**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 Kuth wrote Software is hard, He is the author of the book The Art of Computer Programming**

1. Programmers start counting at what number?

**Programmers start counting at zero.**

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

**The original sense of a hacker was “Obsessive programming tinkerer” rather than the later, tabloid sense of “digital break in artist”**

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

**“According to the 2002 NIST study 2 out of 3 projects came in significantly late and over budget or was cancelled.”**

1. Who wrote the 1987 essay entitled “No Silver Bullet?”

**Fred Brookes wrote the essay entitled “No Silver Bullets.”**

**Chapter 1**

1. What roles in the Chandler project did Michael Toy, John Anderson, Ted Burgess, Mitchell Kapor, and Lou Montulli hold.
2. What is “[Bugzilla](http://www.bugzilla.org)?”

**Bug 44 was known as Bugzilla the famous “flicker-free window display when resizing windows.”**

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

**Open Source Application Foundation is also known as OSAF.**

1. What is the projects name?

**Flicker free window display was the projects name.**

1. What will the software do?

**Supposedly it was supposed to take away the flicker when a window resizes.**

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

**“He called it 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?”

**Toy referred to “important problems that we don’t have consensus on how to attack. Snake superseded a looser usage at OSAF of he word dragon to describe the same phenomenon” as snakes.**

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

**“In the software development world, lateness was so common that a new euphemism had to be invented for it: slippage.”**

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

**He was the manager for the IBM system/360 project**

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

**Brooks’s Law is “Adding manpower to a late software project makes it later.”**

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

**1/6 of their time was spent in writing code**

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

**Brooks also found that ½ their schedule for fixing and testing bugs.**

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

**It only works 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?

**The program you install is a dense sequence of ones and zeros where as a source code is what companies sell so that you can run their product. A program is like a translator because it takes lines of code and translates it into the computers language.**

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

**GNU**

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

**A good programmer knows what to write. A great one knows what to rewrite.**

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.

**Unix and Linux. Linux is a open sourced software that is bare boned. Unix has a multiuser platform and is one of the strongest computers out there.**

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

**It doesn’t in fact refute Brooks law but map an alternate universe in programming.**

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

**It was that they needed to get started. They were told that they needed to get exciting work going.**

**Chapter 2**

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

**He was saying that science is seeing someone else’s work and building on it. Whereas with witchcraft someone figures out how to do something and then keeps it their small secret refusing to share how they did it.**

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

**“Booting up” a computer was meant by Engelbart as “Improving of the improvement process.”**

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

**The Graphical User Interface was developed at Xerox’s Palo Alto Research center**

1. List three software project “train wrecks.”  
   **The FBI’s Virtual Case File, The FBI’s Trilogy, and McDonald’s “Innovate”**

**Chapter 3**

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

**Because you most certainly won’t get it right the first time.**

1. What is a “core” dump? Why the use of the word core?
2. Rather than writing program statements in binary code, 110101110  1001101111, programmers developed a shorthand language called what?
3. 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?

**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.