**1. Hata Ayıklamanın Önemi Nedir?**  
Hata ayıklama, bir arızanın nedenlerini bulma, analiz etme ve düzeltme sürecidir. Bu süreç, arızaların temel nedenlerini belirleyerek aynı hataların tekrarını önlemeyi sağlar. Test sırasında tespit edilen hataların düzeltilmesi, daha kaliteli yazılım ürünlerine ulaşılmasına katkı sağlar.  
**Reference:** Page 19-20​(ISTQB\_CTFL\_Syllabus-v40…).

**2. Neden Yazılım Testine İhtiyaç Duyarız?**  
Yazılım testi, kullanıcıların karşılaşabileceği sorunları önceden tespit ederek yazılım kalitesini artırır. Hataları erken tespit etmek maliyetleri azaltır ve yazılımın düzenleyici gereksinimlere uyumunu sağlar. Ayrıca, kullanıcı ihtiyaçlarını karşılama ve riskleri minimize etme açısından da kritik bir süreçtir.  
**Reference:** Page 19​(ISTQB\_CTFL\_Syllabus-v40…).

**3. Kaliteyi Etkileme Faktörleri Nelerdir?**  
Kalite, test süreci ve hataların etkin bir şekilde yönetimi ile doğrudan ilişkilidir. İyi bir test süreci, yazılımın gereksinimleri karşıladığını ve kullanıcı ihtiyaçlarını yerine getirdiğini doğrular. Kalite güvence (KG) süreçleri ise bu hedeflerin sürekliliğini sağlar.  
**Reference:** Page 20​(ISTQB\_CTFL\_Syllabus-v40…).

**4. Yazılım Testinin Otomatikleştirilmesinin Faydaları ve Dezavantajları Nelerdir?**  
**Faydaları:**

* Tekrarlayan testlerin hızlı ve tutarlı bir şekilde gerçekleştirilmesi.
* Büyük veri kümeleri üzerinde test yapabilme.
* Zaman ve maliyet tasarrufu.

**Dezavantajları:**

* Test otomasyonu için yüksek başlangıç maliyetleri.
* Test senaryolarının düzenli bakım ihtiyacı.
* Bazı karmaşık senaryoların otomatikleştirilememesi.  
  **Reference:** Page 61​(ISTQB\_CTFL\_Syllabus-v40…).

**5. Bir Yazılım Testi Uzmanının Sahip Olması Gereken Özellikler Nelerdir?**

* Eleştirel ve analitik düşünme becerisi.
* Teknik bilgi ve tecrübe.
* Detaylara dikkat etme ve problem çözme yeteneği.
* İyi bir iletişim becerisi ve takım çalışmasına yatkınlık.  
  **Reference:** Page 24​(ISTQB\_CTFL\_Syllabus-v40…).

**6. Yazılım Testinin Türleri Nelerdir?**

* İşlevsel test.
* Performans testi.
* Güvenlik testi.
* Kullanılabilirlik testi.
* Uyum testi.  
  **Reference:** Page 31​(ISTQB\_CTFL\_Syllabus-v40…).

**7. Test Çeşitleri Nelerdir (Seviyelere Göre)?**

* Birim testi.
* Entegrasyon testi.
* Sistem testi.
* Kabul testi.  
  **Reference:** Page 30​(ISTQB\_CTFL\_Syllabus-v40…).

**8. Statik Test Nasıl Yapılır ve Hangi Araçlar Kullanılabilir?**  
Statik test, yazılımın çalıştırılmadan incelenmesini içerir. Kod incelemeleri, gözden geçirme toplantıları ve statik analiz araçları bu testlerde kullanılır. Popüler araçlar arasında SonarQube ve Checkstyle bulunur.  
**Reference:** Page 35-36​(ISTQB\_CTFL\_Syllabus-v40…).

**9. Dinamik Testin Prensiplerine Ayrıntılı Örnek Verin.**  
Dinamik test, yazılımı çalıştırarak hataları bulmayı amaçlar. Örneğin, giriş değerlerinin belirli bir aralıkta olup olmadığını kontrol eden sınır değeri analizi bu tür testlere örnek olabilir.  
**Reference:** Page 42​(ISTQB\_CTFL\_Syllabus-v40…).

**10. Dinamik Test ile Statik Testi Karşılaştırınız.**

* **Statik Test:** Yazılım çalıştırılmadan gerçekleştirilir, gözden geçirme ve analiz içerir.
* **Dinamik Test:** Yazılım çalıştırılır, sistemin gerçek davranışı incelenir.  
  **Reference:** Page 36​(ISTQB\_CTFL\_Syllabus-v40…).

**11. Yazılım Testlerinde Kullanılan Test Araçları Nelerdir?**

* **Test Yönetimi Araçları:** JIRA, TestRail.
* **Test Otomasyon Araçları:** Selenium, Appium.
* **Performans Test Araçları:** JMeter, LoadRunner.  
  **Reference:** Page 60-61​(ISTQB\_CTFL\_Syllabus-v40…).