

Git Kullanımı

Git, yazılım geliştirme süreçlerinde ekip çalışmasını ve sürüm kontrolünü sağlayan güçlü bir versiyon kontrol sistemidir. Git sayesinde kodlarınızı güvenli bir şekilde yönetebilir, değişiklikleri takip edebilir ve projelerinizi etkin bir şekilde paylaşabilirsiniz.



Git'in Temel Kavramları

Depo (Repository)

Git'te projenizin tüm verilerinin saklandığı merkezi konumdur. Burada tüm dosyaların geçmişini ve değişikliklerini takip edebilirsiniz.

Dollar (Branches)

Projenizde farklı özellikler üzerinde aynı anda çalışabilmenizi sağlayan bağımsız geliştirme yollarıdır.

Commit

Yaptığınız değişiklikleri deponuza kaydetmek için kullanılan, değişikliklerin tarihçesini oluşturan işlemlerdir.

Pull & Push

Pull: Başka bir kullanıcının güncelleştirmelerini almak

Push: Yaptığınız değişiklikleri uzak depoya göndermek

Repository

Git repository yazılım projelerinizde tüm dosyalarınızı ve değişikliklerinizi sakladığınız bir depodur. Aynı zamanda repository, projenizin tarihçesini takip edebilmenizi ve farklı geliştirme ekipleri ile işbirliği yapabilmenizi sağlar.

Ayrıca Private ve Public seçenekler ile depolarınızı diğer kullanıcılarla paylaşabilir veya sadece sizin(veya ekibinizin) ulaşabileceğiniz şekilde saklayabilirsiniz.



Repository

Repositories Developers Spoken Language: Any ▾ Language: Any ▾ Date range: This month ▾

 apple / corenet CoreNet: A library for training deep neural networks Python 6,545 ⭐ 6,545 509 ⚡ 509 Built by 	 Star ▾ 6,603 stars this month
 myshell-ai / OpenVoice Instant voice cloning by MyShell. Python 25,831 ⭐ 25,831 2,425 ⚡ 2,425 Built by 	 Star ▾ 8,672 stars this month
 zyronon / douyin Vue3 + Pinia 仿抖音，Vue 在移动端的最佳实践 . Imitate TikTok , Vue Best practices on Mobile Vue 7,887 ⭐ 7,887 1,924 ⚡ 1,924 Built by 	 Star ▾ 4,926 stars this month
 wandb / openui OpenUI let's you describe UI using your imagination, then see it rendered live. HTML 14,754 ⭐ 14,754 1,246 ⚡ 1,246 Built by 	 Star ▾ 8,700 stars this month
 pagefaultgames / pokerogue A browser based Pokémon fangame heavily inspired by the roguelite genre. TypeScript 2,312 ⭐ 2,312 975 ⚡ 975 Built by 	 Star ▾ 1,653 stars this month

Commit

Commit, Git'in en temel kavramlarından biridir. Commit, yaptığınız değişiklikleri kalıcı olarak kaydetmenizi sağlar. Her commit, önceki durumla kıyaslanabilir bir anlık görüntü oluşturur.

Bir commit, yeni eklenen dosyaları, değiştirilen dosyaları ve silinen dosyaları içerir. Commit mesajı, bu değişikliklerin ne olduğunu açıklar.



Push & Pull



Push

Git Push işlemi ile yereldeki değişikliklerinizi uzak sunucudaki Git deposuna gönderebilirsiniz. Bu, başkalarının sizin yaptığınız değişiklikleri görebilmesini ve kullanabilmesini sağlar.



Pull

Git Pull işlemi ise uzak sunucudaki değişiklikleri yerel Git deposuna çeker. Bu, başkalarının yaptığı değişiklikleri güncellemenizi ve onları kullanabilmenizi sağlar.



