

CS1001

Lecture 11

Overview

- Software Engineering
- Design Methodologies
- Software Ownership

Announcements

- Homework Pickup: Fri. 2-3pm in TA Room (or on Monday)
- Midterm: Mar 24th, in class
- Apr 19th: Final Paper due
- May 3rd: Technical Project
- Final (Projected): Wed, May 12th
4:10pm-7pm

Goals

- Understand the basics of the typical software development cycle
- Design Methodologies
- Understand Introductory Algorithm Analysis (evaluate how good/bad an algorithm is)

Assignments

- Brookshear: Ch 6.1,6.2; Ch 4
(Read)
- Read linked documents on these slides
(slides will be posted in courseworks)

“Learn to Program in 24 Hours”

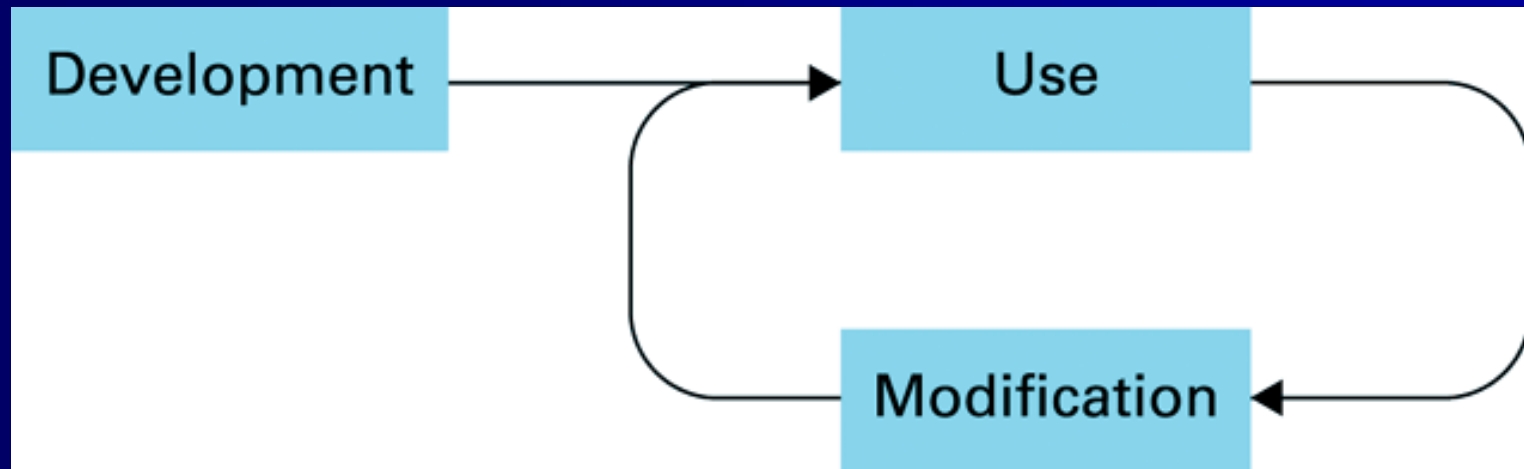
- Building large, critical software systems is an engineering discipline
- Challenges:
 - In traditional engineering, you can use “off the shelf” parts (interchangeable parts). Software has tried repeatedly to mimic this... somewhat successfully
 - Tolerances: Mechanical systems can fail, but in doing so they rarely wreak complete havoc on their surroundings. A software crash *often* can bring down an operating system or cluster of computers
 - Metrics: How do you evaluate whether software is performing well? (refer to previous lecture on usability)

