



## TERİMLER SÖZLÜĞÜ

Eğitim Adı	Yazılım Geliştirme Araçları
<b>İngilizce</b>	<b>Açıklaması</b>
<b>Acceptance Criteria (Kabul Kriterleri)</b>	Bir kullanıcı hikayesinin başarıyla tamamlanması için karşılanması gereken koşullar.
<b>Agile Manifesto</b>	Agile yazılım geliştirme yöntemlerinin temel ilkelerini tanımlayan bir metin.
<b>Algoritma</b>	Bir problemi çözmek veya belirli bir işlemi gerçekleştirmek için tasarlanan adım adım talimatlar kümesi.
<b>API (Application Programming Interface)</b>	Uygulamalar arasında iletişim ve veri alışverişini sağlayan arayüz.
<b>Artifact</b>	Build sürecinde oluşturulan çıktı dosyaları. Örneğin, derlenmiş kod veya paketlenmiş uygulama dosyası.
<b>AWS</b>	Amazon Web Services
<b>Backlog</b>	Projeye ait yapılacak işlerin listesi.
<b>Branch (Dal)</b>	Ana kod tabanından ayrı olarak geliştirme yapmak için oluşturulan paralel bir kopya. Farklı özellikler veya düzeltmeler için kullanılır.
<b>Build</b>	Bir yazılımı çalıştırılabilir hale getirmek için gerekli derleme, bağlama ve paketleme işlemlerinin tümü.
<b>Build Tool (Derleme Aracı)</b>	Yazılım projelerini derlemek, test etmek ve çalıştırmak için kullanılan araçlar. Örneğin, Maven, Gradle.
<b>CI/CD (Continuous Integration/Continuous Deployment)</b>	Kod değişikliklerinin otomatik olarak test edilmesi ve uygulamaya alınması süreçlerini ifade eder.
<b>Clean Code</b>	Basit, okunabilir, anlaşılabilir ve bakım yapılabilir kod yazma ilkelerini ifade eder. Kodun işlevselliğini kaybetmeden düzenli bir şekilde yapılandırılması için rehber ilkeler sağlar.
<b>Commit</b>	Bir geliştiricinin yaptığı değişiklikleri versiyon kontrol sistemine kaydetme işlemi. Her commit, yapılan değişikliklerin bir özetini içerir.
<b>Compiler (Derleyici)</b>	Programlama dillerinde yazılan kodları makine diline çeviren araç.
<b>Configuration Management (Konfigürasyon Yönetimi)</b>	Yazılım ve donanım bileşenlerinin düzenlenmesi ve takip edilmesi süreci.
<b>Conflict (Çatışma)</b>	Aynı dosyada farklı değişiklikler yapıldığında ortaya çıkan uyumsuzluk durumu. Çatışmalar çözülmeden birleştirme tamamlanamaz.
<b>Containerization (Konteynerleştirme)</b>	Yazılımı ve bağımlılıklarını bir konteyner içine alarak taşınabilir ve uyumlu hale getirme yöntemi. Örneğin, Docker.
<b>Continuous Feedback (Sürekli Geri Bildirim)</b>	Yazılım geliştirme sürecinde sürekli olarak geri bildirim alınarak yapılan iyileştirme süreci.

