

TERİMLER SÖZLÜĞÜ

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

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

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

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