Parallels

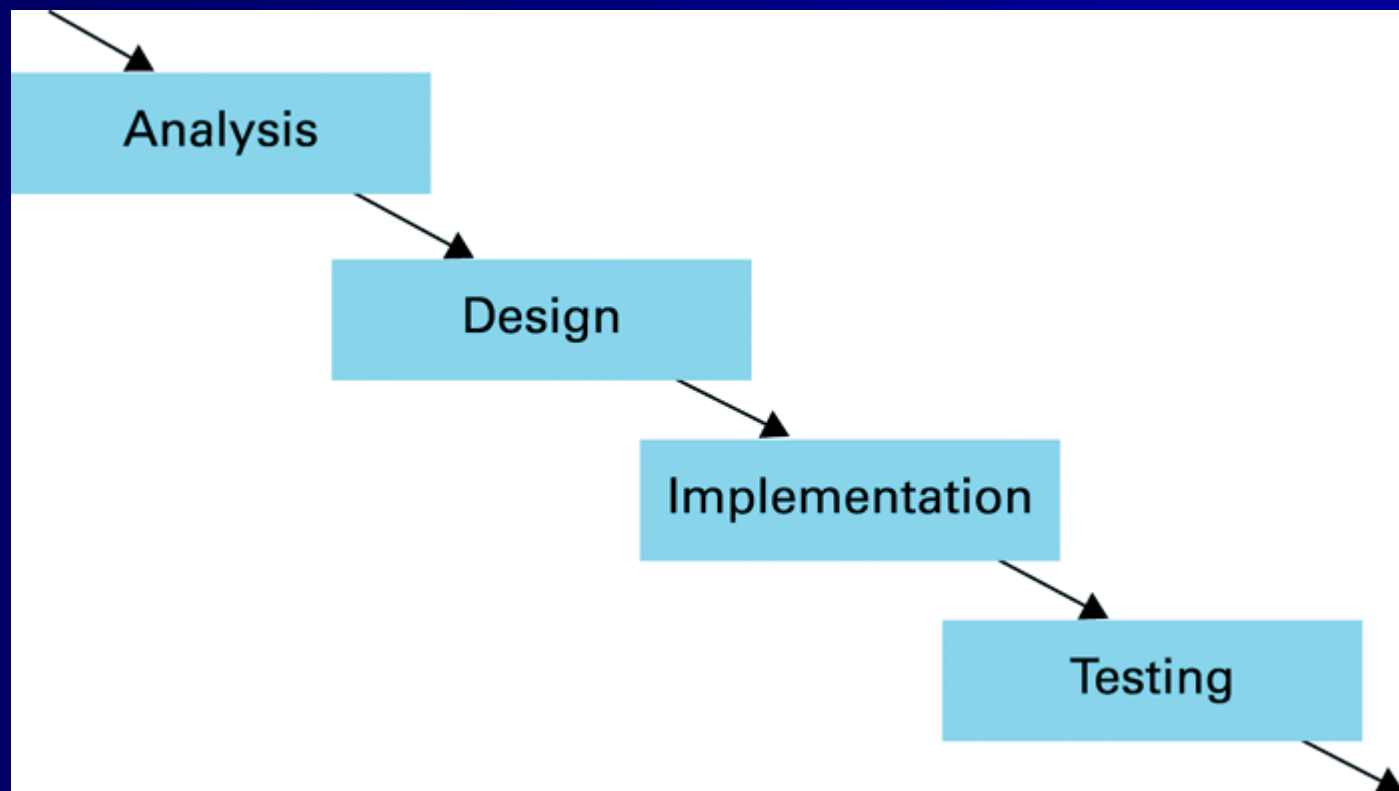
- Software is a fundamentally different engineering discipline, *but* software systems still must be reliable.
- We have PE (Professional Engineering) certifications for Engineers, MLEs, the BAR, Architecture licenses, yet no standardized software license
- Once methods and metrics established, software engineering can become a more structured field

Iterative Software Development

- Continually build and test your work
- Similar to how one might undergo a physical engineering project



The Development Phase



Trends

- The waterfall model of development
- Throwaway vs evolutionary programming
- Extreme programming (enforced teamwork)

Figure 4.10: Sorting the list Fred, Alice, David, Bill, and Carol alphabetically (continued)

