**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

1. Programmers start counting at what number?

Zero

1. What was the original sense of a “hacker?”

"obsessive programming tinkerer"

1. According to a 2002 NIST study what % of software came in significantly late, over budget, or was canceled?

2/3, or 66%

1. 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.

John Anderson- systems architect, lead coder. Mitchell Kapor- founder and funder of OSAF, Ted Burgess- deckhand,

1. What is “[Bugzilla](http://www.bugzilla.org)?”

A program created to contain bug lists

1. What is [OSAF](http://www.osafoundation.org)?

Open Source Applications Foundation

1. What is the projects name?

Chandler

1. What will the software do?

"personal information manager"

(PIM) for organizing and sharing calendars, email, to-do lists, and all the

other stray information in our lives.

1. What is Toy’s keyword for “black hole” bugs?

scary

1. What scared Toy so much about Bug 44?

the impossibility of

knowing how long it would take to fix.

1. What did Toy refer to as a “snake?”

An "important problem that we don't have consensus on how to attack."

1. In the software world, what does “slippage” mean?

The peculiar

resistance of software projects to routine scheduling is both notorious and

widely accepted.

1. Fredrick Brooks was a programming manager for what software project?

creation of the operating system for the IBM System/360

1. What is [Brooks's Law](http://scottberkun.com/2006/exceptions-to-brooks-law/)?

"Adding manpower to a late software project makes it later."

1. Brooks found what % of project time was spent writing code?

one-sixth of their time

1. Brooks found what % of project time was for testing and fixing bugs?

Fifty percent

1. Brooks observed that the unit of effort named “man-month” only applied under what conditions?

If productivity could be broken down into discrete, identical, fungible units or "men and months are interchangeable commodities only when a task can be partitioned among many workers with no communication among them."

1. What is the difference between source code and the program you install (.exe) on your computer?

"binary" or "machine-executable" file: It contains

machine-level code, impenetrably dense sequences of zeros and ones that

is unreadable by most human beings and even most programmers. This

binary file is not the direct product of human labor but, rather, the output

of a compiler—a computer program that takes lines of code that human

programmers wrote in a language like Java or C or C++ and translates them

into the machine's language.

If you want to understand, say, how Microsoft Word was written, you

can't find out by peering into its binary file; you would need to see the source

code—the thousands of lines of human-written program code that was grist

for the compiler. But Microsoft, like most commercial software enterprises,

won't let you. The source code to its programs is its most coveted asset, and

a Berlin wall of intellectual property law protects the treasure.

1. What is the one “article of faith” that all “open source” or “free” software advocates share?

that software anyone can tinker with is bound to

improve over time in ways that "closed" software can't match.

1. What is the difference between a “good” programmer and a “great” programmer?

"Good programmers know what to write. Great ones know what to rewrite (and

reuse)."

1. Eric Raymond’s book “[The Cathedral and the Bazaar](http://www.catb.org/esr/writings/cathedral-bazaar/)” made a distinction between two important project development ideas, briefly contrast them.

Eric describes his preferred method of developing akin to building a cathedral and linus torvald’s method to that of an open bazaar, or “open source”

1. Has “open source” software project development refuted Brooks’s “mythical man-month” concerns?

"The Cathedral and the Bazaar" doesn't actually repeal Brooks's Law and solve the problem of time in software development; it maps an alternate universe for programming in which

time is simply less important because the work is cooperative rather than corporate, the workers are all volunteers, and the motivation is fun and ego, not financial reward. So no, not entirely.

1. What was [Andy Hertzfeld](http://andy.hertzfeld.usesthis.com)’s input when the Chandler project appeared to have stalled?

“You're working on it to do something great. But you need to get it started! The key is getting exciting work going; the rest of it will take care of itself. You're sparking off each other—a virtuous cycle—once you're doing the thing you're there to do."

**Chapter 2**

1. Linus Torvalds used a “science” and “witchcraft” analogy referring to software, explain.

Science allows other people to study and build off of it, witchcraft is a closely guarded secret know by few and like (traditional software) witchcraft it will die out

1. People often refer to starting their computer as “booting” their computer. What was the origin of this term?

The builders of early computer systems had borrowed the term from the concept of pulling one's self up by the bootstraps to describe the paradox of getting a computer up and running

1. Where was the graphical user interface (GUI) developed?

Xerox’s Palo Alto research center

1. List three software project “train wrecks.”

Virtual Case File project, Future Combat System, Everest

**Chapter 3**

1. When introducing a new technology or design, why did Frederick Brooks advise “plan to throw one away?”

Because you almost certainly won’t get it right the first time

1. What is a “core” dump? Why the use of the word core?

the computer drops everything, grinds to a halt, and spits out a file reporting the exact contents of its memory at the moment of failure, offering bug hunters a heap of clues to dig through. And due to the memory banks of early computers being wound wire coils known as ferrite cores

1. Rather than writing program statements in binary code, 110101110 1001101111, programmers developed a shorthand language called what?

Assembly

1. 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?
2. What did the Lego Hypothesis refer to?
3. Give one reason why the Lego Hypothesis seems to not work so well.

**Chapter 5**

1. What is the three-way trade-off that many software projects struggle to overcome.
2. What is the more recent definition of “geek?”
3. What does “refactoring” mean to programmers?
4. What is “yak-shaving?”

**Chapter 6**

1. What is term “edge cases” referring to in software development?
2. Summarize briefly Linus Torvalds advice about “large projects” give in 2004

**Chapter 7**

1. Briefly describe Hungarian notation
2. What does the author state is the “...single most challenging demand of software development.”

**Chapter 8**

1. What does “eat your own dogfood” mean?
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?

**Chapter 9**

1. Structured programming evolved to address what programming practice?
2. Was structured programming a solution to the problem of software development?
3. Have any techniques shown to improve the software development process?
4. The “waterfall model” of programming was/is popular. What were some problems with this model?
5. What are the four tenets of Agile Software Development?
6. What did a 2004 study find about the development practices of some two hundred software team leaders?
7. What is the “Joel Test” and what did he say about Microsoft and the Joel Test.
8. What is Rosenberg’s Law?

**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.

**Remaining Pages**

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