|  |  |  |
| --- | --- | --- |
| Yazılım Testi Nedir?  • İşlevsel ve otomasyon araçlarını kullanarak yazılım hatalarını bulmak amacıyla bir programı veya uygulamayı yürütme süreci  • Bir yazılım programını/uygulamayı doğrulama/doğrulama süreci  • Testçiler, testten geçmek için değil, kırılmak için test etmelidir. | Yazılım testi nedir?  -Manuel veya otomasyon araçlarını kullanarak bir yazılımdaki istenmeyen davranışları bulmak amacıyla yapılan operasyonlar sürecidir. | What is software testing?  It is the process of operations performed to find unwanted behavior in a software using manual or automation tools. |
| 2. Yazılım Gereksinimleri Spesifikasyonu nedir? • Bir yazılım gereksinimleri belirtimi, müşteri ile tedarikçi arasında bir sözleşme işlevi gören bir belgedir. • Bu SRS, son kullanıcının söz konusu uygulamaya ilişkin tüm gereksinimlerini içerir. SRS, müşteri ile tedarikçi arasında bir iletişim ortamı olarak kullanılabilir. • Geliştirici ve test eden kişi, uygulamayı SRS'de yazılı gerekliliklere göre hazırlar ve inceler. • Dokümante edilen SRS, İş Analisti tarafından müşteri için tüm gereksinimler dikkate alınarak hazırlanır. | Yazılım gereksinimleri sertifikasyonu nedir?  -Müşteri ile tedarikçi arasında sözleşme görevi gören bir belgedir.  SRS belgesi  - son kullanıcının tüm gereksinimlerini içerir  - ayrıca bir iletişim belgesidir | What is software requirements certification?  It is a document that acts as a contract between the customer and the supplier.  SRS certificate  - contains all the requirements of the end user  - is also a communication document |
| 3. Yazılım Geliştirme Yaşam Döngüsü (SDLC) - SDLC nedir? • SDLC, yazılım veya uygulama oluşturma aşamalarını tanımlar. o Proje Planlama o Gereksinim Toplama (Projeyi planlamak için kullanılan bilgilerin toplanması, Risklerin belirlenmesi) o Tasarım (Uygulamanın nasıl oluşturulacağı) o Kodlama (geliştirme) (Gereksinimlere göre, geliştiriciler uygulamayı yazacaktır) o Test o Üretim (dağıtım) (Ürünün piyasaya sürülmesi) o Bakım (Ürünün kararlı olduğundan emin olmak, hatalar hakkında müşteri raporuna bakmak ve düzeltmek) | Yazılım geliştirme yaşam döngüsü(SDLC) nedir?  -SDLC yazılım veya uygulama aşamalarını kapsar. Bunlar  - Planlama  - Gereksinim Toplama  - Tasarım  - Kodlama-Geliştirme  - Test  - Dağıtım  - Bakım  aşamarını kapsar | What is software development lifecycle (SDLC)?  -SDLC covers software or application phases. These  - Planning  - Requirements Gathering  - Design  - Coding-Development  - Test  - Distribution  - Care  covers the stages |
| **4. Yazılım Testi Yaşam Döngüsü (STLC) - STLC nedir? • STLC, yazılım veya uygulamanın test edilmesindeki aşamaları tanımlar. STLC sürecinde, ürün kalitesini artırmak için farklı faaliyetler yürütülür.**  **o Gereksinim analizi**  **o Test Planlama**  **o Test Tasarımı**  **o Test Ortamı Kurulumu**  **o Test Yürütme**  **o Test Raporlama** | Yazılım Testi Yaşam Döngüsü(STLC) nedir?  -STLC yazılım ve uygulamanın test edilmesindeki aşamalarını kapsar. Bu aşamalar  -Gereksinim analizi  -Test Planlama  -Test Tasarımı  -Test Ortamı kurulumu  -Test Yürütme  -Test Raporlama | What is Software Testing Lifecycle (STLC)?  -STLC covers the stages of software and application testing. These stages  -Requirements analysis  -Test Planning  -Test Design  -Test Environment setup  -Test Execution  -Test Reporting |
