

DEVOPS Nedir?



Muhammet Ali Kaya

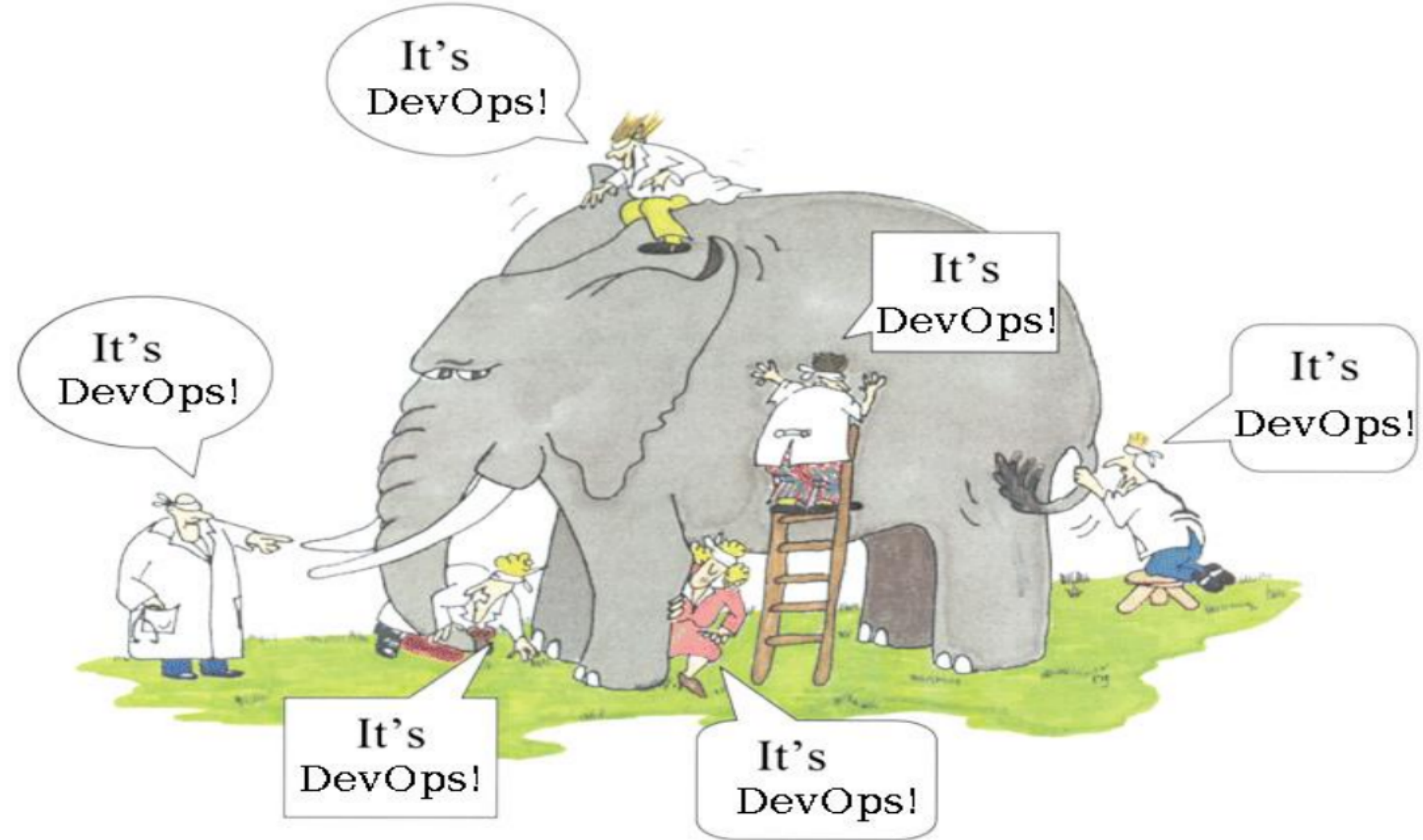


İçerik

- 01 *Devops Nedir?*
- 02 *DevOps'un Özellikleri Nelerdir?*
- 03 *DevOps'taki Etkili Uygulamalar Nelerdir?*
- 04 *DevOps İçin Kullanılan Araçlar Nelerdir?*
- 05 *DevOps Lifecycle*



DEVOPS NEDİR?



