Yanıt Anahtarı

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Soru Numarası (#) | Doğru Yanıt | ÖH | K - Seviyesi | Puan |  | Soru Numarası (#) | Doğru Yanıt | ÖH | K - Seviyesi | Puan |
| 1 | b | FL-1.1.1 | K1 | 1 | 21 | d | FL-4.2.2 | K3 | 1 |
| 2 | c | FL-1.1.2 | K2 | 1 | 22 | d | FL-4.2.3 | K3 | 1 |
| 3 | b | FL-1.3.1 | K2 | 1 | 23 | a | FL-4.2.4 | K3 | 1 |
| 4 | b, e | FL-1.4.1 | K2 | 1 | 24 | c | FL-4.3.2 | K2 | 1 |
| 5 | a | FL-1.4.3 | K2 | 1 | 25 | a | FL-4.3.3 | K2 | 1 |
| 6 | c | FL-1.4.5 | K2 | 1 | 26 | b | FL-4.4.1 | K2 | 1 |
| 7 | b | FL-1.5.2 | K1 | 1 | 27 | d | FL-4.4.3 | K2 | 1 |
| 8 | a | FL-1.5.3 | K2 | 1 | 28 | b | FL-4.5.2 | K2 | 1 |
| 9 | d | FL-2.1.2 | K1 | 1 | 29 | d | FL-4.5.3 | K3 | 1 |
| 10 | d | FL-2.1.3 | K1 | 1 | 30 | a | FL-5.1.1 | K2 | 1 |
| 11 | b | FL-2.1.5 | K2 | 1 | 31 | c | FL-5.1.4 | K3 | 1 |
| 12 | c | FL-2.1.6 | K2 | 1 | 32 | a | FL-5.1.5 | K3 | 1 |
| 13 | d | FL-2.2.1 | K2 | 1 | 33 | b | FL-5.1.6 | K1 | 1 |
| 14 | b | FL-2.2.3 | K2 | 1 | 34 | d | FL-5.1.7 | K2 | 1 |
| 15 | d | FL-3.1.3 | K2 | 1 | 35 | c | FL-5.2.3 | K2 | 1 |
| 16 | a | FL-3.2.1 | K1 | 1 | 36 | b | FL-5.3.2 | K2 | 1 |
| 17 | b | FL-3.2.4 | K2 | 1 | 37 | d | FL-5.4.1 | K2 | 1 |
| 18 | b | FL-3.2.5 | K1 | 1 | 38 | b | FL-5.5.1 | K3 | 1 |
| 19 | c | FL-4.1.1 | K2 | 1 | 39 | d | FL-6.1.1 | K2 | 1 |
| 20 | c | FL-4.2.1 | K3 | 1 | 40 | d | FL-6.2.1 | K1 | 1 |

Yanıtlar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi (ÖH) | K - Seviyesi | Puan |
| 1 | b | 1. Doğru değil. Belgelendirilmiş gereksinimlerin karşılandığını onaylamak yanlıştır, çünkü onaylama kullanıcı gereksinimlerini ve beklentilerini karşılamakla ilgilidir, doğrulama ise belirtilen gereksinimleri karşılamakla ilgilidir, bu nedenle "onaylama"yı "doğrulama" ile değiştirirsek bu doğru olacaktır. 2. Bu doğru. Arızalara neden olmak ve kusurları tespit etmek muhtemelen dinamik testlerin en yaygın amacıdır. 3. Doğru değil. Hataları başlatmak ve kök nedenleri belirlemek yanlıştır, çünkü test uzmanları hataları başlatmazlar, arızalara neden olmaya çalışırlar. Hatalar genellikle geliştiriciler tarafından yapılır (ve gerçekten başlatılamaz) ve test uzmanlarının doğrudan statik test yoluyla veya dolaylı olarak dinamik testler yoluyla tespit etmeye çalıştıkları kusurlarla sonuçlanır. Kök nedenleri belirlemek faydalıdır, ancak test için ayrı bir aktivite olan hata ayıklamanın bir parçasıdır. 4. Doğru değil. Test nesnesinin kullanıcı beklentilerini karşıladığını doğrulamak yanlıştır, çünkü doğrulama belirtilen (belgelenen) gereksinimlerin karşılandığını kontrol etmekle ilgiliyken, onaylama kullanıcı gereksinimlerini ve beklentilerini karşılamakla ilgilidir, bu nedenle 'doğrulama' yerine 'onaylama' yazarsak bu doğru olacaktır. | FL-1.1.1 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 2 | c | 1. Doğru değil. Dinamik testler, daha sonrasında hataların tespit edilip düzeltilebileceği arızalara neden olur. Bununla birlikte, hata ayıklama, kusurları bulmak ve bu kusurları düzeltmekle ilgilidir. Bu nedenle, hata ayıklama arızaları düzeltmez. 2. Doğru değil. Hem test hem de hata ayıklama, test nesnesinin kalitesini iyileştirmeye katkıda bulunur, bu nedenle her ikisi de gerçekten olumlu olarak değerlendirilmelidir. Hata ayıklama, bir şeyi düzelttiği için genellikle olumlu bir aktivite olarak kabul edilir. Dinamik test, test nesnesinin kasıtlı olarak başarısız olmasına neden olmayı içerir, bu nedenle bazı insanlar bunu olumsuz bir aktivite olarak görür, ancak bu çok dar bir görüştür (ve genellikle test uzmanları tarafından benimsenen bir görüş değildir). Hem pozitif hem de negatif test durumları mümkündür. Pozitif test durumları, test nesnesinin yapması gerekeni doğru bir şekilde gerçekleştirdiğini kontrol ederken, negatif testler test nesnesinin yapmaması gerekeni yapmadığını kontrol eder. 3. Bu doğru. Testler, kusurların varlığını ya doğrudan gözden geçirmelerde kusurların gözlemlenmesi yoluyla (veya statik analizde bir araç tarafından) ya da dolaylı olarak dinamik testlerde bir arızaya neden olarak belirler. Hata ayıklama, testten ayrı bir aktivitedir (normalde geliştiriciler tarafından gerçekleştirilir) ve kusurları bulmak (yalnızca dinamik test için) ve kusurları düzeltmekle ilgilidir. 4. Doğru değil. Kusurların nedenleri tipik olarak insan hatalarıdır. Testler, ya doğrudan statik testler yoluyla ya da dolaylı olarak dinamik testler ile arızalara neden olan kusurları bulur ve hata ayıklama kusurları düzeltir. Dolayısıyla, test hataların nedenini bulmaz ve hata ayıklama kusurların nedenlerini düzeltmez. | FL-1.1.2 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 3 | b | ‘Kusurların yokluğu yanılgısı ', gereksinimlere uygun doğruluğun sağlanmasının (yani, uygulama kusurlarının yokluğunun doğrulanması) sistem ile ilgili kullanıcı memnuniyetini garanti etmediği fikriyle ilgilidir. Bunu ele almak için, sistemin kullanıcıların ihtiyaç ve beklentilerini karşıladığını, iş hedeflerini yerine getirdiğini ve rakip sistemlerden daha iyi performans gösterdiğini onaylamak da gereklidir.   1. Doğru değil. 'Test, kusurların varlığını gösterir, yokluğunu değil' prensibi, testin test nesnesindeki kusurların varlığını tespit edebilmesine rağmen, hiçbir kusur olmadığını göstermenin ve dolayısıyla doğruluğunu garanti etmenin mümkün olmadığını açıklar. Bu nedenle, testin kusurların yokluğunu göstermesinin mümkün olmadığını açıklamak, 'kusurların yokluğu‘ yanılgısını değil, bu prensibi kısmen ele alacaktır. 2. Bu doğru. Son kullanıcının kabul testi gerçekleştirmesini destekleyerek, sistemin kullanıcıların ihtiyaçlarını ve beklentilerini karşıladığını onaylamak mümkün olmalıdır. 3. Doğru değil. Teslim edilen sistemde hiçbir uygulama kusurunun kalmadığından emin olmak mümkün değildir, çünkü 'Test, kusurların varlığını gösterir, yokluğunu değil' prensibi, testin test nesnesindeki kusurların varlığını tespit edebilmesine rağmen, hiçbir kusur olmadığını göstermenin ve dolayısıyla doğruluğunu garanti etmenin mümkün olmadığını açıklar. 