<b>Continuous Deployment</b>	Yazılımın manuel müdahale olmaksızın otomatik olarak üretim ortamına alınması.
<b>Continuous Development (Sürekli Geliştirme)</b>	Yazılımın sürekli olarak iyileştirilmesi ve güncellenmesi süreci.
<b>Continuous Improvement (Sürekli İyileştirme)</b>	Proje süreçlerinin sürekli olarak geliştirilmesi ve optimize edilmesi için yapılan çalışmalar.
<b>Continuous Monitoring (Sürekli İzleme)</b>	CI/CD süreçleri sırasında sistemin performansını ve güvenliğini takip etme süreci.
<b>Data</b>	Bilgi toplama, depolama, işleme ve analiz etme süreçlerinde kullanılan ham bilgi.
<b>Data Science (Veri Bilimi)</b>	Verileri anlamlandırmak ve değerli bilgiler elde etmek için kullanılan disiplinler arası bir alan.
<b>Data Scientist (Veri Bilimci)</b>	Veri toplama, analiz etme ve bu verilerden anlam çıkarma konularında uzmanlaşmış kişi.
<b>Debugger (Hata Ayıklayıcı)</b>	Yazılım geliştirme sürecinde koddaki hataları bulup düzeltmek için kullanılan araç.
<b>Defect Tracking (Hata Takibi)</b>	Yazılımda bulunan hataların kaydedilmesi, izlenmesi ve çözülmesi için kullanılan süreç ve araçlar.
<b>Delivery (Teslimat)</b>	Yazılım geliştirme sürecinde, ürünün müşteriye veya üretim ortamına teslim edilmesini ifade eder. Bu süreç, genellikle testlerin tamamlanması, onayların alınması ve yazılımın kullanıma hazır hale getirilmesi aşamalarını içerir.
<b>Dependency (Bağımlılık)</b>	Bir yazılımın çalışması için ihtiyaç duyduğu diğer kütüphaneler veya bileşenler.
<b>Dependency Injection (Bağımlılık Enjeksiyonu)</b>	Yazılımın bileşenleri arasındaki gevşek bağılılığı sağlamak için uygulanan bir teknik.
<b>Deployment (Dağıtım)</b>	Yazılımın geliştirme ortamından üretim ortamına taşınması süreci.
<b>Design Pattern (Tasarım Deseni)</b>	Yazılımda sık karşılaşılan problemleri çözmek için kullanılan tekrar eden çözümler.
<b>Dokümantasyon Araçları</b>	Yazılım projeleri için kullanıcı kılavuzları, API dokümantasyonları gibi belgeler oluşturmak için kullanılan araçlar. Örneğin, Confluence.
<b>Dynamic Code Analysis (Dinamik Kod Analizi)</b>	Yazılımın çalıştırıldığı sırada, güvenlik açıklarını veya performans sorunlarını tespit etmek için yapılan kod analizi yöntemi.
<b>Dynamic Testing (Dinamik Test)</b>	Yazılımın çalıştırılarak, davranışlarının test edilmesi süreci.
<b>Editor</b>	Kod yazmak, düzenlemek ve biçimlendirmek için kullanılan araç. Örneğin, Visual Studio Code.
<b>End-to-End Testing (Uçtan Uca Test)</b>	Yazılımın bir bütün olarak tüm işlevlerinin doğru çalıştığını doğrulamak için yapılan testler.
<b>Feature Branch</b>	Yeni bir özelliği geliştirmek için oluşturulan özel bir dal.
<b>Fuzzing (Fuzz Testi)</b>	Yazılımın güvenlik açıklarını veya hatalarını tespit etmek için rastgele veya beklenmedik girdiler sağlayarak yapılan bir test yöntemi.
<b>Hotfix</b>	Acil bir problemi çözmek için hızlı bir şekilde yapılan düzeltme.
<b>IDE (Integrated Development Environment - Entegre Geliştirme Ortamı)</b>	Yazılım geliştirme işlerini kolaylaştırmak için kod yazma, hata ayıklama, test etme gibi özellikler sunan bir yazılım.
<b>Incremental Development (Artımlı Geliştirme)</b>	Yazılımın küçük ve yönetilebilir parçalar halinde geliştirilmesi ve bu parçaların tamamlandıkça birleştirilmesi yöntemi.
<b>Integration Testing (Entegrasyon Testi)</b>	Farklı bileşenlerin bir araya geldiğinde doğru çalışıp çalışmadığını kontrol eden test süreci.

