2019.12.29 Supercharging your GitHub workflow: Plural Sight –

Known: Add, commit, push, and pull

1. Introduction
   1. Continuous delivery
   2. Continuous deployment
   3. Continuous integration
   4. InnerSource
2. Modern software development pipeline
   1. Git workflows: repeatable and reviewable
   2. Git Flow
      1. Master branch – deployed to production
      2. Developer branch – Development and tracking changes (purpose, rules and interaction with each other)
         1. Hotfix change
         2. Feature addition
         3. Bug fix
         4. Release
         5. Develop
         6. Helper scripts – enforce rules in workflow
   3. GitHub flow: Code, collaborate, and ship
      1. Master branch: Deployed production
      2. Feature branch: Short lived and deleted when merged into master
      3. Open pull request or git pull: Review before pushing to master
      4. Deploy into testing
      5. Merge into master branch
      6. Beyond basics – Rules for each branch
         1. Security concerns: Block direct to master - only authorized users
         2. Automated testing
         3. Code reviews
         4. Issues
         5. Control access
         6. Status checks for each commit
      7. Continuous integration: Process of automating the build and testing of changes when a commit is pushed to a branch
         1. Platform switcher broken out into each component – pass
         2. Using JS- classes so we know what is happening – pass
         3. Removing duplicate css – pass
         4. Cleaning up the content – fail
         5. Content updates – fail
         6. Spacing on tips - pass
      8. Continuous delivery: develop-build-test-deploy-release
         1. Being production ready doesn’t always translate to releasing new code every time. It gives you flexibility to release when you want
      9. Continuous deployment: Takes continuous delivery one step forward
         1. Releases are based on when the team decides it should go
         2. Closes the customer feedback, feature additions, and bug fixes
      10. Issues and pull requests
          1. Issues are more discussion based e.g. new features, direction, hackathons, and project scope
          2. Pull request are changes made on the branch – discuss and review changes
          3. Allow to modify, test and confirm changes in a safe non-destructive environment
      11. Supercharged GitHub flow
          1. Master (release team) – protection (restrictions to collaborators and committers)
          2. Sub-branch – Feature – commits – setup CI (**send commits** to GitHub so checks can be run) – protection (pull request reviewed before merging)
          3. Sub-branch – Hotfix
          4. Sub-branch - Test
3. The supercharged GitHub workflow in action: Hands on application – security, code reviews, collaboration, discussions and other features
4. Customizing your GitHub workflow
5. Resources
   1. Aaaron Stewart: www.github.com/a-a-ron
6. Knowledge repo
   1. What are some of the advanced GitHub commands?
   2. Why is three musketeers not showing up in my profile under projects?
   3. How do I change my location of cloning on desktop
   4. Should I create a team or organization within my account or separately?
   5. How to create projects?
   6. How to assign different rights?
   7. How to check in and check out of certain branches?
   8. How to transfer those changes to your branch and then eventually to the final branch?
   9. How to change a commit message? Pushed online and not pushed online?
   10. Difference between repo, packages, and teams within an organization? - <https://github.com/learn-co-students>
   11. Can you make the above private?
   12. How puts in the pull request? Person reviewing or done making changes?