4. Doğru değil. Az sayıda kusurun kaldığından emin olmak için hiçbir arızaya neden olmayan testleri değiştirmek, ‘testlerin yıpranması‘ prensibini ele almanın bir yoludur. Bu prensip, değiştirilmemiş kod üzerinde aynı testlerin tekrarlanmasının yeni kusurları ortaya çıkarma olasılığının düşük olduğu ve bu nedenle testlerin değiştirilmesinin gerekli olabileceği fikriyle ilgilidir. Bu, sistemin kullanıcıların ihtiyaç ve beklentilerini karşıladığını onaylamayacaktır. | FL-1.3.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 4 | b, e | Test analizinin aşağıdaki açıklaması göz önüne alındığında:  Test gerektiren özellikleri belirlemek için test temeli analiz edilir ve test koşulları olarak tanımlanır, daha sonra ilgili risklerle birlikte önceliklendirilir. Kapsam öğeleri olarak test koşullarının sistematik olarak tanımlanması genellikle hem test analizi sırasında hem de test tasarım aktivitesinin bir parçası olarak test tekniklerinin kullanılmasını içerir.  Yukarıdaki açıklamadan test analiz ve test tasarım aktivitelerinde sıklıkla test tekniklerinin kullanıldığı görülmektedir. Sınır değer analizi ve eşdeğerlik bölümlemesi test teknikleridir.   1. Doğru değil. Test teknikleri test durumları oluştururken, test uygulaması çoğunlukla test durumlarının test prosedürleri ile birleştirilmesiyle ilgili olduğundan, test tekniklerinin kullanımını içermesi muhtemel değildir. 2. Bu doğru. Test tasarımının, test koşullarından ve kapsam öğelerinden test durumları oluşturmak için test tekniklerinin kullanılmasını içermesi muhtemeldir. 3. Doğru değil. Test yürütme, çoğunlukla test prosedürlerinin (ve dolayısıyla test durumlarının) yürütülmesiyle ilgili olduğu için test tekniklerinin kullanımını içermesi muhtemel değildir, test teknikleri ise test durumları oluşturur. 4. Doğru değil. Test izlemenin test tekniklerinin kullanımını içermesi muhtemel değildir. Test izleme çoğunlukla planın takip edildiğinden emin olmak için devam eden sürekli kontrollerle ilgilidir, test teknikleri ise test durumları oluşturur. 5. Bu doğru. Test analizinin, test koşullarını belirlemek için test tekniklerinin kullanılmasını içermesi muhtemeldir. | FL-1.4.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 5 | a | Listelenen test aktivitelerinin her birini ve bunların çıktısı test yazılımlarını göz önünde bulundurarak:   1. Test analizi - önceliklendirilmiş test koşulları (4) (örneğin, kabul kriterleri) ve test temelinde tanımlanan kusurlar için kusur raporları 2. Test tasarımı - önceliklendirilmiş test durumları, test başlatma belgeleri, kapsam öğeleri (1), test verileri gereksinimleri ve test ortamı gereksinimleri 3. Test uygulaması - test prosedürleri, otomatik test komut dosyaları, test takımları, test verileri, test yürütme zaman çizelgesi (3) ve taslaklar, sürücüler, simülatörler ve hizmet sanallaştırmaları gibi test ortamı öğeleri 4. Test tamamlama - test tamamlama raporu, belgelenmiş alınan dersler, iyileştirme için eylem öğeleri ve değişiklik talepleri (2) (ürün iş listesi öğeleri olarak)   Bu nedenle:   1. Bu doğru. Doğru eşleşme: 1B, 2D, 3C, 4A 2. Doğru değil. 3. Doğru değil. 4. Doğru değil. | FL-1.4.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 6 | c | 1. Doğru değil. Çevik yazılım geliştirmede bazı test yönetimi görevlerinin Çevik ekibin kendisi tarafından ele alınabileceğini söylemek doğru olsa da, test rolü öncelikle ekip dışından tek bir bireyin sorumluluğunda değildir. Bunun yerine, testlerin tüm ekip yaklaşımını izleyen çeşitli ekip üyeleri tarafından gerçekleştirilmesi daha olasıdır. 2. Doğru değil. Test yönetimi rolü öncelikle test planlama, test izleme ve kontrol ve test tamamlama ile ilgili aktiviteleri içerir. Dolayısıyla, bu ifade kısmen doğru olsa da, test rolünün öncelikle test izleme ve kontrolünden sorumlu olduğunu söylemek yanlıştır. 3. Bu doğru. Çevik yazılım geliştirmede, test yönetimi görevlerinin bir kısmı Çevik ekibin kendisi tarafından yerine getirilebilir. Bununla birlikte, bir organizasyon içindeki birden fazla ekibi kapsayan test aktiviteleri için, geliştirme ekibi dışındaki test yöneticileri bu görevleri gerçekleştirebilir. 4. Doğru değil. Test yönetimi rolü öncelikle test planlama, test izleme ve kontrol ve test tamamlama ile ilgili aktiviteleri içerirken, test rolü öncelikle test analizi, test tasarımı, test uygulaması ve test yürütme gibi testin teknik ve mühendislik yönlerinden sorumludur. Bu nedenle, test yönetimi rolü normalde test analizi ve test tasarımından sorumlu değildir, ancak test rolünün öncelikle test uygulama ve yürütmeden sorumlu olduğunu söylemek doğrudur. | FL-1.4.5 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 7 | b | 1. Doğru değil. Tüm ekip yaklaşımında, test uzmanları test uzmanlıklarını ekiple paylaşarak ve ürün geliştirmeye rehberlik ederek kritik bir rol oynarlar. İstenen kalite seviyelerine ulaşmak için diğer ekip üyeleriyle iş birliği yaparlar ve kabul testleri oluşturmak için iş temsilcileriyle birlikte çalışırlar. Test uzmanları ayrıca en uygun test stratejisini ve otomasyon yaklaşımlarını belirlemek için geliştiricilerle işbirliği yapar. 2. Bu doğru. Tüm ekip yaklaşımı, her bir ekip üyesinin farklı beceri setlerinden en etkili şekilde yararlanarak üstün ekip dinamiklerini teşvik eder, güçlü iletişim ve işbirliğini destekler ve projenin tamamına fayda sağlayan sinerjik bir etki yaratır. 3. Doğru değil. Tüm ekip yaklaşımı, gerekli beceri ve bilgiye sahip herhangi bir ekip üyesinin herhangi bir görevi üstlenmesine izin verir, bu nedenle uzman ekip üyeleri bu yaklaşımın bir avantajı değildir. 