1- Regression test nedir? Kısaca açıklayınız.

Yeni kod değişikliklerinin mevcut işlevler üzerinde yan etkileri olmamasını sağlamak için yapılır. En son kod değişiklikleri yapıldığında eski kodun çalışmaya devam etmesini sağlar.

2- A/B nedir? Kısaca açıklayınız.

A/B testi, temelde bir sayfanın iki veya daha fazla varyantının kullanıcılara rastgele gösterildiği ve belirli bir dönüşüm hedefi için hangi varyasyonun daha iyi performans gösterdiğini belirlemek için istatistiksel analizin kullanıldığı bir denemedir. Yapılan değişikliklerin etkisi hakkında veri toplamamıza olanak tanır.

3- Black box / White box test kavramlarını açıklayınız?

Black box (Kara kutu) testi ettiğiniz yazılımın nasıl programlandığı hakkında hiçbir fikriniz olmadığı zamandır**.** Bu tür testler genellikle programlama bilgisi çok az olan veya hiç olmayan test uzmanları tarafından yapılır. Aslında bu iyi bir şey çünkü normal bir son kullanıcının bakış açısına sahipler.

Avantajları:

1. Büyük kod segmentleri için verimli
2. Kod erişimi gerekli değil
3. Kullanıcının ve geliştiricinin bakış açıları arasındaki ayrım

Dezavantajları:

1. Test senaryolarının yalnızca bir kısmı gerçekleştirildiğinden sınırlı kapsam
2. Testçinin yazılımın dahili bileşenleri hakkındaki bilgi şansı nedeniyle verimsiz test
3. Test kullanıcısı uygulama hakkında sınırlı bilgiye sahip olduğundan kör kapsama alanı

White box, açık kutu testi olarak da bilinen beyaz kutu testi, kod hakkında fikir sahibi olduğunuzda ve/veya söz konusu yazılımın mimarisi hakkında genel bilgi sahibi olduğunuzda gerçekleşir. Düşük seviyeli testler arasında sayılır ve temel olarak entegrasyon ve birim testine odaklanır.

Avantajları:

1. Hataları ve sorunları bulmada etkili
2. Test edilen yazılımın dâhili bilgileri hakkında gerekli bilgi, kapsamlı testler için faydalıdır.
3. Gizli hataları bulmaya izin verir.
4. Programcıların iç gözlemi
5. Kodu optimize etmeye yardımcı olur.

Dezavantajları:

1. Uygulanmamış veya eksik özellikler bulamayabilir
2. Test edilen yazılımın iç yapısı hakkında yüksek düzeyde bilgi gerektirir
3. Kod erişimi gerektirir

Başarılı yazılım teslimi için hem beyaz hem de kara kutu testi gereklidir. Çoğu durumda kara kutu testi, özel test uzmanları tarafından yapılırken, beyaz kutu testi geliştiriciler tarafından yapılır. TDD, ATDD ve BDD süreçleri ve destekleyici araçların ortaya çıkması, hataların erken tespitine olanak tanır ve odağı QC'den (Kalite Kontrolü) QA'ya(Kalite Güvencesi) kaydırır.

4- Mutation test nedir? Kısaca açıklayınız?

Kod mutasyon testi olarak da bilinen mutasyon testi, bir yazılım test paketinin değişiklikleri algılayabilmesini sağlamak için testçilerin bir uygulamanın kaynak kodunun belirli bileşenlerini değiştirdiği bir beyaz kutu testi şeklidir. Yazılımda yapılan değişikliklerin programda hatalara yol açması amaçlanır. Mutasyon testi, paketin test etmeye devam edeceği uygulamaların değil, bir yazılım test paketinin kalitesini sağlamaya yöneliktir.

Mutasyon testi tipik olarak birim testleri yapmak için kullanılır. Amaç, yazılım testinin mutasyona uğramış tüm kodları tespit edebilmesidir. Değişiklikler (mutasyonlar olarak adlandırılır), mevcut bir kod satırında farklı bir değerde değişiklikler yapılarak uygulanabilir. Örneğin, bir ifade silinebilir veya çoğaltılabilir, doğru veya yanlış ifadeler değiştirilebilir veya diğer değişkenler değiştirilebilir.

5- Behavior Driven Development(BDD) nedir? Neyi amaçlamaktadır?

