Dreaming in Code Questions

Overarching Themes

Pay attention / record the various roles that software engineers have on the Chandler Project.
Pay attention / record the many scheduling issues related to Chandler Project.

Chapter 0

1. Who wrote "software is hard?" Who is that guy?
   Donald Knuth
2. Programmers start counting at what number?
   0
3. What was the original sense of a "hacker?"
   A type of programmer
4. According to a 2002 NIST study what % of software came in significantly late, over budget, or was canceled?
   66.6%
5. Who wrote the 1987 essay entitled "No Silver Bullet?"
   Frederick P. Brooks Jr.

Chapter 1

1. What roles in the Chandler project did Michael Toy, John Anderson, Ted Burgess, Mitchell Kapor, and Lou Montulli hold.
   They are programmers. Michael Toy is the lead, John Anderson is a systems architect.
2. What is "Bugzilla?"
   It's a bug management program.
3. What is OSAF?
   Open Source Applications Foundation
4. What is the projects name?
   Chandler
5. What will the software do?
   Manage information of the "personal" variety.
6. What is Toy's keyword for "black hole" bugs?
   scary
7. What scared Toy so much about Bug 44?
   Ultimately, he had no idea how much time it would take to resolve.
8. What did Toy refer to as a "snake?"
   An important problem with no plan of action.
9. In the software world, what does "slippage" mean?
   Lateness of software development
10. Fredrick Brooks was a programming manager for what software project?
    The operating system of the IBM System/360
11. What is Brooks' Law?
    "Adding people to a late software project makes in later"

12. Brooks found what % of project time was spent writing code?
    .166666666%
13. Brooks found what % of project time was for testing and fixing bugs?
    50%
14. Brooks observed that the unit of effort named "man-month" only applied under what conditions?
    When tasks can be split between workers and they don't have to communicate between each other.
15. What is the difference between source code and the program you install (.exe) on your computer?
    The source code gives opportunity for others to easily inspect and modify. The install program does not.
16. What is the one "article of faith" that all "open source" or "free" software advocates share?
    "Open" software will improve much better than "closed" software.
17. What is the difference between a "good" programmer and a "great" programmer?
    A great programmer know what to rewrite.
18. Eric Raymond's book "The Cathedral and the Bazaar" made a distinction between two important project development ideas, briefly contrast them.
    Closed vs. open development.
19. Has "open source" software project development refuted Brooks's "mythical man-month" concerns?
    No, open source confirms Brooks' law. It's like a Rotary engine. It is less efficient, but works faster because the sheer volume of "strokes/work."
20. What was Andy Hertzfeld's input when the Chandler project appeared to have stalled?
    Stop designing and start coding.

Chapter 2

1. Linus Torvalds used a "science" and "witchcraft" analogy referring to software, explain.
   He compares traditional software to witchcraft…guarded and in secret. Open sources like science people building upon each other.
2. People often refer to starting their computer as "booting" their computer. What was the origin of this term?
   The computer must pull itself up by it's bootstraps.
3. Where was the graphical user interface (GUI) developed?
   Palo Alto, CA
4. List three software project "train wrecks."
   Virtual Case File project, Innovate by Mc'Ds, Everest by Ford.

Chapter 3

1. When introducing a new technology or design, why did Frederick Brooks advise "plan to throw one away?"
   He says this because it probably won't be right the first time.
2. What is a "core" dump? Why the use of the word core?
   The old computers would store memory in wire coils called ferrite cores. Therefore the use of the word core carried over even though cores were no longer used. A core dump is a memory dump.

This happens when there is an error and the computer stops makes of file showing what it's memory content was when the error occurred.

3. Rather than writing program statements in binary code, 110101110 1001101111, programmers developed a shorthand language called what?
   assembly language
4. Adding layers of abstraction, new programming languages were created: Lisp, Cobol, Algol, Basic. Fortran was the first widely used. What kind of program converted Fortran to binary?
   compiler

Chapter 4

1. What do "front ends" and "back ends" mean to software developers?
   The front end is general where a user interacts with a program. The back end is more behind the scenes and its hopefully only seem by the programmers. The front end draws of the back end.
2. What did the Lego Hypothesis refer to?
   "Programs will be built out of reusable parts"
3. Give one reason why the Lego Hypothesis seems to not work so well.
   Often times parts of programs and not very similar or reusable.

