CST0006 – Computer Programming Foundations

INTRODUCTION to WEEK00

It is my opinion that there is nothing more frustrating and nothing more rewarding than writing software. As with any industry, having the right tools and a background knowledge of how to do the individual tasks needed to complete a project, will make all the difference in building your project properly and efficiently or having it take way too long and potentially needing to call someone that can actually do the job for you. Properly.

Programming has many tools that are used and understanding how to use these tools before you even learn how to write a line of code can help you get a great head start. When most people begin programming, a lot of the time they grab a text editor and start blindly coding away, hoping to bash out a working program. The common scenario is that they are working on a program, things are awesome, everything they test is working, then they make some changes and BAM! Nothing works… and they've changed the code so much that there is no way to simply go back to when it was working again. This can potentially take hours to backtrack to the point where the code works again. All of that time is wasted time.

A Source Code Management System is a tool that you can use to help you track the changes you make to your code and immediately step back in time to when the code was working.

SCM - Source Code Management System

Source Code Management Systems allow you to track the changes you make to your programs.

There are many Source Code Management Systems in use today, but Git is by far the most popular source code management system in the world. Another name that an SCM can go by is VCS – Version Control Systems. I personally think that SCM is a more thorough and descriptive name and can include VCS as part of its description.

Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. (Description from their website)

A distributed version control system means you get a complete local copy of the entire software repository and the history of all the changes that have been made. You are able to code away and build awesome changes on your own and when those changes are complete, you can share those changes. Compare this to a centralized version control system where you work on a shared codebase and if the server ever crashes or the code becomes buggy, you're done working until these issues are fixed.

One of the strengths of Git is you can benefit from both types of systems by including a central software repository.

Central Software Repositories

A Central Software Repository is useful because there is one location that everyone can sync with allows for better collaboration. So that when you submit your awesome changes, everyone else in the group can pull your changes. Some famous Central Software Repositories are Gitlab, Bitbucket and Github. Most software projects you will work on will be a **collaborative** effort, having a Source Code Management System keeps everyone's code organized and one central location for everyone to keep their git repositories synced together.

Another reason to use a Central Software Repository is that some of the popular online tools like Github and Gitlab has work-flow tools to allow you to collaborate and track the workload between each member.

Work-flow

Most software projects you will work on will be a collaborative effort. Having a system to control who works on what, and how far along they are is important for the project maintainers to know. That way they can move people around or help in areas that are missing deadlines.

Waterfall vs Agile

There are two main categories to track the work-flow of software development. The one that is considered by a lot of people to be the old way, is the waterfall method. Picture water flowing down a staircase and you have a picture of how this method works. There is a step for planning, then when planning is finished, control goes to development and then down to testing and so on. If there are any changes that need to happen, you have to go all the way back up to the top and flow down again. Changes are costly and move the release date further away then was originally planned.

The newer method is called Agile. Agile's main selling point is that you do just enough planning and code and testing so that you can adapt to changes. In fact, change is considered a part of the overall development cycle. Agile was developed from the TPS (Toyota Production System) and Agile now considered the umbrella term for all development methodologies, including TPS. There are many different types of Agile development methods, but scrum tends to be the one that is used most often. The main idea behind scrum is short daily meetings called a scrum and a short development cycle usually in two week increments. Each member takes on a task or two and does their part. If they run into any issues, there is a scrum master that will resolve those issues to keep development flowing.

Kanban Boards vs Scrum Boards

Kanban boards and Scrum boards are very similar in many ways. They are both tools to separate the individual tasks that are needed to be preformed to end up with a working product. Each of these tools is specifically designed for the type of Agile Methodologies that you are using. There is no set rule about you split up your work-flow, but there is a general design of a pre-production, in-production and complete.

IDEs - Integrated Development Environments

An Integrated Development Environment is a software tool that groups useful development tools and integrates them into one larger tool. There is a source code editing tool that allows you to write your program, there is usually some kind of debugging tool, there is an integrated compiler or interpreter and one of the best reasons to use an IDE is that it highlights or colourizes your code and lets you know if you've made any mistakes. Some people think this is cheating or makes you a lazy coder. I disagree. Any tool that allows you to work more efficiently is a good tool.

Resources:

Python Book:	http://greenteapress.com/thinkpython2/thinkpython2.pdf
Git Book:	https://git-scm.com/book/