

Git Branch Yönetimi ve Yazılım Projelerinde Uyarlanması

İlk olarak, piyasada bulunan Gitflow, GitHub, GitLab gibi platformların branchlerine kısa bir göz gezdirip sonrasında Gitflow branch modelini kendimiz için nasıl uyarlayabiliriz onun üzerinde fikir yürütelim.



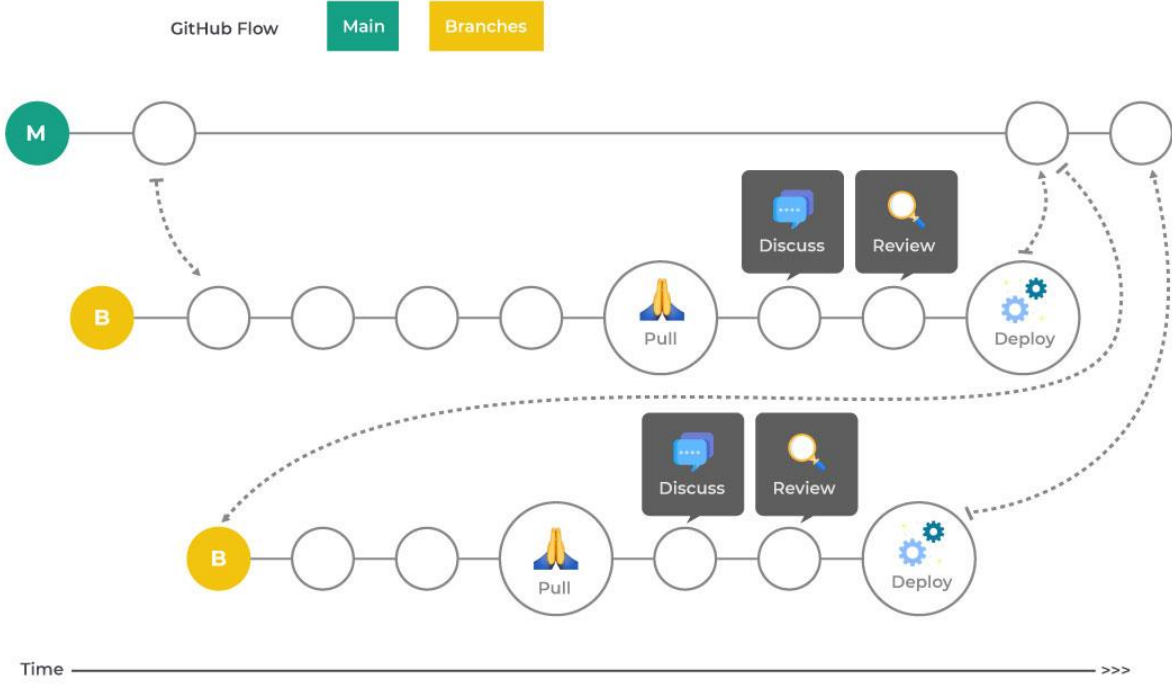
1. **Gitflow Branch Model:** Bu model, açık kaynak yazılım projeleri için geliştirilmiş bir branch modelidir. Bu model, projenin ana çalışma dalı olan “master” dalının yanı sıra “develop” ve “feature” dallarını içerir. Bu dallar arasındaki iletişim, “merge” işlemiyle sağlanır.
2. **GitHub Flow:** Bu model, projenin geliştirilmesi sırasında kolaylık sağlamayı amaçlar. Bu model, sadece iki dal kullanır: “master” dalı ve “feature” dalı. Bu dallar arasındaki iletişim, “pull request” ve “merge” işlemleriyle sağlanır.
3. **GitLab Branch Model:** GitLab, yazılım geliştirme projelerinde kullanılabilecek çeşitli branch modellerini destekler. Bu modellerden bazıları “production”, “pre-production” ve “master” branch modelleridir.

Piyasada kullanılan platformların branch modellerini görmekteyiz. Pekiyi kendi şirketimiz için nasıl bir branch modeli tasarlayabiliriz? Bu durumu aslında nitelik olarak proje bazlı ayırabiliriz. Nasıl diye soracak olursanız, her yazılım projesinin kapsamı farklıdır. Piyasayı baz alarak ve biraz üzerinde değişiklik yapacağımızı varsayarsak, şirketinizde yazılım projelerinizi üç ayrı başlık altında kategorize edelim. Bunlar Açık Kaynak Yazılım Projeleri, Özel Yazılım Projeleri ve Ölçeklenebilir Yazılım Projeleri

Ekibinizin 30’dan fazla üzerinde çalıştığı proje var diyelim bu projeleri kategorize ettikten sonra ilk aşamada;