4. Doğru değil. Tüm ekip yaklaşımını kullanan ekiplerin optimum büyüklüğü konusunda belirli bir kılavuz yoktur ve daha büyük ekiplerin daha iyi olduğuna dair bir öneri de bulunmamaktadır. | FL-1.5.2 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 8 | a | 1. Bu doğru. Testte bağımsızlığın birincil yararı, test uzmanlarının çeşitli geçmişleri, teknik bakış açıları ve bilişsel önyargı da dahil olmak üzere potansiyel önyargıları nedeniyle geliştiricilere kıyasla farklı arıza ve kusur türlerini belirleme olasılıklarının daha yüksek olmasıdır. Bununla birlikte, testte bağımsızlığın ana dezavantajı, test uzmanlarının geliştirme ekibinden izole olabilmeleri ve bunun iletişim sorunlarına, iş birliği eksikliğine ve potansiyel olarak düşmanca bir ilişkiye yol açması ve test uzmanlarının sürüm sürecindeki gecikmeler ve darboğazlar için suçlanmasıdır. 2. Doğru değil. Bir geliştiricinin koda aşina olması, kodda nadiren kusur bulacağı anlamına gelmez, bunun yerine bu aşinalık, kendi kodlarında birçok kusuru verimli bir şekilde bulabilecekleri anlamına gelir. Ayrıca, geliştiricilerin ve test uzmanlarının ortak bir geçmişe sahip olmasından ziyade, geliştiricilerin test uzmanlarından farklı bir geçmişe sahip olması, normalde test uzmanlarının ve geliştiricilerin farklı türde kusurlar bulmasının nedeni olarak gösterilmektedir. 3. Doğru değil. Testler, testi yazan için hiç bağımsızlık olmamasından organizasyon dışından gelen test uzmanları için çok yüksek bağımsızlığa kadar değişen farklı bağımsızlık seviyelerinde gerçekleştirilebilir. Çoğu projede, geliştiricilerin bileşen ve bileşen entegrasyon testlerini, test ekibinin sistem ve sistem entegrasyon testlerini ve iş temsilcilerinin kabul testlerini gerçekleştirdiği çoklu bağımsızlık seviyeleri kullanılır. Bu nedenle, test uzmanları geliştiricinin ekibinde olabilir ve organizasyon dışından gelmelerine gerek yoktur. Uygulama alanı bilgisi durumdan duruma değişecektir ve bağımsızlık seviyesine bağlı değildir. 4. Doğru değil. Testler, testleri yazan için hiç bağımsızlık olmamasından organizasyon dışından gelen test uzmanları için çok yüksek bağımsızlığa kadar değişen farklı bağımsızlık seviyelerinde gerçekleştirilebilir; geliştirici ekibinin dışından gelen test uzmanları genellikle ekip içinden gelen test uzmanlarından daha bağımsızdır. Bununla birlikte, ekip dışından gelen test uzmanlarının geliştiricilerden daha fazla izole edileceğine ve dolayısıyla ürün sürümündeki gecikmelerden sorumlu tutulma olasılıklarının daha yüksek olacağına inanmak için daha fazla neden vardır. | FL-1.5.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 9 | d | 1. Doğru değil. Kalite kontrol tüm geliştirme aktiviteleri için geçerlidir, yani her yazılım geliştirme aktivitesine karşılık gelen bir test aktivitesi vardır. Bununla birlikte, burada test seviyelerini geliştirme seviyeleriyle eşitlemeye çalışıyoruz ve ‘test seviyeleri' ile ne kastedildiğini bilmemize rağmen, ’geliştirme seviyesi‘ terimine ilişkin ortak bir anlayış bulunmamaktadır. 2. Doğru değil. Her yazılım geliştirme aktivitesine karşılık gelen bir test aktivitesi vardır; ancak test hedefleri oldukça farklıdır. Örneğin, bir test nesnesinin teslimattan önce belirli bir test türünün gerçekleştirilmesi gerektiğine dair bir sözleşme şartına uymasını sağlamaya yönelik bir test hedefi olabilir. Bu durumda, ilgili bir geliştirme hedefinin olması için hiçbir neden yoktur. 3. Doğru değil. Kalite kontrol tüm geliştirme aktiviteleri için geçerlidir, yani her yazılım geliştirme aktivitesine karşılık gelen bir test aktivitesi vardır. Ancak, aynı simetri test ve kullanıcı aktiviteleri için geçerli değildir. Örneğin, bazı sistemler için son kullanıcıları tanımlamak bile zordur. Ayrıca, bazı test aktiviteleri, kullanıcı yönü olmayan geliştiricilere (örneğin, bakım kolaylığı için test) odaklanır. 4. Bu doğru. Kalite kontrol tüm geliştirme aktiviteleri için geçerlidir, yani her yazılım geliştirme aktivitesine karşılık gelen bir test aktivitesi vardır. | FL-2.1.2 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 10 | d | 1. Doğru değil. Bileşen Testi Odaklı Geliştirme, geliştirmeye yönelik test öncelikli yaklaşımın doğru bir örneği değildir. 2. Doğru değil. Entegrasyon Testi Odaklı Geliştirme, geliştirmeye yönelik test öncelikli yaklaşımın doğru bir örneği değildir. 3. Doğru değil. Sistem Testi Odaklı Geliştirme, geliştirmeye yönelik test öncelikli yaklaşımın doğru bir örneği değildir. 4. Bu doğru. Kabul Testi Odaklı Geliştirme (ATDD), geliştirmeye yönelik test öncelikli yaklaşımın iyi bilinen bir örneğidir. | FL-2.1.3 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 11 | b | 1. Doğru değil. Sola kaydırma testine dahil olan uygulamalar, SDLC'yi soldan sağa doğru hareket eden bir şekilde tasvir ederek, geliştirme yaşam döngüsünün erken fazlarında daha fazla test aktivitesi uygulamayı amaçlamaktadır.   Test sürecinin sol tarafı diye bir şey yoktur   1. Bu doğru. Sola kaydırma, yazılım geliştirme yaşam döngüsünde (SDLC) teste daha erken başlamanın önemini vurgulamaktadır. Sola kaydırma testlerinin uygulanması, SDLC'nin erken fazlarında ek eğitim ve daha fazla çaba ve maliyet gerektirir, ancak genel tasarruf daha yüksek olmalıdır. 2. Doğru değil. Regresyon testi için otomatik bileşen testleri ve bileşen entegrasyon testleri genellikle değerli olsa da, bu testlerin oluşturulması normalde geliştiricilerin sorumluluğundadır ve sürekli entegrasyon/sürekli teslimat (CI/CD) yaklaşımı izleniyorsa, bu testler kodla birlikte sunulmuş olacaktır. Bazı durumlarda test uzmanı regresyon testleri ve hatta bazen bileşen testleri ve bileşen entegrasyon testleri için testleri otomatikleştirebilir, ancak bu durum testleri SDLC'nin başlarına taşıyan 'sola kaydırma' yaklaşımının bir parçası değildir. 3. Doğru değil. Test uzmanlarını SDLC'nin erken fazlarında görevlerini yerine getirmeleri için eğitmek, testlere SDLC'nin erken fazlarında başlamanın önemini vurgulayarak sola kaydırma yaklaşımını destekleyecektir.   