| 5. STLC ve SDLC arasındaki fark nedir? • STLC, SDLC'nin bir parçasıdır. STLC'nin SDLC kümesinin bir alt kümesi olduğu söylenebilir. • STLC, yazılım veya ürün kalitesinin garanti edildiği test aşamasıyla sınırlıdır. SDLC, bir yazılım veya ürünün tam olarak geliştirilmesinde çok geniş ve hayati bir role sahiptir. • Ancak STLC, SDLC'nin çok önemli bir aşamasıdır ve nihai ürün veya yazılım, STLC sürecinden geçmeden piyasaya sürülemez. • STLC aynı zamanda yayın sonrası/güncelleme döngüsünün, bilinen kusurların giderildiği veya yazılıma yeni bir işlevsellik eklendiği SDLC'nin bakım aşamasının bir parçasıdır. | STLC ile SDLC arasındaki fark nedir?  -STLC, SDLC nin bir alt aşamasındır.  -STLC yazılımın kalitesinin garanti edildiği test aşamasıdır  -SDLC, yazılım geliştirmenin tüm aşamalarını kapsar.  -Ayrıca STLC, yazılımın bakım aşamasının bir parçasıdır | What is the difference between STLC and SDLC?  -STLC is a sub-stage of SDLC.  -STLC is the testing phase where the quality of the software is guaranteed  -SDLC covers all phases of software development.  -Also STLC is part of the maintenance phase of the software |
| 6. Gereklilik nedir? • Gereksinimler, kullanıcıların yazılım veya ürünle ilgili beklentilerini iletir. • Müşteriden gereksinimleri toplama, analiz etme ve belgeleme süreci, gereksinim mühendisliği olarak bilinir. • Gereksinim mühendisliğinin amacı, gelişmiş ve açıklayıcı SRS "Sistem Gereksinimleri Spesifikasyonu" Belgesini geliştirmek ve sürdürmektir. | Gereklilik - gereksinimler nedir?  -Kullanıcıların yazılım veya ürünle ilgili tüm beklentilerini içerir.  -Gereksinimler SRS olarak bilinen belge ile belirtilir  -SRS belgesini Gereksinim mühendisleri hazırlar  -Test aşaması SRS belgesindeki gereksinimler üzerine yapılır | Requirement - what are the requirements?  It includes all the expectations of the users about the software or product.  -Requirements are specified by document known as SRS  - Requirements engineers prepare the SRS document  -The testing phase is done on the requirements in the SRS document |
| 7. Gereksinim nereden geliyor? • Müşteriler uygulama için gereksinimleri belirtir • Son kullanıcılarla konuşun - bu uygulamayı en çok kullanacak kişiyle konuşun • İş Ortaklarıyla konuşun – • Etki Alanı Uzmanlarıyla konuşun – bu uygulamayı daha önce benzer şekilde oluşturmuş olan kodlayıcılar ve geliştiriciler veya uzmandır üretilen ürünün türü • Sektör Analistleri ve rakipler hakkında bilgiler | Gereksinim nereden geliyor?  -Müşterilerden toplananlar  -Son kullanıcılardan toplananlar  -iş ortaklarından toplananlar  -Etki alanı uzmanlarından toplananlar  -Sektör analistlerinin fikirleri  -Rakip uygulamalardan alınan veriler | Where does the requirement come from?  - Collected from customers  - Gathered from end users  - collected from business partners  - Gathered from domain experts  -Ideas of industry analysts  -Data from competing apps |
| 8. Test ne zaman başlar? • Test, gereksinimlerin test edilmesiyle başlar (en olası cevap gibi görünen kodlama aşamasından sonra değil.) • İlk etapta gereksinimin doğru olduğundan emin olmalıyız. Yanlış gereksinim ile hatasız uygulama oluşturmak imkansızdır. | Test Nezaman Başlar?  -Yapılan ilk test toplanan gereksinimleri test etmektir.  -Yanlış gereksinimler kaliteli ürün üretemezsiniz | When Does the Test Start?  -The first test done is to test the collected requirements.  -Wrong requirements can not produce quality product |
| **9. Gereksinimin iyi mi yoksa kötü mü olduğu nasıl anlaşılır?**  **• Gereksinim (SMART) olmalıdır**  **- Belirli >Kullanıcı oturum açabilmelidir. Geçerli kullanıcı adı ve şifreye sahip yetkili kullanıcı oturum açabilmelidir.**  **- Ölçülebilir > Kullanıcı çok hızlı oturum açabilmelidir (giriş düğmesine tıkladıktan sonra 2 saniye içinde).**  **- Ulaşılabilir**  **- Gerçekçi**  **- Test edilebilir > Kullanıcı, makbuzu çok hızlı indirebilmelidir (2 saniyede** | Gereksinimin iyi mi kötü mü olduğu nasul anlaşılır?  -Gereksinim (SMART)  -Belirli  -Ölçülebilir  -Ulaşılabilir  -Gerçekçi  -Test Edilebilir  Olmalıdır. | How do you know if the requirement is good or bad?  -Requirement must be (SMART)  -Clear  -Measurable  -Accessible  -Realistic  -Testable |