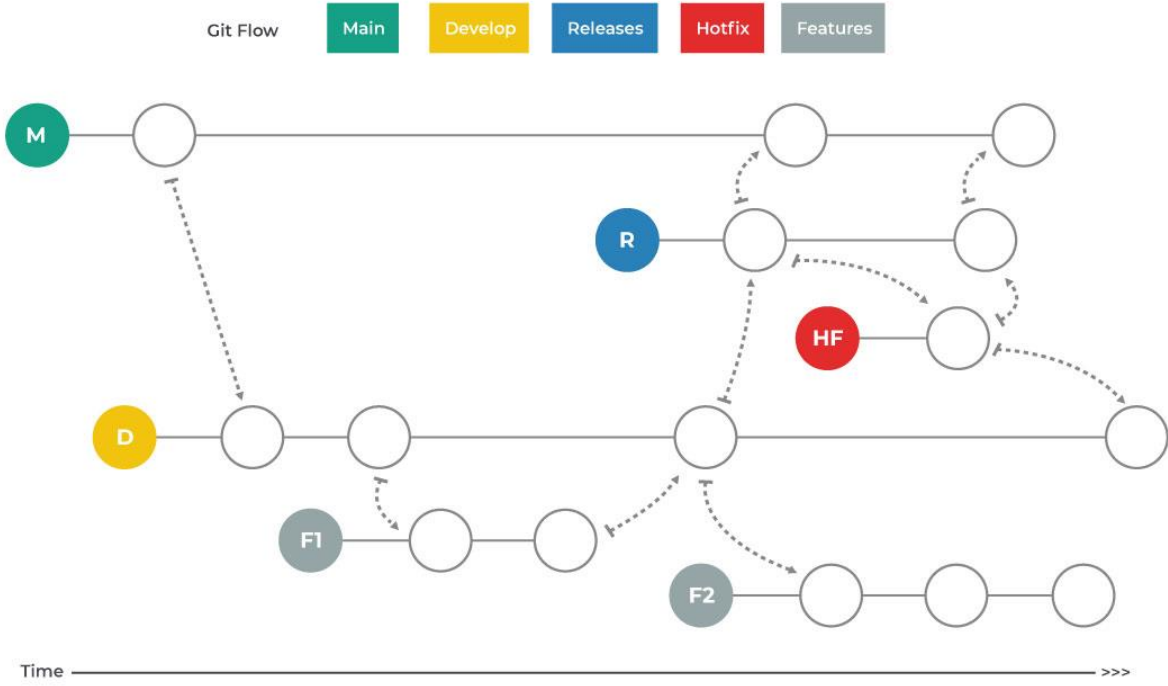
Açık Kaynak Yazılım Projeleri GitHub(main-feature) modeli, projelerinizin bazıları daha stabil bir aşamada ve özellikler nadiren ekleniyorsa, canlıya çıkmanız durumunda UAT gerektirmiyor ve sadece SonarQube gibi teknolojiler aracılığıyla statik kod analizi yaptırmak istiyorsanız, bunun için branch işlemlerini çoklandırmaya gerek yoktur. Bu durum için “master” branch modeli daha uygun olabilir. Bu model, tüm çalışmaların master branch üzerinde yürütülmesini ve bu çalışmaların sürekli olarak güncellenmesini sağlar. Aynı zamanda bu model ile Code Review süreci de daha kolay ve hızlı bir şekilde

gerçekleştirilebilir.