Ancak, SDLC'nin ilerleyen fazlarında gerçekleştirilmek üzere daha fazla test aktivitesinin otomatikleştirilmesi "sola kaydırma" yaklaşımının bir parçası değildir. | FL-2.1.5 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 12 | c | 1. Doğru değil. Retrospektiflerin amaçlarından biri, uygulamaya konulması halinde geliştirme sürecinin gelecekteki çıktılarının (test nesneleri) kalitesinin daha yüksek olmasıyla sonuçlanacak potansiyel süreç iyileştirmelerini belirlemektir. Bu nedenle, bunun bir retrospektifin sonucu olarak ortaya çıkması muhtemeldir. 2. Doğru değil. Test için retrospektiflerin bir faydası, süreç iyileştirmeleri yoluyla test verimliliğinin artırılmasını içerir. Bu nedenle, bunun bir retrospektifin sonucu olarak ortaya çıkması muhtemeldir. 3. Bu doğru. Retrospektif katılımcıları genellikle test uzmanları, geliştiriciler, mimarlar, ürün sahipleri ve iş analistlerini içerir, ancak son kullanıcılar nadiren davet edilir veya bu toplantılara katılır - ve bu toplantılardan herhangi bir rapor almaları da pek olası değildir. Dolayısıyla, retrospektifler aracılığıyla geliştirme ve test süreçleri hakkında daha fazla şey öğrenmeleri ve anlamaları pek olası değildir 4. Doğru değil. Test için retrospektiflerin bir yararı, geliştiricilerle ortak incelemeler yoluyla test yazılımının (otomatik test komut dosyaları dahil) kalitesinin artırılmasını içerir. Bu nedenle, bunun bir retrospektifin sonucu olarak ortaya çıkması muhtemeldir. | FL-2.1.6 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 13 | d | 1. Doğru değil. Bileşen testi (birim testi olarak da adlandırılır), tek tek bileşenlerin izole bir şekilde test edilmesini içerir ve çoğunlukla kullanıcı ihtiyaçlarına göre doğrulamadan ziyade bir spesifikasyona göre doğrulanır. Ancak, bu test normalde test uzmanları tarafından gerçekleştirilmez, çünkü geliştiriciler bu testi genellikle kendi geliştirme ortamlarında gerçekleştirirler. 2. Doğru değil. Bileşen entegrasyon testi, bileşenler arasındaki arayüzlerin ve etkileşimlerin test edilmesini içerir ve çoğunlukla kullanıcı ihtiyaçlarına göre onaylamadan ziyade bir spesifikasyona göre doğrulamadır. Ancak bu test normalde test uzmanları tarafından yapılmaz, çünkü bu testi genellikle geliştiriciler yapar. 3. Doğru değil. Sistem entegrasyon testi, diğer sistemler ve harici hizmetlerle olan arayüzleri inceler ve çoğunlukla kullanıcı ihtiyaçlarına göre onaylamadan ziyade bir spesifikasyona göre doğrulamadır. Bu tür testler de çoğunlukla test uzmanları tarafından gerçekleştirilir. 4. Bu doğru. Kabul testi, sistemin kullanıcının iş ihtiyaçlarını karşıladığını ve dağıtıma hazır olduğunu onaylamaya odaklanır.   İdeal olarak, bu test son kullanıcılar tarafından gerçekleştirilir | FL-2.2.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 14 | b | 1. Doğru değil. Güncellemelerin doğru bir uygulama ile sonuçlandığını kontrol etmek için doğrulama testi gereklidir, ancak daha sonra sistemin değişmemiş alanlarında herhangi bir kusurun ortaya çıkmadığından veya ortaya çıkarılmadığından emin olmak için regresyon testi yapmak mantıklı olacaktır. 2. Bu doğru. Doğrulama testi, güncellemelerin doğru bir uygulama ile sonuçlandığını kontrol edecek ve daha sonra sistemin değişmemiş alanlarında herhangi bir kusurun ortaya çıkmadığından veya mevcut olmadığından emin olmak için regresyon testi kullanılacaktır. 3. Doğru değil. Güncelleme yapıldığında sistemin değişmemiş alanlarında herhangi bir kusurun ortaya çıkmadığından veya ortaya çıkmadığından emin olmak için regresyon testi kullanılmalıdır, ancak güncellemelerin doğru bir uygulama ile sonuçlanıp sonuçlanmadığını kontrol edecek doğrulama testlerinin yapılması da gereklidir. 4. Doğru değil. Doğrulama testi, güncellemelerin doğru bir uygulama ile sonuçlandığını kontrol edecek ve sistemin değişmemiş alanlarında herhangi bir kusurun ortaya çıkmadığından veya ortaya çıkarılmadığından emin olmak için regresyon testi kullanılacaktır. Bununla birlikte, gerçekleştirildiğinde (yani, bir güncellemenin test edilmesi gerektiğinde), doğrulama testi regresyon testinden daha önce gelir. | FL-2.2.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15 | d | Listelenen örnek kusurların her birini göz önünde bulundurarak:   1. Tasarımın karmaşıklığı nedeniyle tasarım şartnamesinin (spesifikasyonunun) iki farklı bölümü aynı fikirde değildir – bu, statik testlerle en kolay bulunabilecek tutarsızlıkları, belirsizlikleri, çelişkileri, eksiklikleri, yanlışlıkları ve tekrarları içeren bir şartname kusurunun bir örneğidir. 2. Yanıt süresi çok uzun ve bu nedenle kullanıcıların sabrının taşmasına neden oluyor. Bu, yalnızca programı yürüterek ve yanıt süresini ölçerek pratikte tespit edilebilen ve dinamik testlerle en kolay bulunabilen bir yanıt süresi kusuru örneğidir. 3. Koddaki bir yola yürütme sırasında ulaşılamıyor - bu, tanımlanmamış değerlere sahip değişkenleri, tanımlanmamış değişkenleri, yinelenen veya ulaşılamayan kodu ve aşırı kod karmaşıklığını içeren ve statik testlerle kolay şekilde bulunabilen bir kodlama kusuruna örnektir. 4. Bir değişken tanımlanır, ancak daha sonra programda asla kullanılmaz - bu, tanımlanmamış değerlere sahip değişkenleri, tanımlanmamış değişkenleri, kopyalanmış veya erişilemeyen kodu ve aşırı kod karmaşıklığını içeren ve statik testlerle kolay şekilde bulunabilen bir kodlama kusuru örneğidir. 5. Programın bir rapor oluşturmak için ihtiyaç duyduğu bellek miktarı çok yüksek – bu, yalnızca programı yürüterek ve kullanılan belleği ölçerek pratikte tespit edilebilen ve dinamik testlerle en kolay bulunabilen bir performans kusuru örneğidir.   Bu nedenle:   1. Doğru değil. 2. Doğru değil. 3. Doğru değil. 4. Bu doğru. Statik test için doğru eşleşme i, iii ve iv'dir. | FL-3.1.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 16 | a | 1. Bu doğru. Yazılım geliştirme sürecinde paydaşlardan erken ve sıklıkla geri bildirim almak oldukça faydalı olabilir. Potansiyel kalite sorunlarının erken iletişimini kolaylaştırır, gereksinimler hakkında yanlış anlamaları önleyebilir ve paydaş gereksinimlerindeki herhangi bir değişikliğin daha erken anlaşılmasını ve uygulanmasını sağlar. 