Davranışa dayalı geliştirme (BDD), bir uygulamanın belgelendiği ve bir kullanıcının onunla etkileşime girerken deneyimlemeyi beklediği davranışa göre tasarlandığı Çevik bir yazılım geliştirme metodolojisidir. BDD, geliştiricileri bir uygulamanın veya programın yalnızca istenen davranışlarına odaklanmaya teşvik ederek şişkinlik, aşırı kod, gereksiz özellikler veya odak eksikliğinin önlenmesine yardımcı olur. Bu metodoloji, teste dayalı geliştirme (TDD) ve kabul testinde kullanılan uygulamaları birleştirir, artırır ve iyileştirir.

Davranış odaklı geliştirme kullanan tipik bir proje, bir ürünün nasıl çalışması gerektiğine dair genel bir resim oluşturmak için geliştiriciler, yöneticiler ve müşteri arasında bir konuşma ile başlar. Ürünün davranışına ilişkin beklentiler daha sonra geliştiriciler için hedef olarak belirlenir ve tüm davranış testleri geçtikten sonra ürün gereksinimlerini karşılar ve müşteriye teslime hazır hale gelir.

6- Agile test quadrant nedir? Quadrant’ların kapsamını açıklayınız?

Çevik testin , geliştirme süreci boyunca test faaliyetlerini düzenli olarak yürütmek için dinamik ve üretken bir test türü olduğunu biliyoruz. Test kullanıcıları, geliştirme süreci ile birlikte kısa bir yineleme süresinde çevik test gerçekleştirmeye yönelik prosedürler veya yaklaşımla ilgili olarak genellikle belirsizlik içinde kalır:

Her türlü test faaliyeti nasıl yapılır?

Ne zaman, ne tür bir test, hangi anda yürütülmeli?

Kim hangi aşamada test yapacak?

Çevik testle ilgili tüm bu tür soruların çözümü, çevik test kadranları kavramı aracılığıyla keşfedilebilir.

Çevik test kadranları, çevik manifestoya uyacak şekilde her farklı seviyede gerçekleştirilecek test tiplerini düzenlemek için tüm çevik test metodolojisini dört çeyreğe bölen Brain Marick tarafından ana hatlarıyla belirtilen bir araç veya kılavuz olarak kabul edilebilir.

Çevik Test Birinci Çeyrek (Q1):

Bu kadran, teknolojilerin ve otomasyonun uygulanmasıyla birlikte geliştiricilerin katılımını, yani ekibi desteklemek için teknoloji odaklı test senaryolarını gören kodun ve bileşenlerin kalitesiyle ilgilidir ve sistemin birim testi ve bileşen testini kapsar. Birim seviyesi.

Çevik Test İkinci Çeyrek (Q2):

İkinci çeyrekte, ekibi desteklemek için iş odaklı test senaryolarının uygulandığını görebiliriz. Temel olarak, bu çeyreğin ana odak noktası iş gereksinimleridir. Genel olarak, hem manuel hem de otomatik testlerin kullanımını içerir ve fonksiyonel testler, hikaye testleri, prototipler ve simülasyonlar, sistem düzeyinde yapılan ikili testler gibi test türlerinden oluşur.

Çevik Test Üçüncü Çeyrek (Q3):

Üçüncü çeyrekte, gerçek kullanıcı deneyimi ile birlikte önceki test aşamasının veya çeyreklerin geri bildirimleri ve incelemeleri, birden fazla senaryo kullanarak yazılım ürününü test etmek için bir itici güç olarak çalışır. Manuel test yöntemi, yazılım uygulamasını kullanıcı gereksinimlerine ve bir testçinin mantıksal düşüncesine dayalı olarak değerlendirmek için kullanılmaktadır. Bu kadran, ürünün kullanılabilirliğini oluşturmak ve güven kazanmak için kullanılır.

Bu çeyrekte kapsanan test türleri, keşif testi , kullanılabilirlik testi, kullanıcı kabul testi , alfa ve beta testidir .

Çevik Test Dördüncü Çeyrek (S4):

Bu kadran, güvenilirlik, güvenlik, uyumluluk, bakım kolaylığı, birlikte çalışabilirlik, kurtarma vb. gibi **işlevsel olmayan gereksinimleri** yerine getirmek için yazılım ürününü operasyonel bir kabul düzeyinde değerlendirme sürecini otomatikleştirmek için teknolojinin, yani araçların kullanımını içerir.

Bu çeyrekte kapsanan test türleri, performans testi, yük testi, stres testi, sürdürülebilirlik testi, güvenlik testi, güvenilirlik testi, kurtarma testi ve yazılım ürününün işlevsel olmayan gereksinimini doğrulamak ve doğrulamak için kullanılan diğer test türlerini içerir.

4. çeyrekte yaygın olarak kullanılan araçlardan bazıları Jconsole, Loadrunner, JMeter vb.'dir.