DEVOPS NEDİR?



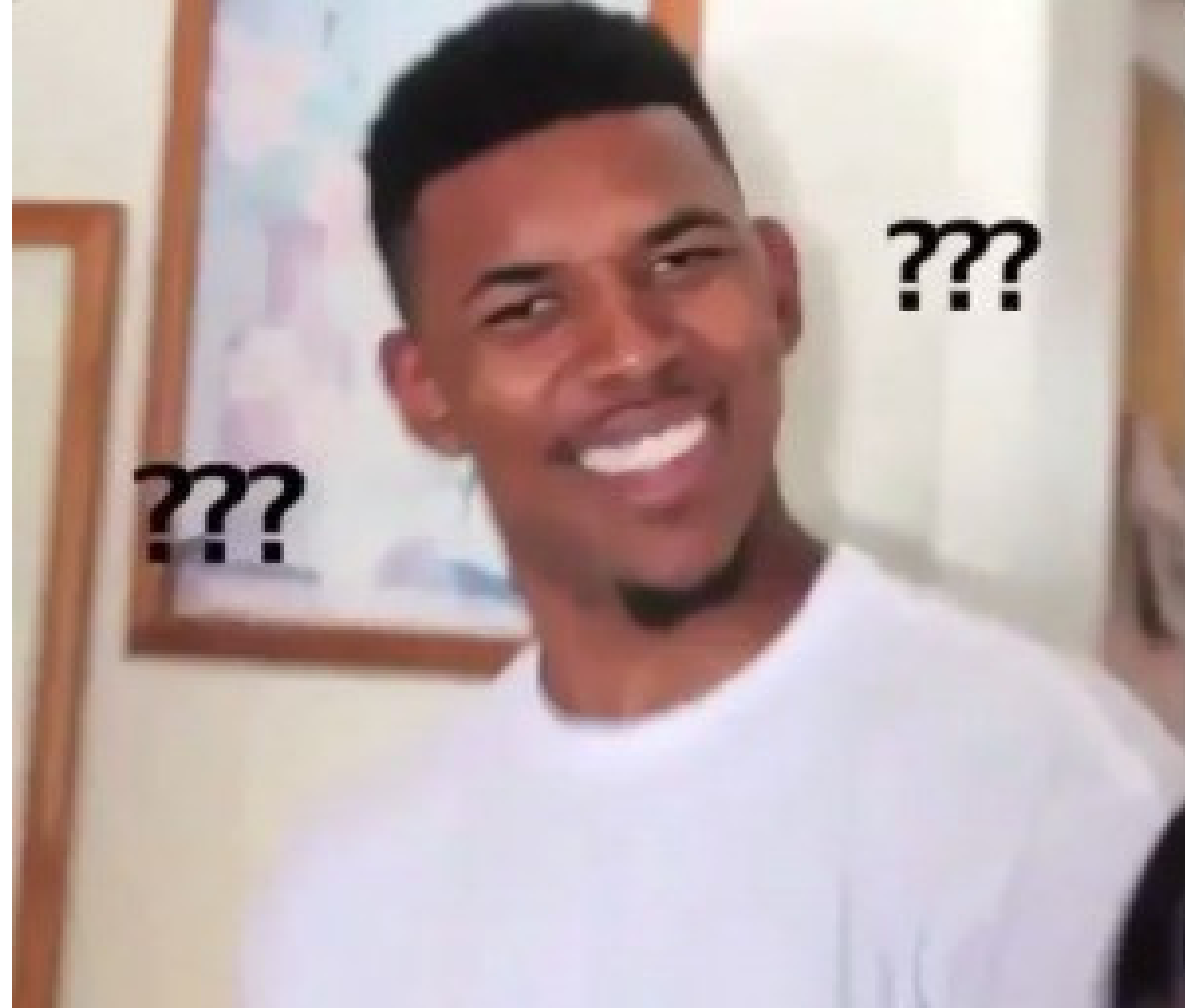
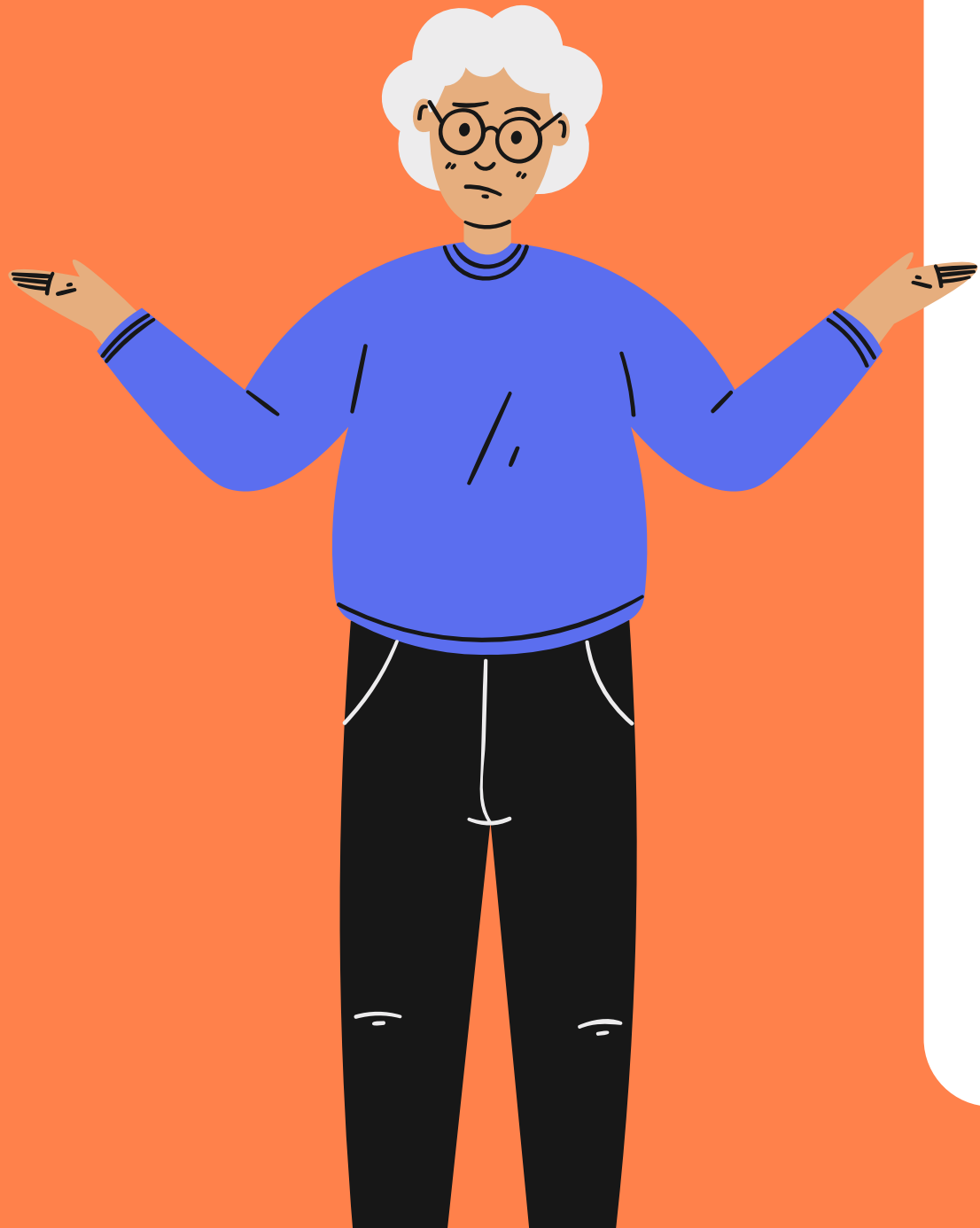
- Aslına bakıldığında Devops tek kelimeyle kendi tanımlamayı başaran bir kavramdır
- Devops Developer-Operations kelimelerin ilk harflerinin birleşmesiyle
- Daha iyi anlayabilmek adına Developer ve Operations ne olduğuna hakim olmamız gerekiyor.
- Developer birimi kodun yazıldı projenin oluşturulduğu kısımdır
- Operations birimi ise oluşturulan ürünün servisinden sorumlu olan birimdir

Neden devops?