Dallar (Branches)

Git'te, dallar ana kodun bağımsız sürümleridir. Yeni işlevler, hatalar veya deneysel değişiklikler için ayrı dallar oluşturulabilir. Bu şekilde, ana kod devamlılığı bozulmadan geliştirme yapılabilir.

Standart olarak ana bir branch(main) olur ve genellikle bu branch üzerinde geliştirme yapılır. Yeni bir dal (branch) oluşturulduğunda, bu dal üzerinde yapılacak değişiklikler ana branch'i etkilemez, böylece farklı geliştirme süreçleri aynı anda yürütülebilir. Bu, ekip çalışmasını ve projenin paralel olarak ilerlemesini kolaylaştırır.

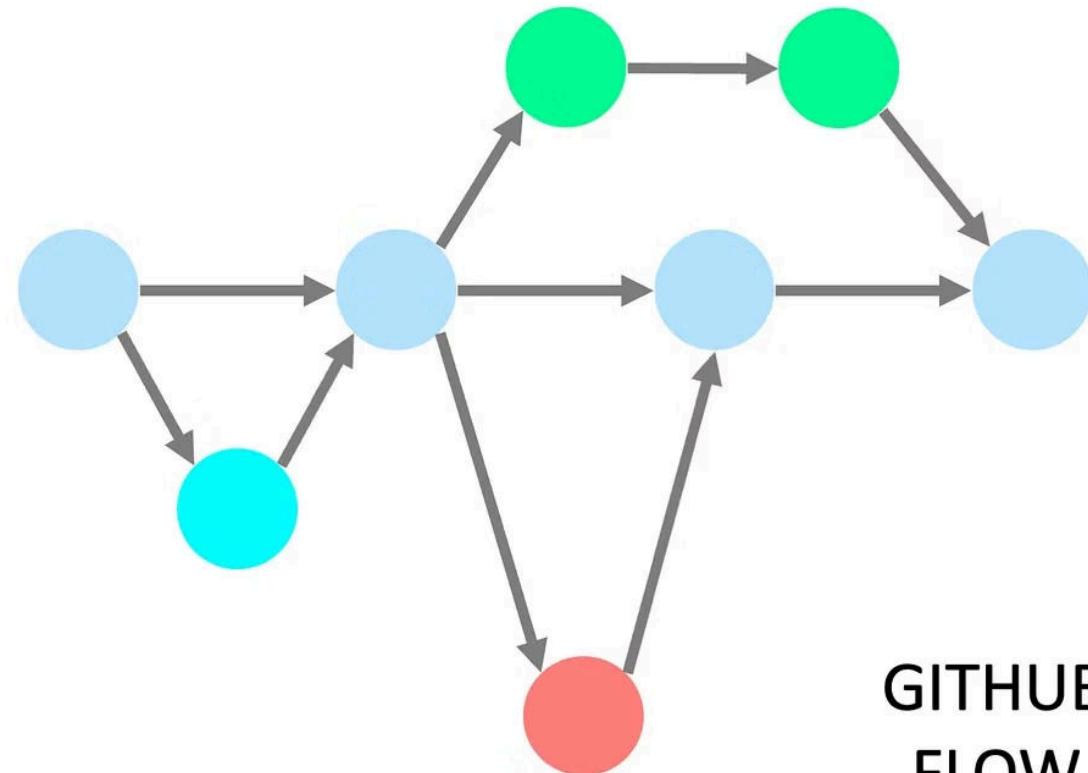
Dallar (Branches)

