

gitHub Akış Checklist

- ☐ 1. **Team Lead** proje için bi repo açacak. (Public olsa iyi olur) >
- ☐ 2. Bu reponun master branch'ını [Settings-Branched-Add Branch protection rules](#) yoluyla korumaya alacak (default seçenek ile) >
- ☐ 3. [Settings-Collaborator-Add people](#)'dan diğer takım arkadaşlarını projesine (reposuna) davet edecek. >
- ☐ 4. IntelliJ de [File-New project-From VCS](#) den Projeyi Local'ine çekecek. >
- ☐ 5. Master a sağ tık - kendi isminde "[New branch from master](#)" yapacak >
- ☐ 6. Yaptığı değişiklikleri [commit](#) ile etiketleyecek.
commit, push, pull(update), pull request ve merge kavramları:
 - 1. **commit**, Barkodlama, Etiketleme, Fotoğraflama - Albüm haline getirme,
 - 2. **push**, Fotoğraf Albümünü Kargolama,
 - 3. **pull(update)**, Fotoğraf Albümüne başkalarının eklediği fotoğraflarla birlikte albümü kargoyla geri almaya benzetilebilir.
 - 4. **pull request** ise, albüm sahibinden, "benim çektiğim fotoğrafları da albüme ekler misin?" diye izin almaya benzer.
 - 5. **merge** de benzer bir metaforla albüm sahibinin, albüme yeni fotoğraflar eklemesine benzetilebilir. >
- ☐ 7. Committediği değişiklikleri kendi reposuna [push](#) layacak (Fotoğraf albümünü sallıyorum isviçredeki github serverının içindeki harddiske yollayacak. Böylece uzak reposunda master a ilaveten kendi adında branch oluşacak) [direkt master da değişiklik yapmasının sakıncaları olabilir]. *Bu arada repo, master, branch... gibi kavramların, aslında içerisinde çeşitli dosyalar bulunan klasörlerden ibaret olduğunu aklımızın bi köşesinde tutalım. Sadece bir program vasıtasıyla (git) bu dosyalarda yaptığımız tüm değişiklikler takip edilir.>*
- ☐ 8. Bi nevi yedeğini de almış olduğu kendi ismindeki branch'ı master'a merge edecek (lead olduğu ve repo kendisine ait olduğu için protection rule karşısına çıkmayacak) >
- ☐ 9. Sonra intelliJ den, local master a geçip, bu sefer [gitHub](#) daki (... *İsviçre deki gitHub server ının içindeki harddiskte bulunan...*) değişikliği update ederek local master 'ına çekecek >
- ☐ 10. Master local 'in güncel halini kendi ismindeki dalına da merge edecek. >
- ☐ 11. **Takımın Her Üyesi** de lead'in davetini mailden veya github dan kabul edecek. Onun Discorddan paylaştığı repo adresini intelliJ [File-New Project from VCS](#) yoluyla kopyalayacak >

- ☐ 12. Kendi projelerinde **Local** ve **Remote** master oluşmuş olacak. Local master a sağ tık - **New branch from master** yoluyla kendi isimlerinde branch oluşturacaklar >

- ☐ 13. Yapacağınız değişiklikleri kendi package veya sınıflarınızda yaparsanız **conflict** ler minimize olacaktır >

- ☐ 14. **Her bir takım üyesi, IntelliJ yi açtıklarında kodlamaya başlamadan, ilk iş olarak remote master a checkout yapıp update ederek, uzak serverdaki master ı kendi local masterına çekip eşitleyecek / senkronize olacak... Hatta üyelerin (collaborator) bunu sık yapmaları iyi olur.** >

- ☐ 15. Locallerinde, Kendi dallarında Yaptıkları değişiklikleri **commit** (barkodlama/fotoğraflama) yapacaklar >

- ☐ 16. Bu değişiklikleri lead'in reposuna (...İsviçre'deki github serverının içindeki harddiske..) **push** layacaklar (1,2,3... Sayıdaki commit albümünü remote github server ına kargolayacaklar) >

- ☐ 17. Böylece lead'in uzak reposunda, üyelerin kendi isimlerinde branchları (klasörleri) oluşacak ve herkesin uzak repoda bi yedeği oluşmuş olacak (*gitHub, gitLab, BitBucket, SourceTree gibi platformları, google drive'ın kodlama ve proje yönetimi için özelleşmiş halleri gibi düşünebiliriz...*). >

- ☐ 18. Üyeler kendi dallarında yaptıkları değişiklikler için ister direkt **github** dan, ister **intelliJ-git-github>Create pull request** yoluyla, "yapılan değişikliğin kabul edilip, ana dala birleştirilmesi (**merge**)" için istek oluşturacaklar >

- ☐ 19. Lead bu isteği ister **review** yaparak, ister direkt kabul ederek ana dala aktaracak (**merge** edecek) >

- ☐ 20. **Böylece uzak repodaki master değişmiş olacak (her uzak repo masterdaki değişiklik discorddan ilan edilebilir, böylece herkesin haberi olur, aslında üyeler her kod yazımı öncesinde master ı update etseler buna gerek de yok)** >

- ☐ 21. Uzak repodaki master da meydana gelen değişikliği herkes **local master**'lerini **update** ederek kendi bilgisayarına (*mesela Ayşe kendi harddiskindeki **local/master** 'ına* çekti ya, şimdi de bu değişikliği, **ayşe-checkout** yoluyla kendi branch ına geçip, **master'a sağ tık-merge master in to ayşe** yaparak kendi **local branchlarına** çekecekler. >

- ☐ 22. Sonra kendi branchlarında kod yazmaya devam... **14-21.** adımlar sprint bitene kadar döngü halinde uygulanacak...