- Developer birimi kodunu yazsın Operations birimi de servisi saęlasın ne gerek var "DEVOPS'A" ?
- Bu ařamada devopsun rolünü iyi kavramak önemli !
- Devops birimi bu iki birim arasındaki iletiřimi saęlamaktan sorumlu olan birimdir.

İLETİŞİM??



PEKİ İLETİŞİMLE NEYİ KAST EDİYORUZ?



- Devops olmayan bir yazılım işletmesi düşünelim
- Devops birimi olmasa da operasyon birimi ve developer birimlerimiz var.
- Yoğun emek ve çabalarının ardından Developerlar ortaya bir ürün koydu bütün özelliklerini düşündüler vs vs...
- Developerlar ürünü operasyon sürecine yollamadan önce kendi local cihazlarında çalıştırdılar.(Şimdilik her şey harika Şimdilik..)
- Daha sonra Operasyon birimine yolladılar operasyon birimi de aldıkları yapıyı platforma yüklemeye çalıştılar.

- Operasyon birimi kodları platformlara yükledi her şey harika fakat bir baktılar ki yükledikleri kod çalışmıyor...
- Tabi ki akıllarına ilk ne geldi "KOD BOZUK" demek geldi developer bölümüne geri bildirimler uğraşlar ve uzayan bir süreç.
- Developer birimi dedi biz localde çalıştırıyoruz açtı bilgisayarı gösteriyo diyo siz yükleyememişsiniz vs vs..
- Bu sıkıntılı süreci bu tarz açmazlarla uzatmak bizim seçimimiz fakat bu sıkıntıları çözmek için devops hizmeti almak da bizim seçimimiz...



HOŞGELDİN DEVOPS BEBEK



- Peki patronu bulduk anlattık dedik bize bi DEVOPS'cu lazım. Patron da dedi ki tamam alın size DEVOPS'cu. (Siz yine de patronunuzun işine pek karışmayın ;))
- Peki DEVOPS mühendisimiz geldi senaryo nasıl değişecek bize neler kazandıracak sürece ne gibi faydaları olucak.

- DEVOPS mühendisimiz öncelikle belli standartlar getirecek bu standartlar operasyon birimimizi de developer birimimizi de etkileyecek ve kapsayacak.
- Bu standartlar sayesinde iki birimimizin iletişimindeki problemler minimize ediliyor.
- Developer birimi uygulama geliştirmeye odaklandı.
- Operasyon birimi servisi platformda sunmaya odaklandı.
- Devops arada oluşacak problem ve pürüzleri gerekli araçlar metodlar ve yaklaşımlarla çözümlendi.
- Verim arttı.



DevOps'un Özellikleri Nelerdir?



- **Hız:** DevOps uygulamaları, daha hızlı inovasyon yapmak, uyum sağlamak için ihtiyaç duyduğunuz hızda hareket etmenize olanak tanır.
- **Hızlı Teslimat:** Sürümlerin hızını artırdığınızda, ürününüzü daha hızlı geliştirebilir ve rekabet avantajı elde edebilirsiniz.
- **Güvenilirlik:** Sürekli entegrasyon ve sürekli teslimat gibi DevOps uygulamaları, uygulama güncellemelerinin ve altyapı değişikliklerinin kalitesini garanti edebilir, böylece son kullanıcılar için optimum deneyimi korurken daha hızlı bir şekilde güvenilir bir şekilde teslimat yapabilirsiniz.

DevOps'taki Etkili Uygulamalar Nelerdir?



DevOps'taki Etkili Uygulamalar Nelerdir?