change

main

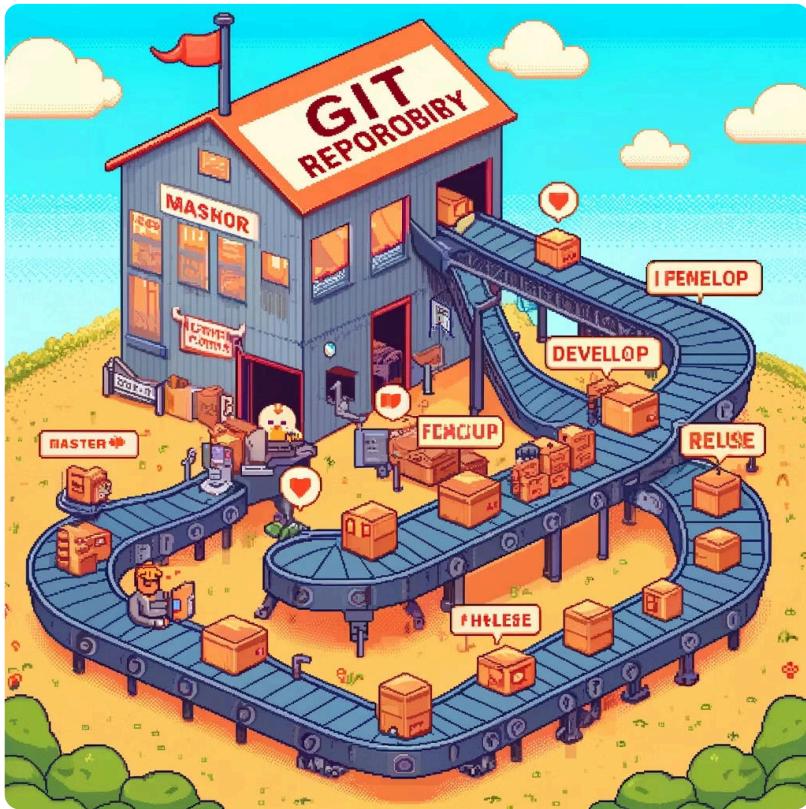
change

change



GITHUB
FLOW

Dallanma Stratejileri



GitFlow

Gitflow bir dallanma stratejisidir ve yazılım projesinin stabilitesini artırmak için geliştirilmiştir. Bu strateji, master branch üzerinde hep stabil olan sürümü erişimi sağlarken, development branch üzerinde ise yeni özelliklerin geliştirilip test edilmesine olanak verir. Feature branch'leri ise bu özelliklerin geliştirmesi için oluşan dallanmalar olarak kullanılır.



Trunk-based

Trunk-based yine bir dallanma stratejisidir ve sürekli entegrasyon (continuous integration) prensibine dayanır. Bu stratejiye göre, tüm geliştiriciler aynı ana dalı (trunk) kullanır ve her bir özellik için kısa süreli dallar (feature branch) açarlar. Bu şekilde, sürekli entegrasyon sağlanır ve günlük düzeyde kod paylaşımı kolaylaşır.

