek sorunları özetleyiniz.

Üst düzeyde bir yönetim altında ve eşgündüm halinde yürütülmeleri durumunda daha fazla yarar sağlanması amacıyla oluşturulur. Gereksinimlere göre tanımlanarak başlatılan proje, bir yönetim oluşturulur. Proje yönetiminde yer alan başlangıç, planlama, yürütme ve kapanış süreçleri çeşitli bileşenlerden oluşur. Bu yönetişel bileşenlerin ne kadarının nasıl uygulanacağı projenin büyüklüğine bağlıdır.

13.13. Sorular

- 13.13.1) Proje örgütlenmesi ne demektir?**
- 13.13.2) Proje ve program tanımını yapınız. Birer örnek vererek karşılaştırınız.**
- 13.13.3) Proje yönetim süreçleri nelerdir?**
- 13.13.4) Proje tümleştirme yönetimi altında neler bulunur?**
- 13.13.5) Proje kapsam yönetimi neler içerir?**
- 13.13.6) Proje zaman planlaması nasıl yapılır?**
- 13.13.7) Maliyet yönetimi hangi süreçleri kapsar? Teknik yönetim ile mali yönetim birleşik mi olmalıdır?**
- 13.13.8) Gecikmekte olan bir projeye daha fazla personel verilmesi onu daha da geciktirebilir veya erken bitirebilir. Bu durumlar için birer örnek veriniz.**
- 13.13.9) Çok sayıda ve ayrik çalışan gruplarla yazılım geliştirmede ne gibi güçlükler ortaya çıkabilir?**
- 13.13.10) Proje kapanışı nasıl yapılır? Kapanışı yapılmayan proje olabilir mi?**
- 13.13.11) Risk ne demektir? Risk yönetimi nasıl yapılır?**
- 13.13.12) Yazılım projelerinde oluşabilecek riskler nelerdir?**
- 13.13.13) Risk yönetimi yapılmadan yürütülen bir yazılım projesinde ne gibi sorunlar ortaya çıkabilir?**
- 13.13.14) Yazılım edinme seçenekleri nelerdir? Hazır satın almak ya da baştan geliştirmek kararı vermek için ne gibi noktalara dikkat edilmelidir?**
- 13.13.15) Projede kestirimler nasıl yapılır?**
- 13.13.16) Bir okulun öğrenci kayıtlarını tutan bir sistem geliştirmek için gerekli insan gücünü ve süreyi hesaplayın.**
- 13.13.17) Yazılım geliştirme yöneticisinin görev ve sorumlulukları nelerdir?**
- 13.13.18) Yazılım geliştirme ekipleri nasıl oluşturulur?**
- 13.13.19) Araştırma projeleri sizce nasıl yönetilmelidir?**