<b>Kanban</b>	Proje yönetiminde kullanılan, şeffaflığı ve süreç optimizasyonunu sağlayan bir yaklaşım.
<b>Kanban Board</b>	Kanban yöntemini görselleştirmek için kullanılan bir araç. Genellikle yapılacaklar, devam eden işler ve tamamlanan işler sütunlarına ayrılır.
<b>Load Testing (Yük Testi)</b>	Yazılımın ağır yük altındaki performansını ölçmek için yapılan testler.
<b>Local Repo</b>	Geliştiricinin kendi bilgisayarında depoladığı yerel kod kopyası.
<b>Logging (Kayıt Tutma)</b>	Yazılımın çalışması sırasında oluşan olayların kaydedilmesi.
<b>Makine Öğrenimi</b>	Sistemlerin, verilerden öğrenerek kendini geliştirme yeteneği.
<b>Merge (Birleştirme)</b>	Farklı branch'lerde yapılan değişiklikleri bir araya getirme işlemi.
<b>Microservices Architecture (Mikroservis Mimarisi)</b>	Yazılımın küçük, bağımsız hizmetlere bölünerek geliştirildiği bir mimari yaklaşım.
<b>Modeling Tool (Modelleme Aracı)</b>	Yazılım sistemlerini görselleştirmek ve tasarım oluşturmak için kullanılan araçlar. Örneğin, UML araçları.
<b>Orchestration (Düzenleme)</b>	Farklı süreç ve hizmetlerin bir araya getirilerek yönetilmesi. Örneğin, Kubernetes.
<b>Package Manager</b>	Yazılımın bağımlılıklarını yönetmek ve kurulumları kolaylaştırmak için kullanılan araç.
<b>Pipeline</b>	Yazılımı geliştirme, test etme ve yayınlama süreçlerini otomatikleştiren bir akış.
<b>Product Owner</b>	Scrum takımı içinde ürün vizyonunu tanımlayan, iş önceliklerini belirleyen ve ürün değerini maksimize etmeye odaklanan kişi.
<b>Proje Yönetim Araçları</b>	Takımların iş akışını organize etmek, zamanı yönetmek ve projeyi izlemek için kullandığı yazılımlar. Örneğin, Jira, Trello.
<b>Prototype (Prototip)</b>	Yazılımın son halinden önceki test edilebilir veya incelenebilir bir versiyonu.
<b>Quality Assurance (Kalite Güvencesi)</b>	Yazılım geliştirme sürecinde, ürünün belirli kalite standartlarını karşıladığından emin olmak için yapılan süreçler bütünü.
<b>Refactoring (Kod Yeniden Düzenleme)</b>	Yazılımın işlevselliğini değiştirmeden kodun okunabilirliğini ve bakımını iyileştirmek için yapılan düzenlemeler.
<b>Regression Testing (Gerileme Testi)</b>	Yeni kod değişikliklerinin mevcut işlevleri olumsuz etkilemediğinden emin olmak için yapılan test.
<b>Release Branch</b>	Yeni bir sürüm oluşturmak için kullanılan ve ana koddan ayrılan dal.
<b>Release Management (Sürüm Yönetimi)</b>	Yazılımın yeni sürümlerinin planlanması, geliştirilmesi ve dağıtılması süreçleri.
<b>Release Pipeline</b>	Yazılım geliştirme sürecinde, yazılımın geliştirme ortamından üretim ortamına taşınmasına kadar olan tüm adımları otomatikleştiren bir akış.
<b>Repository (Depo)</b>	Tüm dosyaların ve bunların değişiklik geçmişinin saklandığı bir veri yapısıdır. Versiyon kontrol sistemlerinde temel bir bileşen olarak kullanılır.
<b>Rollback</b>	Hatalı bir sürümün önceki, çalışan bir sürüme geri döndürülmesi işlemi.
<b>Runner (Çalıştırıcı)</b>	Testlerin çalıştırılmasını ve sonuçların raporlanmasını sağlayan bir araç.
<b>Scrum</b>	Agile yazılım geliştirme yaklaşımının bir uygulaması, framework. İterasyonlarla ilerleyen ve takım çalışmasını teşvik eden bir süreçtir.
<b>Scalability Testing (Ölçeklenebilirlik Testi)</b>	Yazılımın artan kullanıcı sayısı veya işlem yükü altında performansını değerlendirme süreci.

<b>Security Testing (Güvenlik Testi)</b>	Yazılımın güvenlik açıklarını tespit etmek ve önlemek için yapılan testler.
<b>Semantic Versioning (Semantik Versiyonlama)</b>	Versiyon numaralarını belirlemek için kullanılan bir standart. Örneğin, "MAJOR.MINOR.PATCH" formatı kullanılır.
<b>Shift-Left</b>	Yazılım geliştirme sürecinde kalite ve test faaliyetlerini mümkün olduğunca erken aşamalara kaydırma yaklaşımıdır.
<b>Smoke Testing</b>	Bir yazılımın temel işlevlerinin doğru çalışıp çalışmadığını kontrol etmek için yapılan yüzeysel testler.
<b>Sprint</b>	Scrum metodolojisinde belirli bir zaman aralığında tamamlanması planlanan işler bütünü.
<b>Static Code Analysis (Statik Kod Analizi)</b>	Kodun çalıştırılmadan önce analiz edilerek hataların ve güvenlik açıklarının tespit edilmesi süreci.
<b>Tag (Etiket)</b>	Bir depo içindeki belirli bir durumu veya sürümü işaretlemek için kullanılan bir etiket.
<b>Test Automation Framework</b>	Testlerin otomatikleştirilmesi için kullanılan araç ve yöntemlerin birleşimi.
<b>Test Coverage (Test Kapsamı)</b>	Test edilen kodun toplam kod tabanı içindeki oranı. Yüksek test kapsamı, daha iyi bir test kalitesine işaret eder.
<b>Test-Driven Development (TDD)</b>	Yazılım geliştirme sürecine testlerin yazılması ile başlanılan bir yöntem. Her yeni özellik, önce testler yazılarak doğrulanır, ardından bu testleri karşılayan kod geliştirilir.
<b>Test Suite</b>	Yazılımın farklı alanlarını test etmek için oluşturulan bir grup test. Her test, belirli bir işlevin veya özelliğin doğru çalıştığını kontrol etmek için tasarlanmıştır.
<b>Workflow (İş Akışı)</b>	Yazılım geliştirme veya proje yönetiminde görevlerin ve süreçlerin nasıl organize edildiğini tanımlayan yöntem.