2. Doğru değil. Geri bildirim paydaşlardan gelmektedir ve paydaşların geri bildirimde bulunmasının kendi kullanıcı gereksinimlerine ilişkin anlayışlarını geliştirmesi pek olası değildir. 3. Doğru değil. Yazılım geliştirme sürecinde paydaşlardan erken ve sıklıkla geri bildirim almak oldukça faydalı olabilir. Potansiyel kalite sorunlarının erken iletişimini kolaylaştırır, gereksinimler hakkında yanlış anlamaları önleyebilir ve paydaş gereksinimlerindeki herhangi bir değişikliğin daha erken anlaşılmasını ve uygulanmasını sağlar. Bununla birlikte, gereksinimlerdeki değişikliklerin daha erken anlaşılabilmesi ve uygulanabilmesi, gereksinimlerde sınırsız değişiklik yapılmasının teşvik edildiği anlamına gelmez. 4. Doğru değil. Geri bildirim paydaşlardan gelir ve onlarla iletişimi kapsamamaktadır. Son kullanıcılara yönelik iletişimler, piyasaya sürülmeden önce hangi gereksinimlerin uygulanmayacağının kendilerine söylenmesini içerebilir, ancak ideal olarak bu hiç gerçekleşmemelidir. | FL-3.2.1 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 17 | b | Listelenen gözden geçirme türlerinin her birini göz önünde bulundurarak:   1. Teknik gözden geçirme - Bu tür bir gözden geçirme, teknik olarak kalifiye gözden geçiriciler tarafından gerçekleştirilir ve bir moderatör tarafından yönetilir. Hedefler, teknik sorunlar üzerinde fikir birliği sağlamak ve kararlar almak, aynı zamanda kaliteyi değerlendirmek ve iş ürününe güven oluşturmak, yeni fikirler üretmek, yazarları motive etmek ve geliştirmelerini sağlamak ve anormallikleri tespit etmektir. 2. Kurallı olmayan gözden geçirme - Temel hedef anormallikleri tespit etmektir. Süreç tanımlanmamıştır ve kurallı olarak belgelenmiş çıktı gerektirmez. 3. Tetkik - Bu, en kurallı gözden geçirme türüdür ve genel gözden geçirme sürecinin tamamını takip eder. Birincil hedef en fazla sayıda anomaliyi bulmaktır ve diğer hedefler arasında kaliteyi değerlendirmek ve iş ürününe güven oluşturmak, yazarları iyileştirme konusunda motive etmek ve etkinleşmelerini sağlamak ve tetkik süreci de dahil olmak üzere yazılım geliştirme yaşam döngüsünü (SDLC) geliştirmek için kullanılabilecek metrikleri toplamak yer almaktadır. Yazar, gözden geçirme lideri veya yazıcı olarak hareket edemez. 4. Üzerinden geçme - Yazar tarafından yönetilen bu tür bir inceleme, kaliteyi değerlendirmek ve iş ürününe güven oluşturmak, gözden geçiricileri eğitmek, fikir birliği sağlamak, yeni fikirler üretmek, yazarları motive etmek ve gelişmelerine olanak sağlamak ve anormallikleri tespit etmek gibi çeşitli hedeflere hizmet eder. Gözden geçiriciler üzerinden geçmeden önce bireysel bir gözden geçirme yapabilir, ancak bu zorunlu değildir.    1. Fikir birliği sağlama, yeni fikirler üretme ve yazarları iyileştirme konusunda motive etme gibi hedefleri içerir (A)    2. Temel hedef potansiyel kusurları tespit etmektir ve kurallı olarak belgelenmiş bir çıktı üretmez (D)    3. Temel hedef potansiyel kusurları tespit etmektir ve süreç iyileştirmeyi desteklemek için metriklerin toplanmasını gerektirir (C) | FL-3.2.4 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
|  |  | D. Gözden geçiricileri eğitmek, fikir birliğine varmak, yeni fikirler üretmek ve potansiyel kusurları tespit etmek gibi hedefleri içerir (B)  Bu nedenle:   1. Doğru değil. 2. Bu doğru. Doğru eşleşme: 1A, 2D, 3C, 4B 3. Doğru değil. 4. Doğru değil. |  |  |  |
| 18 | b | 1. Doğru değil. Başarılı gözden geçirmeler sağlamak için, yönetimin gözden geçirme sürecine desteğini sağlamak önemlidir, ancak bu durum onların gözden geçiriciler olarak katılmaları gerektiği anlamına gelmez. 2. Bu doğru. Başarılı gözden geçirmeler sağlamak için, gözden geçiricilerin bireysel gözden geçirme veya gözden geçirme toplantıları sırasında odaklarını kaybetmelerini önlemek amacıyla iş ürününü makul bir zaman çizelgesi içinde gözden geçirilebilecek kadar küçük parçalara ayırmak önemlidir. 3. Doğru değil. Başarılı gözden geçirmeler sağlamak için, katılımcıları değerlendirmeksizin hedefleri ve ölçülebilir çıkış kriterlerini açıkça tanımlamak önemlidir. 4. Doğru değil. Başarılı gözden geçirmeler sağlamak için, gözden geçiricilerin bireysel gözden geçirme veya gözden geçirme toplantıları sırasında odağını kaybetmesini önlemek için gözden geçirmeyi daha küçük parçalara bölmek önemlidir. Bu nedenle, her gözden geçirme için bir belgeyi ele almayı planlamamalısınız. | FL-3.2.5 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 19 | c | 1. Doğru değil. Çoğu durumda, aynı test nesneleri için hem kara kutu test teknikleri hem de deneyim temelli test teknikleri kullanılabilir. 2. Doğru değil. Hem kara kutu test teknikleri hem de deneyim temelli test teknikleri tüm test seviyelerinde kullanılabilir. 3. Bu doğru. Kara kutu test teknikleri (spesifikasyona (şartnameye) dayalı teknikler olarak da bilinir), test nesnesinin iç yapısına atıfta bulunmadan belirtilen davranışının analizine dayanır. Bu nedenle, test temeli genellikle bir spesifikasyondur. Deneyim temelli test teknikleri, test durumlarının tasarımı ve uygulanması için test uzmanlarının bilgi ve deneyimlerini etkili bir şekilde kullanır. Bu, test uzmanının testleri tasarlarken spesifikasyonu hiç kullanmayabileceği anlamına gelir. 4. Doğru değil. Deneyim temelli test teknikleri, kara kutu (ve beyaz kutu) test tekniklerini kullanarak gözden kaçabilecek kusurları tespit edebilir. Dolayısıyla, deneyim temelli test teknikleri, kara kutu test tekniklerini ve beyaz kutu test tekniklerini tamamlayıcı niteliktedir ve hem kara kutu test teknikleri hem de deneyim temelli test teknikleri tüm Yazılım Geliştirme Yaşam Döngülerinde kullanılabilir. | FL-4.1.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 20 | c | 1. Doğru değil. Bu üç değer, eşdeğerlik bölümlerinin tam kapsamını sağlar, ancak iki değerle de tam kapsam elde edilebilir. 