GitFlow



Gelişmiş Dallanma Modeli

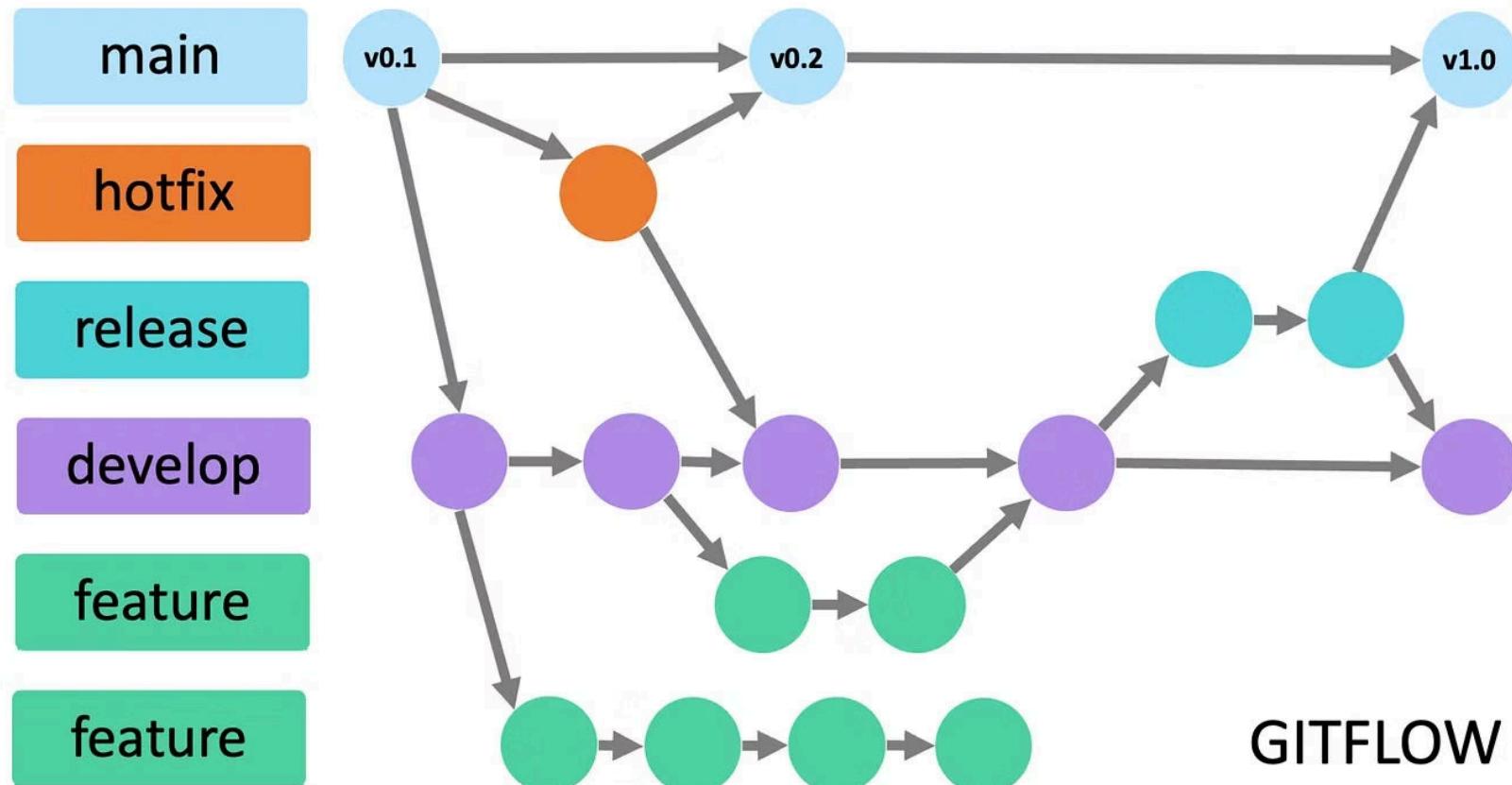
GitFlow, yazılım geliştirme sürecinde kullanılan bir dalma modelidir. Bu model, ana dal üzerinde değişiklik yapmanın riskli olabileceği durumlarda, çözümlerin daha güvenli bir şekilde test edilebilmesi için birçok dal kullanımını sağlar.



Ayrı Dallar, Güvenli Geliştirme

Projenin ana dalı olan "master" dalı, üretim için hazırlanmış sürümün yer aldığı dal olarak kullanılırken, "develop" dalı ise yeni özelliklerin eklenerek geliştirilmesi için kullanılır.

GitFlow



Trunk-Based Development



Sürekli Entegrasyon

Trunk-Based Development, tüm geliştirici değişikliklerinin hızlı bir şekilde ana dala entegre edilmesini sağlar.

Küçük Ekipler

Bu yöntem özellikle küçük ve orta ölçekli ekipler için uygun, hızlı geliştirme süreçlerini destekler.

DevOps Uyumlu

Trunk-Based Development, Agile ve DevOps pratiklerine uyumlu bir dallanma stratejisi sunar.