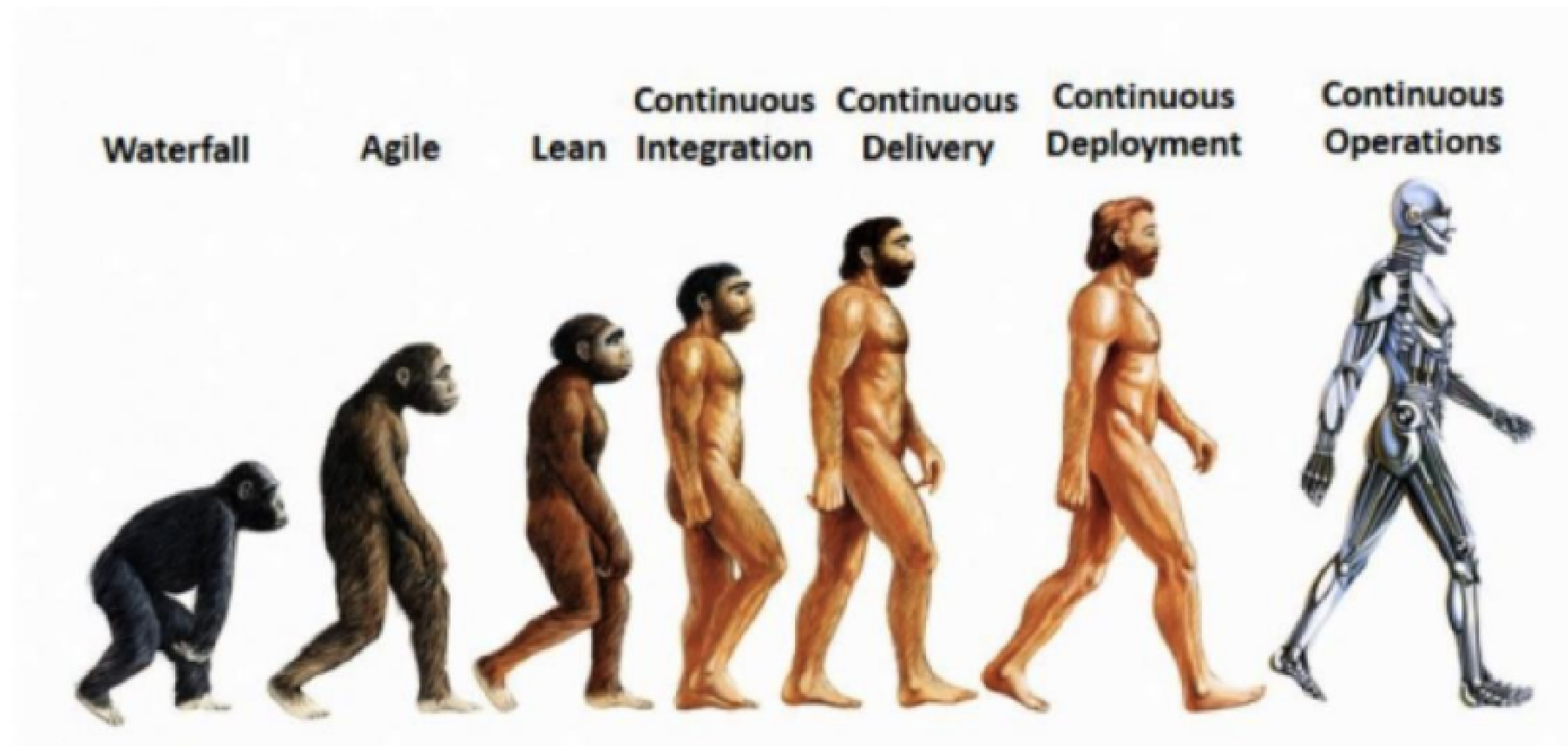


- **Otomatik Test:** Otomatik regresyon testi, sorunları hemen düzeltmeye ve daha yüksek kalitede kod göndermeye yardımcı olur. Bu, DevOps'ta da işe yarar, çünkü operasyon personelinin en büyük ihtiyacı, gönderilen kodun belirli bir kalite standardını karşılamasıdır.
- **Sürekli Entegrasyon (CI):** Sürekli entegrasyon ile, güncellenen kod sürüm kontrol sistemine her kontrol edildiğinde kod test edilir ve analiz edilir. Bu, geliştiricilerin çok az riskle yüksek kaliteli bir çözüm oluşturmaya olanak tanıyan kod kusurları hakkında anında geri bildirim sağlar.

DevOps'taki Etkili Uygulamalar Nelerdir?