2. Doğru değil. Bu üç değer, eşdeğerlik bölümlerinin tam kapsamını sağlar, ancak iki değerle de tam kapsam elde edilebilir. 3. Bu doğru. "1" değeri "hatalı uzunluk" ve "hatalı farklı basamak sayısı"nı kapsar. "1234" değeri "doğru uzunluk" ve "doğru farklı basamak sayısı"nı kapsar. Bu iki değer, tanımlanan dört eşdeğerlik bölümünün tümünü kapsar. 4. Doğru değil. Bu küme "hatalı farklı basamak sayısı" eşdeğerlik bölümünü kapsamamaktadır. | FL-4.2.1 | K3 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 21 | d | Eşdeğerlik bölümleri şunlardır: {…, 99, 100}, {101, 102 ,…, 198, 199}, {200, 201, …}.  Bu nedenle, 100, 101, 199 ve 200 olmak üzere 4 sınır değeri vardır.  2 değerli sınır değer analizinde, her sınır değeri için iki kapsam öğesi vardır (sınır değeri ve bitişik bölüme ait en yakın komşusu). En yakın komşular da bitişik bölümdeki sınır değerler olduğundan, sadece dört kapsam öğesi vardır.  Bu nedenle:   1. Doğru değil. Sadece 100 ve 200, 2 değerli sınır değer analizi için geçerli kapsam öğeleridir, bu nedenle %50 kapsam elde ederiz. 2. Doğru değil. Sadece 100 ve 200, 2 değerli sınır değer analizi için geçerli kapsam öğeleridir, bu nedenle %50 kapsam elde ederiz. 3. Doğru değil. Sadece 100 ve 101, 2 değerli sınır değer analizi için geçerli kapsam öğeleridir, bu nedenle %50 kapsam elde ederiz. 4. Bu doğru. 101, 199 ve 200, 2 değerli BVA için geçerli kapsam öğeleridir, bu nedenle %75 kapsam elde ederiz. | FL-4.2.2 | K3 | 1 |
| 22 | d | 1. Doğru değil. (D, D, Y) kombinasyonu hiçbir kurala uymamaktadır. Bu bir hariç tutma örneğidir, çelişki değildir. 2. Doğru değil. (D, Y, D) kombinasyonu yalnızca bir sütun olan R2 ile eşleşir, bu nedenle herhangi bir çelişki yoktur. 3. Doğru değil. Her iki kombinasyon (D, D, D) ve (Y, D, D) sadece bir sütun, R1 ile eşleşir, bu nedenle herhangi bir çelişki yoktur. 4. Bu doğru. (Y, Y, Y) kombinasyonu hem R2 hem de R3 ile eşleşir, ancak R2 ve R3 farklı eylemlere sahiptir, bu nedenle R2 ve R3 arasında bir çelişki olduğunu gösterir. | FL-4.2.3 | K3 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 23 | a | Aşağıdaki üç geçiş:  “TALEP EDİYOR -> ONAYLANDI” “BEKLEME LİSTESİ -> ONAYLANDI” “BEKLEME LİSTESİ -> SON” aynı test durumunda görülemez, bu da en az üç test durumunun gerekli olduğunu gösterir.  Diğer tüm geçişler bu üç geçişten biri veya daha fazlası ile birlikte görünebilir, bu nedenle en az üç test durumuna ihtiyacımız vardır. Aslında, sadece üç dizi mümkündür:  TD1: BAŞLAT (Oda talebi) TALEP EDİYOR (Mevcut)  ONAYLANDI (Ödeme) SON  TD2: BAŞLAT (Oda talebi) TALEP EDİYOR (Mevcut Değil)  BEKLEME LİSTESİ (Mevcut) ONAYLANDI (Ödeme) SON  TD3: BAŞLAT (Oda talebi) TALEP EDİYOR (Mevcut değil)  BEKLEME LİSTESİ (İptal) SON  Bu nedenle:   1. Bu doğru. 2. Doğru değil. 3. Doğru değil. 4. Doğru değil. | FL-4.2.4 | K3 | 1 |
| 24 | c | Dal testinde, kapsam öğeleri, bir kontrol akış grafiğinin kenarlarıyla temsil edilen dallardır. Kontrol akış grafiğinde 8 kenar bulunmaktadır.  Bu nedenle:   1. Doğru değil. 2. Doğru değil. 3. Bu doğru. 4. Doğru değil. | FL-4.3.2 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 25 | a | 1. Bu doğru. Yalnızca kara kutu testi yapmak, gerçek kod kapsamının bir ölçüsünü sağlamaz. Beyaz kutu kapsam ölçümleri, kapsamın objektif bir ölçümünü sağlar ve bu kapsamı artırmak için ek testlerin oluşturulmasına izin vermek için gerekli bilgileri sağlar ve ardından koda olan güveni artırır. 2. Doğru değil. Bu ifade yanlıştır ve ayrıca soruyu cevaplamaz (kara kutu testi ile ilgisi yoktur) 3. Doğru değil. Genel olarak beyaz kutu ve kara kutu teknikleri arasında herhangi bir ilişki yoktur. 4. Doğru değil. Test nesnesinin kendisine dayalı testler tasarlamak için beyaz kutu teknikleri kullanılırken, spesifikasyona dayalı testler tasarlamak için kara kutu teknikleri kullanılır. Bu nedenle,   Dolayısıyla, bu iki tür teknikten elde edilen kapsam öğeleri arasında bir ilişki yoktur. | FL-4.3.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 26 | b | 1. Doğru değil. Keşif testi, olası kusurların/arızaların bir listesini değil, test başlatma belgelerini kullanır. Keşif testi, diğer test tekniklerinin kullanımını içerebilse de, bu durumda hata saldırısı en olası seçenektir. 2. Bu doğru. Bu, olası arızaların bir listesidir. Hata saldırıları, hata tahmininin uygulanmasına yönelik metodik bir yaklaşımdır ve test uzmanının olası hataların, kusurların ve arızaların bir listesini oluşturmasını veya elde etmesini ve hatalarla ilişkili kusurları tanımlayacak, kusurları ortaya çıkaracak veya arızalara neden olacak testler tasarlamasını gerektirir. 3. Doğru değil. Test uzmanı, testlerini desteklemek için bir kontrol listesi kullanıyor. Hem hata tahmini hem de kontrol listesi tabanlı testler bu tür listeleri kullanılır, ancak buradaki liste test koşulları değil, olası arızalar listesidir ve bu nedenle EN OLASI test tekniği, hatalara, kusurlara ve arızalara odaklanan hata saldırısıdır. 4. Doğru değil. Sınır değer analizi, eşdeğerlik bölümlerinin sınır değerlerinin analizine dayanmaktadır. Yukarıdaki liste eşdeğerlik bölümlerinden veya bunların sınırlarından bahsetmemektedir. | FL-4.4.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 27 | d | 1. Doğru değil. Test uzmanının kontrol listesine dayalı olarak ayrıntılı test durumlarını uygulayabileceği ve yürütebileceği doğru olsa da, bunun kapsamı nasıl artıracağı açıklanmamaktadır. 2. Doğru değil. Kontrol listesi öğeleri otomatikleştirilmemelidir. Ancak öyle olsalar bile, otomatik test komut dosyaları testleri her zaman aynı şekilde yürütür, bu da genellikle kapsamın artmasına neden olmaz. 3. Doğru değil. Her bir kontrol listesi öğesinin ayrı ayrı ve bağımsız olarak test edilmesi gerektiği doğrudur. Ancak bu, test yürütme sırasını etkiler ve elde edilen kapsamı etkilemez ve bu nedenle kapsamın artmasına neden olmaz. 