Chapter 5

1. What is the three-way trade-off that many software projects struggle to overcome.
   Cost, speed & quality
2. What is the more recent definition of "geek?"
   "Geek: A person who has chosen concentration rather than conformity; one who pursues skill and imagination, not mainstream social acceptance."
3. What does "refactoring" mean to programmers?
   Rewrite code to make it better
4. What is "yak-shaving?"
   An activity that seems pointless, but solves problems that eventual lead to solving the real problem.

Chapter 6

1. What is term "edge cases" referring to in software development?
   It is a problem that lies outside of typical operation.
2. Summarize briefly Linus Torvalds advice about "large projects" give in 2004
   He said no one should take on large projects. Torvalds thinks a programmer should start small, focusing on the details, then go from there.

Chapter 7

1. Briefly describe Hungarian notation
   It's a style of naming variables. The programmer attaches a prefix to every variable name. This prefix is supposed to help give a clue to what the variable is or what it does. Some people are not amused.

2. What does the author state is the "...single most challenging demand of software development."
   Communicating between machine and programmer, programmers among each other, and programs to users

Chapter 8

1. What does "eat your own dogfood" mean?
   Using the software yourself in a very real way to root out bugs. Dogfood = software functionality
2. Quote: "When people ask for numbers that far out, the traditional thing that engineers do ...."
   When discussing the timeline for Chandler, how was the quote above completed?
   They are sitting in a meeting and Dusseault makes this statement looking out the window. He finishes it with: "is make them up." It's his response when given a timetable that he thinks is out of the question.

Chapter 9

1. Structured programming evolved to address what programming practice?
   Taking all the code and throwing it all together into one pot.
2. Was structured programming a solution to the problem of software development?
   No. According to the author it "laid a solid foundation for more ambitious software to come". Rosenberg also says that this software found "new ways to fail" (241)
3. Have any techniques shown to improve the software development process?
   It's a little hard to say. Rosenberg does not seem to give an astounding thumbs up to any one process, though they may have had positive attributes. It's interesting, I find that I do my best and most productive work in a completely unstructured and unsupervised environment. However, the same can be said of my worst work!
4. The "waterfall model" of programming was/is popular. What were some problems with this model?
   It took too long, and nothing worked right. Programmers were either idle or working too far ahead.
5. What are the four tenets of Agile Software Development?
   a. Individuals and interactions over processes and tools
   b. Working software over comprehensive documentation
   c. Customer collaboration over contract negotiation
   d. Responding to change over following a plan
6. What did a 2004 study find about the development practices of some two hundred software team leaders?
   One third of developers did not have any development practices
7. What is the "Joel Test" and what did he say about Microsoft and the Joel Test.
   The Joel Test is a set of principles in the form of 12 questions. Joel is looking for a yes on no less than 11 questions. Joel Spolasky say most software companies are at 2 or 3, but Microsoft runs at 12.
8. What is Rosenberg's Law?
   "Software is easy to make, except when you want it to do something new. And then, of course, there is a corollary: The only software that's worth making is software that does something new"

Chapter 10

1. Chapter 10 is about the notion of "Software Engineering" and the difficulty of applying this label to the development of software. The author suggests that Yertle the Turtle provides an important lesson for programmers. Describe it.
   If a layer (turtle) of software fails then all the layers above that layer fail as well. There could be many lessons. Make a firm foundation and don't build your house on sand. Reinforce your foundation before you add another layer to it. Double check the foundation. Don't add another layer to crap. Make sure each layer is done "right."

Remaining Pages

Complete the reading reflecting on the Chandler Projects **scheduling** issues and the various **project roles** that were important on the project.