- **Sürekli Dağıtım (CD):** Sürekli teslimin amacı, paketlenmiş bir yapıyı üretim ortamına teslim etmektir. CD, dağıtım süreci de dahil olmak üzere tüm teslimat sürecini otomatikleştirir. CD sorumlulukları, altyapı sağlamayı, değişiklikleri yönetmeyi (biletleme), yapıtları dağıtmayı, bu değişiklikleri doğrulamayı ve izlemeyi ve herhangi bir sorun varsa bu değişikliklerin gerçekleşmemesini sağlamayı içerebilir. CD ise sürekli entegrasyonun bittiği yerde başlar. CD, uygulamaların seçilen altyapı ortamlarına teslimini otomatikleştirir.
- **Üretim desteği:** Üretim sorunlarının çözümüne dahil olan üçüncü ve son ekip olmalarına rağmen, bu oldukça yaygın bir olaydır ve onlara daha iyi çözümler tasarımlarına yardımcı olan üretim sorunları hakkında fikir verir.
- **Uygulama İzleme:** Bu, çözümlerin üretime girdikten sonra gerçek zamanlı olarak izlenmesi ve kaydedilmesi anlamına gelir. Bu bize çözümün güvenilirliğini artıran ve arızaları önleyen performans ölçütleri verir.





DevOps İin Kullanılan Aralar Nelerdir?



DevOps İin Kullanılan Aralar Nelerdir?



- **Scrum:** Scrum, bir takımın üyelerinin geliştirme ve kalite güvence projelerini hızlandırmak için birlikte nasıl alışması gerektiğini tanımlar.
- **Kanban:** Kanban, Toyota fabrika katında kazanılan verimliliklerden kaynaklanıyordu. Kanban, devam eden yazılım projesi durumunun (WIP) bir Kanban panosunda izlenmesini öngörür.
- **evik (Agile):** Daha önceki evik yazılım geliştirme yöntemleri, DevOps uygulamalarını ve araçlarını büyük ölçüde etkilemeye devam ediyor. Scrum ve Kanban dahil olmak üzere birçok DevOps yöntemi, evik programlama unsurlarını içerir. Bazı evik uygulamalar, deėişen ihtiyaçlara ve gereksinimlere daha fazla yanıt verme, gereksinimleri kullanıcı hikayeleri olarak belgeleme, günlük standuplar gerçekleştirme ve sürekli müşteri geri bildirimlerini dahil etme ile ilişkilidir. evik ayrıca uzun, geleneksel "şelale" geliştirme yöntemleri yerine daha kısa yazılım geliştirme yaşam döngüleri öngörür.