4. Bu doğru. Kontrol listeleri üst seviye ise, gerçek testte bazı değişkenliklerin ortaya çıkması muhtemeldir, bu da potansiyel olarak daha geniş kapsama alanı ancak daha az tekrarlanabilirlik ile sonuçlanır. İki test uzmanı üst seviye öğelerden oluşan bir kontrol listesini takip ederse, her biri farklı test verileri, test adımları vb. kullanabilir. Bu şekilde, bir test uzmanı muhtemelen diğer test uzmanı tarafından kapsanmayan bazı alanları kapsayacak ve bu da kapsamın artmasına neden olacaktır. | FL-4.4.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 28 | b | 1. Doğru değil. Bu kabul kriteri, sistemin hangi kurallara veya düzenlemelere uyması gerektiğini açıklar (bu durumda unutulma hakkı). Bu, kural odaklı bir kabul kriterinin bir örneğidir. 2. Bu doğru. Bu kabul kriteri, sistem tarafından gerçekleştirilmesi gereken örnek bir senaryoyu açıklamaktadır. Bu, senaryo odaklı bir kabul kriterinin bir örneğidir. 3. Doğru değil. Bu cümle daha çok bazı iş kurallarını uygulayan bir kod satırına benziyor. Kabul kriterleri iş temsilcileri ile iş birliği içinde yazılmalı ve bu nedenle onların anladıkları dilde yazılmalıdır. Bu cümle büyük olasılıkla bu paydaşlar için anlaşılmaz olacaktır. 4. Doğru değil. Bu kabul kriteri, sistemin hangi kurallara veya düzenlemelere uyması gerektiğini ve uyumluluğun nasıl sağlanacağını açıklar. Dolayısıyla bu, senaryo tabanlı bir kabul kriterine değil, kural odaklı bir kabul kriterine örnektir. | FL-4.5.2 | K2 | 1 |
| 29 | d | 1. Doğru değil. Özel kullanıcıların Normal kullanıcıların haklarına sahip olup olmadığını kontrol etmek istiyoruz, bu nedenle Normal bir kullanıcı için değil, Özel bir kullanıcı için erişim haklarını test etmemiz gerekiyor. 2. Doğru değil. Özel kullanıcıların Normal kullanıcıların haklarına sahip olup olmadığını kontrol etmek istiyoruz, bu nedenle Normal bir kullanıcı için değil, Özel bir kullanıcı için erişim haklarını test etmemiz gerekiyor. 3. Doğru değil. Kabul kriterlerinde tanımlanan 5. kat yoktur. Test durumları, kullanıcı hikayesinin kapsamını genişletmemelidir. Ancak negatif test yapmak istesek bile, bu test doğrudan AC3 ile ilgili değildir. 4. Bu doğru. Bu şekilde, Özel bir kullanıcının Normal bir kullanıcının erişebildiği katlara erişip erişemeyeceğini kontrol edebiliriz. | FL-4.5.3 | K3 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 30 | a | 1. Bu doğru. Test planı, test verileri *gereksinimlerini* içerebilir (test yaklaşımının bir parçası olarak), ancak test durumları için ayrıntılı test verilerini içermeyebilir. Test verileri test planının değil, test durumlarının bir parçasıdır. Ayrıca, test planı oluşturulduğunda bu tür verileri tanımlamak genellikle imkansızdır, çünkü bileşenlerin neye benzeyeceği tam olarak bilinmemektedir. 2. Doğru değil. Bir test planının amaçlarından biri, giriş kriterleri ve çıkış kriterlerini dahil ederek, gerçekleştirilen test aktivitelerinin belirlenen kriterleri karşılamasını sağlamaya yardımcı olmaktır. Kod kapsama kriterleri, bileşen test seviyesi için bu tür kriterlere bir örnektir. 3. Doğru değil. Dokümantasyon şablonları, bir test planının tipik içeriğidir. Bu, iletişim veya raporlama için standart bir yöntem tanımlayarak paydaşlar arasındaki iletişimi kolaylaştırmaya yardımcı olur 4. Doğru değil. Bir test planının amaçlarından biri, testin mevcut test politikasına ve test stratejisine bağlı kalacağını göstermek veya testin bunlardan neden sapacağını açıklamaktır. Bu, takip edilecek (veya edilmeyecek) test seviyelerine ilişkin sapmayı açıklayan bir örnektir. | FL-5.1.1 | K2 | 1 |
| 31 | c | Elimizdeki grafikten:  A(4)=6 ve A(3)=8 (son iki gri kutu).  Elde ettiğimiz formülden:  E(5) = (3\*A(4) + A(3)) / 4 = (3\*6+8) / 4 = 26 / 4 = 6.5 kişi-gün.  Bu nedenle:   1. Doğru değil. 2. Doğru değil. 3. Bu doğru. 4. Doğru değil. | FL-5.1.4 | K3 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 32 | a | Test durumlarını önceliklerine göre yürütmek istiyoruz, ancak bağımlılıkları da göz önünde bulundurmamız gerekiyor.  Sadece öncelikleri göz önünde bulundurursak, önce TD 5 ve TD 7 'yi (en yüksek öncelik), ardından TD 1, TD 3 ve TD 4'ü ve son olarak TD 2 ve TD 6'yı (en düşük öncelik) çalıştırmak istiyoruz.  Ancak, TD 7'yi çalıştırmak için önce TD 4'ü çalıştırmamız gerekir.  TD 5'i çalıştırmak için TD 4 ve TD 2'yi çalıştırmamız gerekir, ancak TD 2, TD 2'den önce çalıştırılması gereken TD 1 tarafından engellenir.  Bu nedenle, 1. öncelikli test durumlarını mümkün olduğunca erken çalıştırmak için ilk beş test durumu şöyle olmalıdır: TD 4 - TD 7 - TD 1 - TD 2 - TD 5.  Daha sonra, TD 3'ü çalıştırmamız gerekiyor, çünkü TD 6'dan daha yüksek önceliğe sahiptir.  Böylece tam çizelge TD 4 – TD 7 – TD 1 – TD 2 – TD 5 – TD 3 – TD 6 olacaktır.  Yani, altıncı test durumu TD 3 olacaktır.  Bu nedenle:   1. Bu doğru. 2. Doğru değil. 3. Doğru değil. 4. Doğru değil. | FL-5.1.5 | K3 | 1 |
| 33 | b | 1. Doğru değil. Test piramidi modeli, test öncelikleri hakkında bilgi sağlamaz. 2. Bu doğru. Test piramidi modeli, farklı testlerin farklı ayrıntı düzeylerine sahip olduğunu gösterir. 3. Doğru değil. Test piramidi modeli kapsama kriterlerinden bağımsızdır. 4. Doğru değil. Test piramidi modeli farklı testler arasındaki herhangi bir ilişkiyi göstermez. | FL-5.1.6 | K1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 34 | d | 1. Doğru değil. Test kuadrantları, test seviyelerini ve test türlerini çeşitli kriterlere göre ayrı ayrı gruplandırır. Test seviyelerinin ve test türlerinin herhangi bir kombinasyonunu temsil etmezler ve bir yazılım geliştirme yaşam döngüsü içindeki herhangi bir konumla ilgili değildirler. Test kuadrantları modelinde hem test seviyeleri hem de test türleri ayrı ayrı ele alınır. 2. Doğru değil. Test kuadrantları, test seviyelerini ve test türlerini çeşitli kriterlere göre gruplandırır. Her bir test seviyesinde gerçekleştirilen tekil test türlerinin ayrıntı derecesini tanımlamazlar. Test seviyelerine ilişkin böyle bir modele test piramidi olarak adlandırılır. 3. Doğru değil. Bu ifade yanlıştır, çünkü genel olarak herhangi bir test türü herhangi bir test seviyesinde gerçekleştirilebilir. 4. Bu doğru. Test kuadrantları, çevik yazılım geliştirmede test seviyelerini, test türlerini, aktivitelerini, test tekniklerini ve iş ürünlerini gruplandırır. Bu modelde, testler işe yönelik veya teknolojiye yönelik olabilir. Testler ekibi destekleyebilir (yani, geliştirmeye kılavuzluk edebilir) veya ürünü eleştirebilir (yani, davranışını beklentilere kıyasla ölçebilir). Bu iki bakış açısının birleşimi dört kuadrantı belirler. | FL-5.1.7 | K2 | 1 |