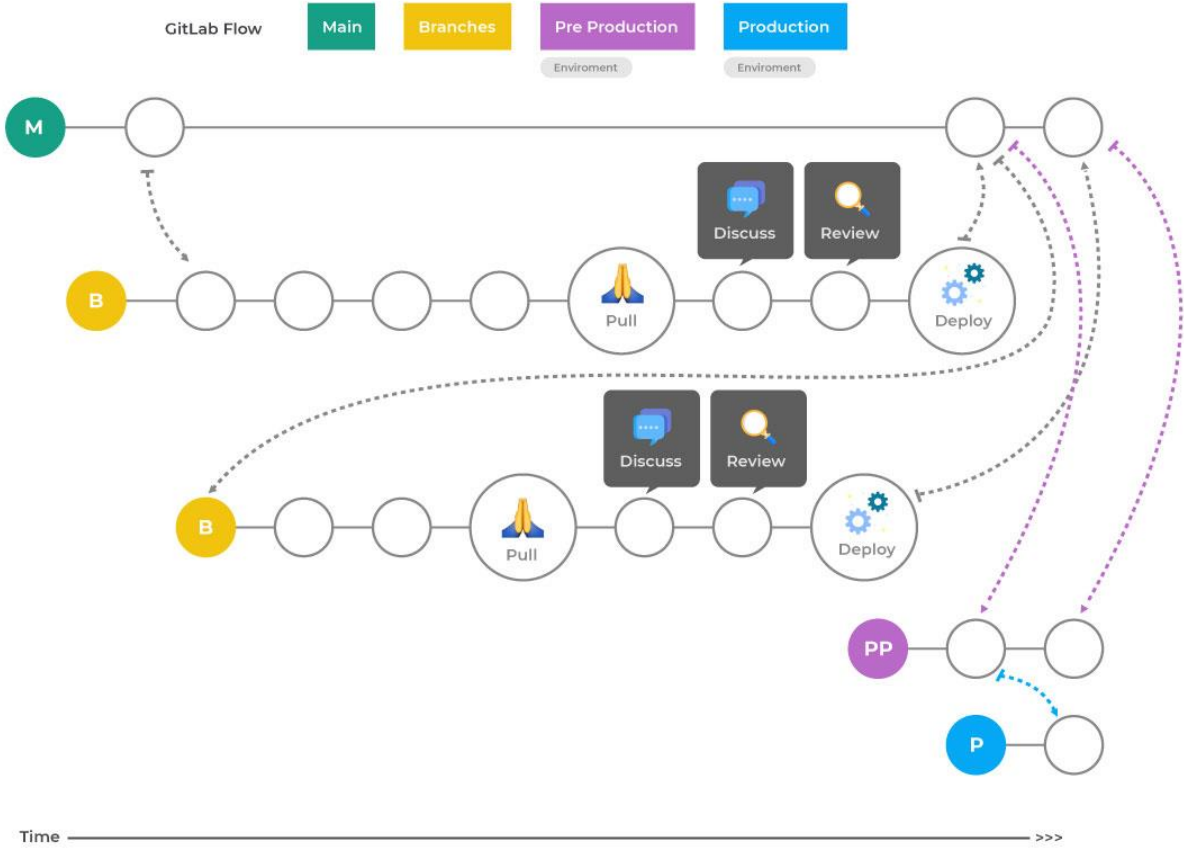
Özel Yazılım Projelerinizin kategorisinde bulunan projeler için Gitflow branch modeli daha sağlıklı görünmektedir. Feature üzerinde yaptığınız geliştirmeleri, develop'a merge etmeden önce Quality Assurance(QA) Selenium ve JMeter testlerini sonrada kod analizini bu feature üzerinden yaptıktan sonra merge işlemini yapabilirsiniz. UAT ve Unit testleri develop branchi üzerinden yaptıktan sonra önce release sonra master'a taşıyabilirsiniz. Feature, develop'a merge edilmeden Code Review yapılabilir bu

da sürecin yine hızlanmasını ve kolaylığını sağlar.



Ölçeklenebilir Yazılım Projeleri için GitLab-GitFlow (production,pre-production,master) branch modeli uyarlanabilir. Bunu detaylandırırsak misalen production altında açılan feature'ın geliştirmesi bitirilerek push edildikten sonra production'a merge edilmeden yine bu feature üzerinde Selenium, JMeter gibi performans testleri, statik kod analizinin ilgili tool aracılığıyla yapılması ve Code Review'ın kabul görmesi durumunda production'a merge edilmesi sağlanabilir. Hemen sonra yine production branchinde pre-production branchine merge edilerek, pre-production branchinde Unit testler yazılabilir ve UAT yine bu branchte yapılabilir. Son aşamada master'a çıkabilirsiniz. Code Review için biraz uzun bir süreç gibi

görünse de paralelde verimli olacağı düşünülmektedir.



Not: GitLab branch, Gitflow branch modeliyle birbirine yakın olduğu için harmanlanmış bir branch modeli olarak düşününebilir.

Geldiğimiz noktada faydalı olabileceğini düşündüğüm notlar;

Code Review'ın feature üzerinde sağlanması

Quality Assurance(QA) testlerin, statik kod analizinin feature'larda gerçekleştirilmesi

UAT'ların master'a çıkmadan hemen önce yapılması

Unit testler'in pre-prod, release ya da develop branchinde sağlanması

Yazılım yaşam döngüsündeki sürecimizin hızlandırılması ve kalitenin korunması için üstte yazdığımız makaleden faydalanabilir, geliştirebilir ve branch yönetimini şirketinize uygun birçok farklı durumda şekillendirebilirsiniz.

Faydalı olması dileğiyle.

Görsel Kaynaklar;

<https://www.virtusway.com/en/workflows-strategies-for-working-with-git.html>