WATERFALL



AGILE



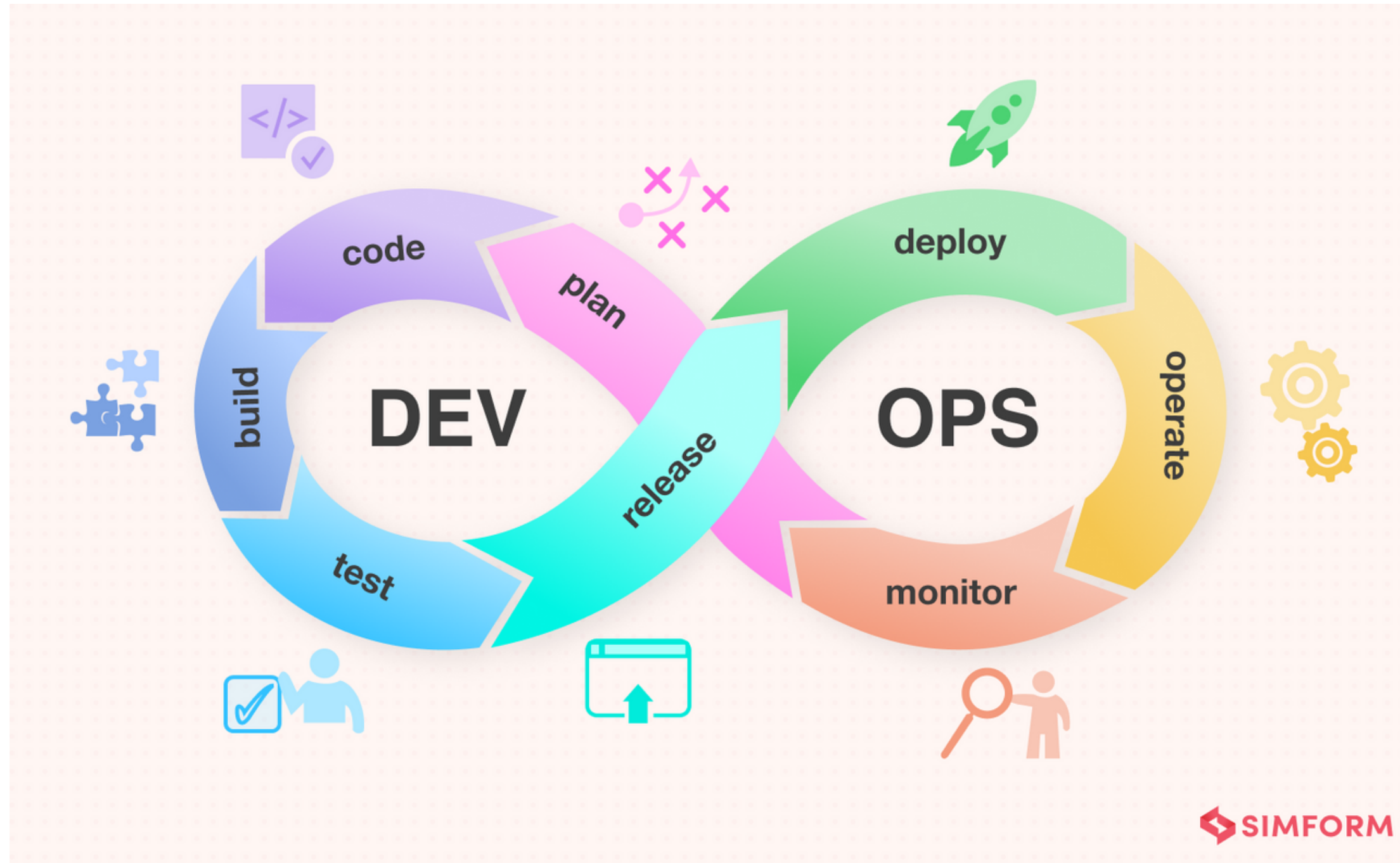
DEVOPS





DevOps Lifecycle





DevOps Lifecycle



- **Planlama:** Bu aşama, iş değerini ve gereksinimlerini tanımlamaya yardımcı olur. Örnek araçlar, bilinen sorunları izlemeye ve proje yönetimini gerçekleştirmeye yardımcı olan Jira veya Git'i içerir.
- **Kodlama:** Bu aşama, yazılım tasarımı ve yazılım kodunun oluşturulmasını içerir. Örnek araçlar arasında GitHub, GitLab, Bitbucket veya Stash bulunur.
- **İnşa Etme:** Bu aşamada, yazılım yapılarını ve sürümlerini yönetirsiniz ve gelecekteki üretim sürümleri için kod derlemeye ve paketlemeye yardımcı olmak için otomatik araçlar kullanırsınız. Ürün sürümü için gerekli altyapıyı "paketleyen" kaynak kodu havuzlarını veya paket depolarını kullanırsınız. Örnek araçlar arasında Docker, Ansible, Puppet, Chef, Gradle, Maven veya JFrog Artifactory bulunur.

DevOps Lifecycle



- **Test Etme:** Bu aşama, optimum kod kalitesini sağlamak için sürekli testi (manuel veya otomatik) içerir. Örnek araçlar arasında JUnit, Codeception, Selenium, Vagrant, TestNG veya BlazeMeter bulunur
- **Entegrasyon:** Bu aşama, ürün sürümlerinin üretime geçişini yönetmeye, koordine etmeye, programlamaya ve otomatikleştirmeye yardımcı olan araçları içerebilir. Örnek araçlar arasında Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker veya Jira bulunur.
- **Çalıştırma:** Bu aşama, üretim sırasında yazılımı yönetir. Örnek araçlar Ansible, Puppet, PowerShell, Chef, Salt veya Otter'ı içerir.
- **Takip Etme:** Bu aşama, üretimdeki belirli bir yazılım sürümünden gelen sorunlar hakkında bilgi tanımlamayı ve toplamayı içerir. Örnek araçlar arasında New Relic, Datadog, Grafana, Wireshark, Splunk, Nagios veya Slack bulunur.



Teşekkürler!

MUHAMMET ALİ KAYA