| 35 | c | 1. Doğru değil. Risk izleme, risk analizinin değil, risk kontrolünün bir parçasıdır. 2. Doğru değil. Risk tanımlamasının kendisi, risk azaltma aktivitelerini uygulamamıza izin vermez. Azaltıcı eylemler risk kontrol fazında tanımlanır. 3. Bu doğru. Bu, risk analizinin testin kapsamını ve eksiksizliğini nasıl etkilediğinin bir örneğidir. 4. Doğru değil. Kapsam öğeleri risk analizi yoluyla değil, test teknikleri kullanılarak türetilir. | FL-5.2.3 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 36 | b | 1. Doğru değil. Test ilerleme raporları çoğunlukla test tasarımı sırasında değil, test izleme ve kontrol ve test tamamlama sırasında kullanılır. 2. Bu doğru. Test tamamlama raporu testin tamamlanması sırasında, bir proje, test seviyesi veya test türü tamamlandığında ve ideal olarak çıkış kriterleri karşılandığında hazırlanır. Bu rapor, test ilerleme raporlarından gelen bilgileri ve diğer verileri kullanır. 3. Doğru değil. Test ilerleme raporları çoğunlukla test analizi sırasında değil, test izleme ve kontrol ve test tamamlama sırasında kullanılır. 4. Doğru değil. Test ilerleme raporları en çok test planlama sırasında değil, test izleme ve kontrol ile test tamamlama sırasında kullanılır. | FL-5.3.2 | K2 | 1 |
| 37 | d | 1. Doğru değil. Bir kullanıcı bir yazılım arızası bildirdiğinde, commit edilen kodların benzersiz tanımlanması sayesinde, kullanıcı tarafından kullanılan yazılım sürümündeki dosyaları (ve test komut dosyalarının ilgili sürümlerini) yeniden birleştirmek ve böylece hatayı yeniden oluşturmak ve hatayı daha hızlı bulmak mümkündür. 2. Doğru değil. Test ortamında yapılan bir değişiklik test sırasında beklenmedik sorunlara neden olursa, yapılandırma yönetimi test uzmanlarının ortamın önceki bir sürümüne geri dönmesine olanak tanır. Bu, testin değişiklikten etkilenmeden devam edebilmesini sağlar. 3. Doğru değil. Yapılandırma yönetimi, tanımlanan tüm belgelere (örneğin, gereksinim spesifikasyonları) ve yazılım öğelerine test dokümantasyonunda (örneğin, test planları) açık bir şekilde atıfta bulunulmasını sağlar. 4. Bu doğru. Bu, yapılandırma yönetimi süreci ile değil, hata yönetimi ile sağlanır. | FL-5.4.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 38 | b | 1. Doğru değil. Bu önemlidir, ancak test ortamı öğeleri kadar önemli değildir. 2. Bu doğru. Eksik olan önemli şey, test için kullanılan tarayıcının ve cihazın tanımlanmasıdır. Tarayıcı ve cihaz bilgileri önemlidir, çünkü böyle bir kusur tarayıcıya veya cihaza özgü olabilir. Örneğin, bir oturum açma düğmesi bir tarayıcıda (veya belirli bir tarayıcının bir sürümünde) iyi çalışabilir, ancak başka bir tarayıcıda çalışmayabilir. Bu nedenle, tarayıcı ve cihaz bilgileri geliştiricilerin sorunu yeniden oluşturmasına ve sorunun kök nedenini daha hızlı bulmasına yardımcı olabilir 3. Doğru değil. Test nesnesi tanımlanır (WebShop v0.99) 4. Doğru değil. Etki dahil edilir – bu şiddettir (yüksek) | FL-5.5.1 | K3 | 1 |
| 39 | d | 1. Doğru değil. Test yürütme ve kapsam araçları, test durumlarının otomatik olarak yürütülmesini ve bu test durumlarının yürütülmesiyle elde edilen kapsamın ölçülmesini kolaylaştırır. Bununla birlikte, bu araçlar kusurların düzenlenmesine ve yapılandırma yönetimine yardımcı olmaz. 2. Doğru değil. Test tasarımı ve uygulama araçları, test durumlarının, test verilerinin ve test prosedürlerinin oluşturulmasını kolaylaştırır, ancak kusurların düzenlenmesine ve yapılandırma yönetimine yardımcı olmazlar. 3. Doğru değil. Kusur yönetimi araçları kusurları yönetmek için kullanılır, ancak test araçları değildir ve test durumlarını veya yapılandırma yönetimini organize etmek için kullanılmaz. 4. Bu doğru. Test yönetimi araçları, yazılım geliştirme yaşam döngüsünün yönetimini kolaylaştırarak test süreci verimliliğini artırır.   (SDLC), gereksinimler, testler, kusurlar ve yapılandırma yönetimi | FL-6.1.1 | K2 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soru Numarası  (#) | Doğru Yanıt | Açıklama / Gerekçe | Öğrenme Hedefi  (ÖH) | K - Seviyesi | Puan |
| 40 | d | 1. Doğru değil. ‘Test temeline erişmeden test durumları oluşturma yeteneği’ mümkün değildir. Test uzmanları veya araçlar tarafından test durumlarının oluşturulması için test temeline erişim gerekir. 2. Doğru değil. "Daha objektif değerlendirme yoluyla daha fazla kapsam elde edilmesi" test otomasyonunun doğrudan bir faydası değildir. Test otomasyonu, kapsamın daha objektif bir şekilde değerlendirilmesini sağlayacaktır, ancak bu objektif değerlendirme kapsamı artırmayacaktır. Kapsamın sonuçlarını yalnızca daha fazla test durumu yazmak için kullanarak kapsam artırılabilir. 3. Doğru değil. "Daha yüksek işlem gücü ile test yürütme sürelerindeki artış" çelişkili bir ifadedir çünkü daha yüksek işlem gücü normalde yürütme sürelerini azaltacaktır ve test daha uzun süreceği için yürütme sürelerinin artması bir fayda değildir. 4. Bu doğru. Test otomasyonu insan hatalarından olumsuz etkilenemeyeceği için, daha fazla tutarlılık ve tekrarlanabilirlik yoluyla insan hatalarının önlenmesi test otomasyonunun bir avantajıdır. Örneğin, bu durum testlerin gereksinimlerden tutarlı bir şekilde türetilmesi, test verilerinin sistematik bir şekilde oluşturulması ve testlerin bir araç tarafından aynı sırada ve aynı sıklıkta yürütülmesi anlamına gelir. | FL-6.2.1 | K1 | 1 |