CSET Notes 9-11-17

* Workflow Definitions
  + Extreme Programming
    - XP works on taking the best part of traditional programming, communication, simplicity, feedback, respect, courage, and taking them to the extreme. Uses paired programming and frequent releases.
  + Traditional Waterfall
    - Straight forward, non-iterative process of software development, where progress flows steadily downward. The phases include conception, initiation, analysis, design, construction, testing, production/implementation, and maintenance. Mostly used in physical media environments, such as design, construction, and printing. Not very open to mid project changes.
  + Feature Driven Development
  + Scrum
    - Subset of Agile, where requirements and solutions change through group efforts between self-organizing cross-functional teams. It is an iterative software development framework for managing production development. Scrum is a very cyclical methodology with an enthuses on review, stand up, and interaction in daily meetings. A backlog is a must mention due to its inevitable reliance on the cyclical nature by completing the most important to least importance.
  + Lean
    - Removes waste by decreasing time and maximizing profit. Improves efficiency and workflow. More of an ideology than methodology.
  + Spiral
    - 4 stage risk driven methodology. Planning, risk analysis, engineering, evaluation. Very pricey.
  + DSDM
  + KanBan
* Git and Github
  + What is version control
    - Way of keeping track of changes to source code, allows for separate versions to fall back to if something breaks
    - Creates accountability
    - Allows multiple devs to work on a large project simultaneously
    - Large projects can be broken down and maintained easier.
    - Let you compare files and merge code
* What is Git
  + Git is a free open source, distributed version control software
  + Primarily used for source control, but doesn’t have to be used for code, can also be used for documentation
  + Used for very small and very large projects
  + Fast
* Git glossary
  + Branch
    - A branch represents an independent line of development.
  + Centralized workflow
    - The Centralized Workflow uses a central repository to serve as the single point of entry for all changes to the project. This workflow doesn’t require any other branches besides master.
  + Commit
* An individual change to a file. When you save a file but with git every time it is saved, it creates a unique ID which is called a git. Pretty much a bookmark. Requires commit messages and commit messages give a status update.
  + Feature branch workflow
    - For each feature or issue you work on, create a new line of evelopment (branch) Then, work on implementation and testing that branch.
  + Forking
    - Creating a “fork” is producing a personal copy of someone else’s project. Forks act as a sort of bridge between the original repository and your personal copy. Forking: enables a developer to have a repository, this means that each contributor has two Git repositories: a private local one and a public server-side one
  + Gitflow workflow
* Streamlines the release cycle by using isolated branches for feature development, release preparation, and maintenance. Its strict branching model also lends some much needed structure to larger projects.
  + HEAD
    - a reference to the last commit in the currently checked out branch
  + Hook
* Scripts that run automatically every time a particular event occurs in a git repository. Hooks let you customize Git’s internal behavior and trigger customizable actions at key points in the development life cycle.
  + Master
    - Default branch that Git creates for you, typically the most up to date version of project, reflects what is live.
  + Merge
    - Combining branches
  + Push
    - Taking what you have locally and sending it remotely. Big progress
  + Pull
    - Bringing something remote to local
  + Pull request
    - Tell others about changes you made, people can review those changes. They give you permission to merge
  + Repository
    - Collection of commits, branches, and tags to identify commits. Where all the code lives. Also called a repo.
  + Tag
    - Reference point typically used to mark a point in a commit chain.
  + Working tree
    - Diagram of files that are being worked on
* Git fundamentals
  + 3 main stages
    - Committed
      * Data is safely stored in local version
    - Modified
      * Changed but not yet committed
    - Staged
      * Marked modified file in current version to go up in next commit
  + 3 main sections
    - Working directory
      * What you’re working on, locally.
    - Staging area
      * File contained in directory that tells what will go into next commit
    - Git Directory
      * Where git stores the metadata for your project
* Git basic workflow
  + Modify files in working directory
  + Stage the files, add snapshots of them to staging area
  + You do a commit, which takes files as they are in staging area and stores them permanently in git directory
* Benefits of Git
  + Branching and merging
  + Small and fast
  + Distributed
  + Data Assurance
  + Staging area
  + Free and open source
* Branching and merging
  + Master
    - Develop
      * Topic
* Allows for disposable experimentation
* Distributed
  + Multiple back ups
  + Various workflows
* Staging area
  + Working directory
    - Git Add
* Staging area
  + Git Commit
* Repository
* Free and opensource
  + Uses GPLv2 to guarantee your freedom to share and change free software---to make sure the software is free for all its users.
* What is github
  + Mostly used for code, but not necessarily
  + Offers features of git and then some
  + Has montly fee for private repos
* Fundamentals of github
  + Clone a repo
  + Create a branch
  + Add commits
  + Open a pull request
  + Discuss and review code
  + Deploy
  + Merge
  + Contribute to open source projects
* Contributing to open source projects
  + Good way to learn, teach, and gain exp about any skill
  + Improve existing skill
  + Learn new skills
  + Collaborate
  + Meet new people
  + Find mentors
  + Share knowledge
  + Learn people skills
  + Have portfolio pieces
  + Build artifacts that help grow reputation
  + Find empowerment to contribute
* What does contributing mean?
  + Not necessarily code
  + Could be fleshing out documentation
  + Testing
  + Any number of ways to contribute