Trunk-Based Development

Bir projde trunk base yöntemini tercih edeceksek burada şunlara dikkat etmeliyiz

- Kısa Ömürlü Dallar:** Örneğin Bir robotik kedi tasarlıyor olsaydık, örnek bir dal ismi "robotik-cat-tail-servo-1" olabilir. Bu sadece 1 motorun kodunu barındırmamasına rağmen açılmış ve ana dal ile birleştirilmiş olurdu. tüm kuyruk oluşana kadar kod bekletilmezdi
- Sürekli Entegrasyon:** Dallarda yapılan değişiklıkların sürekli olarak ana dala entegre edilmesi önemlidir. Her bir değişiklik yapıldığında otomatik olarak test edilip, ana dala entegre edilmesi sağlanmalıdır. Bu sayede hataların erken tespit edilmesi ve düzeltilmesi mümkün olur.
- Takım İletişimi ve Koordinasyon:** Bu kadar bağımlı ve dinamik çalışılan Trunk-Based yapıda , takım iletişimini ve koordinasyonunu oldukça önemlidir. Herhangi bir hata ya da sorun olduğunda takım üyelerinin hızlıca iletişime geçip çözüm üretebilmeleri gerekmektedir. Bu da projenin başarısı için oldukça kritiktir.



Feature Toggle

Feature Toggle olarak adlandırdığımız kavram, yazılım geliştirme sürecinde belirli bir özelliğin kullanılabilirliğini kontrol etmek için kullanılan bir yöntemdir. Bu yöntem sayesinde, bir özelliği belirli kullanıcılara veya kullanım durumlarına göre etkinleştirmek veya devre dışı bırakmak mümkün olur. Ayrıca tamamlanmamış bir geliştirme eğer ki projenin ana dalına dahil edilmesi gerekirse, feature Toggle kullanılabilir. Bu sayede, tamamlanmamış bir özellik diğer kullanıcılara etkileşim vermeden güvenli bir şekilde test edilebilir ve geliştirilebilir.

Avantajları



GitFlow

Belirli dal türleriyle düzenli bir yapı sağlar, versiyonlama sürecini kolaylaştırır ve birden fazla özelliğin aynı anda geliştirilmesine olanak tanır. Üretimdeki acil hatalar da hızlıca düzeltilebilir.



Trunk-Based Development (TBD)

Tek bir ana dal kullanımıyla yönetim kolaylaşır, değişiklikler sürekli entegre edilir ve hatalar hızlıca tespit edilip düzelttilir. Kısa dal ömürleri sayesinde güncel kalmak da kolaydır.





Dezavantajları



GitFlow

Dal yönetimi küçük takımlar için karmaşık olabilir. Dal birleşmeleri zaman alabilir. Dal geçişleri ve yönetimi zaman alıcıdır.



Trunk-Based Development (TBD)

Geliştiricilerin sürekli iletişim halinde olması gereklidir. Hızlı entegrasyon, yeterince test edilmemiş kodların dahil edilme riskini artırır. Çok büyük projelerde tek dal üzerinde çalışmak zor olabilir.



Merge & Conflicts

Merge işlemi , farklı dallardaki değişikliklerin birleştirilmesi anlamına gelir. Birleştirme sırasında, farklı dallardaki değişiklikler çakışabilir(conflict) ve çözüm gerektirebilir. Bu çakışmalar, çeşitli araçlar ve teknikler kullanılarak çözülebilir. Çakışmaların başarılı bir şekilde çözülmesi, kod tabanının bütünlüğünü korurken farklı dallardaki çalışmaların birleştirilmesini sağlar.

Örnek Senaryo

**lameddinc/
it_ogrendim**

num sonrası alıştırma amaçlı açılmıştır.

1 Contributor 0 Issues 0 Stars 0 Forks



GitHub

GitHub - alameddinc/git_ogrendim: Sunum sonras...

Sunum sonrası alıştırma amaçlı açılmıştır. Contribute to alameddinc/git_ogrendim development by creating an...

Teşekkürler



by Alameddin Çelik