| 10. Neden test ediyoruz? • Hatasız uygulama oluşturmak için. • Memnun son kullanıcı ve müşteri için. • Daha fazla gelir elde etmek için harika bir ürün oluşturmak. • Test etmeyi seviyorum ve test etmek benim tutkum. | Neden Test ediyoruz?  -mükemmel bir ürün oluşturmak için  -Memnun müşteriler ve son kullanıcılar için  -Kar oranını ve geliri artırmak için | Why We Test  -to create a perfect product  -For satisfied customers and end users  -To increase profit rate and revenue |
| 11. Testi yapanın ana sorumluluğu nedir? • Hatayı olabildiğince erken bulmak. Hataların çoğunun giderildiğinden emin olun. • Son kullanıcıyı ve müşteriyi memnun etmek için hatasız ve kullanıcı dostu bir uygulama satın alın. | Test yapmanın ana sorumluluğu nedir?  - Olabilecek hatayı olabildiğince erken bulmak  - Mümkün olduğunca hatasız bir uygulama üretmek  - Müşteri memnuniyetini geliştirmek  - Kullanıcı dostu bir uygulama üretmek | What is the main responsibility of testing?  - Finding possible fault as early as possible  - Producing an application that is as error-free as possible  - Improving customer satisfaction  - Producing a user-friendly application |
| 12. %100 test mümkün mü? • Hayal bile edemeyeceğimiz sınırsız senaryo olduğu için uygulamayı %100 test edemiyoruz. • Yazılım testi, olabildiğince çok test edebileceğimiz işlevselliğin önceliğine dayanan risk tabanlı bir faaliyettir. • %100 test mümkün olmasa da %100 müşteri memnuniyetinin kesinlikle mümkün olduğuna inanıyorum. | %100 test mümkünmü?  -Bir uygulamayı %100 test etmek mümkün değil(sınırsız senaryo var)  -Her zaman son kullanıcılar yazılımda bir hata bulabilir  -Zamanla değişen genel şartlar uygulamayı hata yapmaya zorlayabilir(ops.) | Is 100% testing possible?  -It is not possible to test an application 100% (there are unlimited scenarios)  -Anytime end users can find a bug in the software  -General conditions that change over time can force the application to make mistakes (ops.) |
| 13. Test hiyerarşisi nedir?  • Birim testi > Geliştiriciler, geliştirme sırasında her bir modülü veya kod bloğunu test eder.  • Bileşen Testi > Bileşen, kendi başına çalışabilen bağımsız bir işlevdir. Eski. Amazon Alıcı İşlevselliği, Satıcı İşlevselliği, Prime Video İşlevselliği.  • Entegrasyon Testi > Tüm İşlevleri Birleştirin. Bunları entegre ettiğimde, tüm fonksiyonları kullanmaya devam edebilir miyim? Hepsinin hala çalıştığından emin olun.  • Sistem Testi > Uçtan Uca test. Baştan sona her şeyi test edin.  • Kabul Testi > Bir UAT (Kullanıcı Kabul Testi) Ekibi İşe Alın veya İş Analisti de Kabul Testi yapabilir.  Test tamamlandıktan sonra, QA ekiplerinin testinin başarılı olduğunu onaylayabilmeleri ve ürünü müşteri için hazır hale getirebilmeleri için başka bir ekibe kabul testi yaptırmanız gerekir. | Test hiyerarşisi nedir?  -Birim testi-> Yazılımın en küçük birimlerinin test edilmesidir. Bu testler, özellikle yazılımın doğru çalıştığından emin olmak için kullanılır. Birim testleri, genellikle yazılım geliştirme aşamasının erken dönemlerinde ve sık sık yürütülür.  -Entegrasyon testi->Yazılımın tüm bileşenlerinin birlikte ve doğru şekilde çalışabildiğinden emin olmak için yapılır  -Sistem Testi->Baştan sona herşeyin test edilmesidir.  -Kabul testi-> Yazılımın dağıtılmadan önce, kullanıcıların yazılımın isteklerine uygunluğunu kontrol etmek için yapılan testlerdir.  . | What is the test hierarchy?  -Unit testing-> Testing the smallest units of the software. These tests are used specifically to make sure the software is working properly. Unit tests are often run in the early stages of software development.  -Integration testing->It is done to make sure that all components of the software can work together and correctly  -System Testing->Testing everything from start to finish.  -Acceptance testing-> These are the tests carried out to check the compliance of the software with the requests of the users before the software is distributed. |