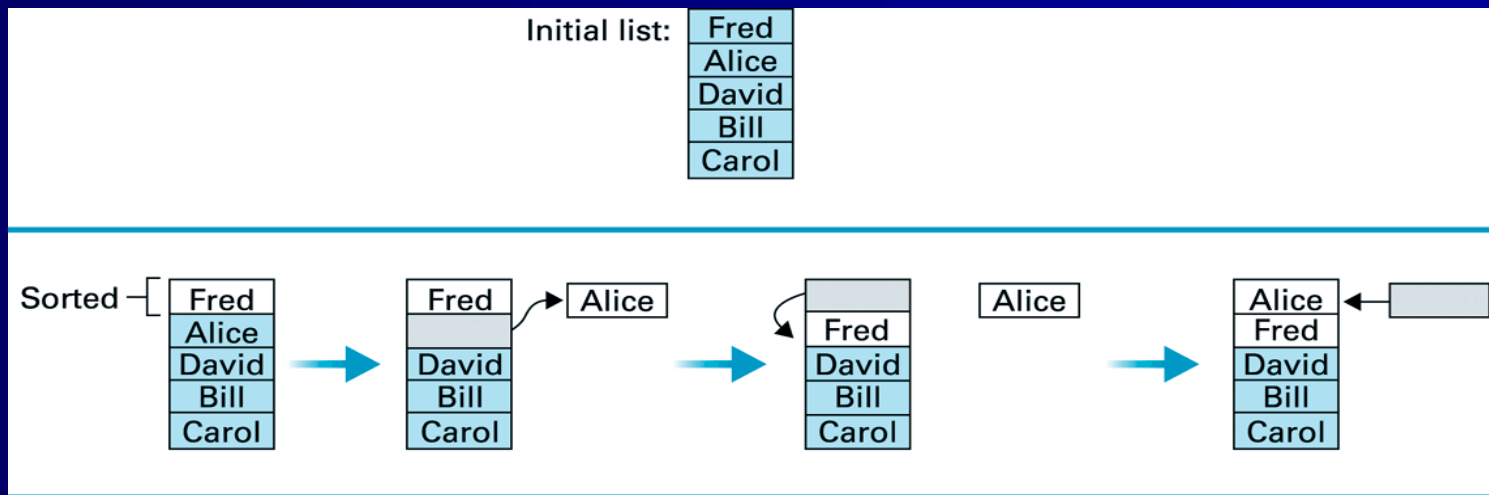


Figure 4.10: Sorting the list Fred, Alice, David, Bill, and Carol alphabetically (continued)

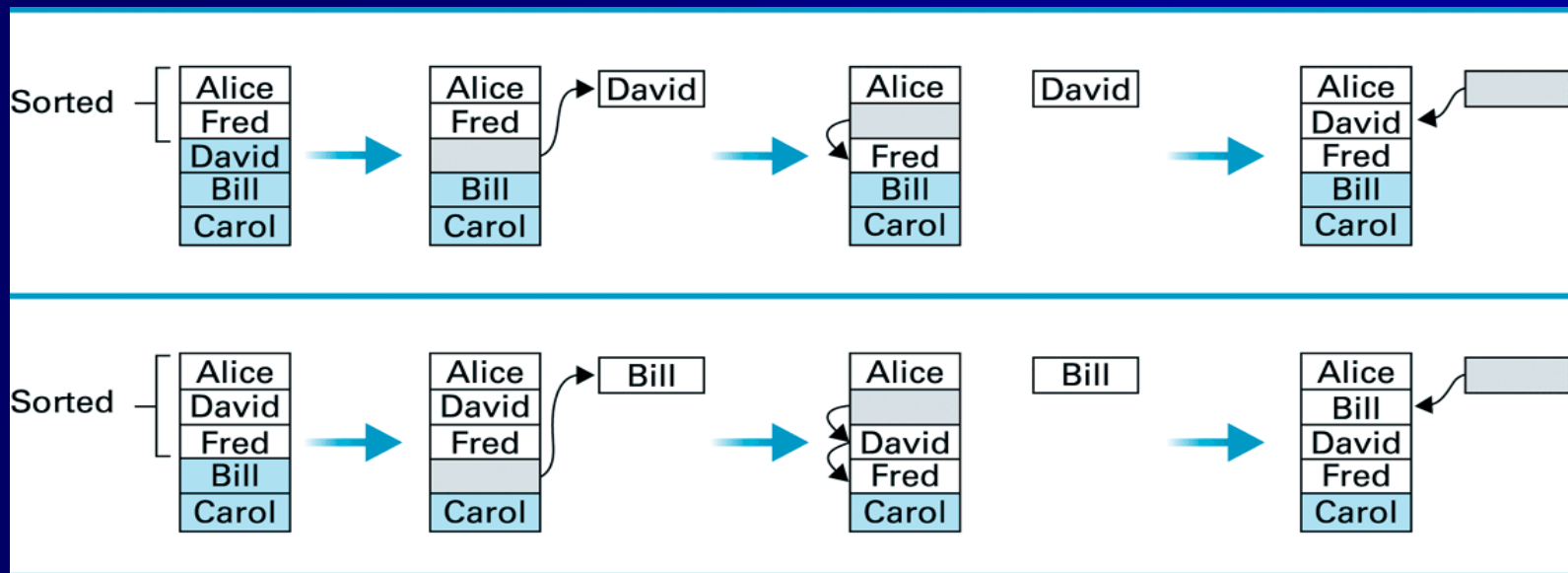


Figure 4.10: Sorting the list Fred, Alice, David, Bill, and Carol alphabetically

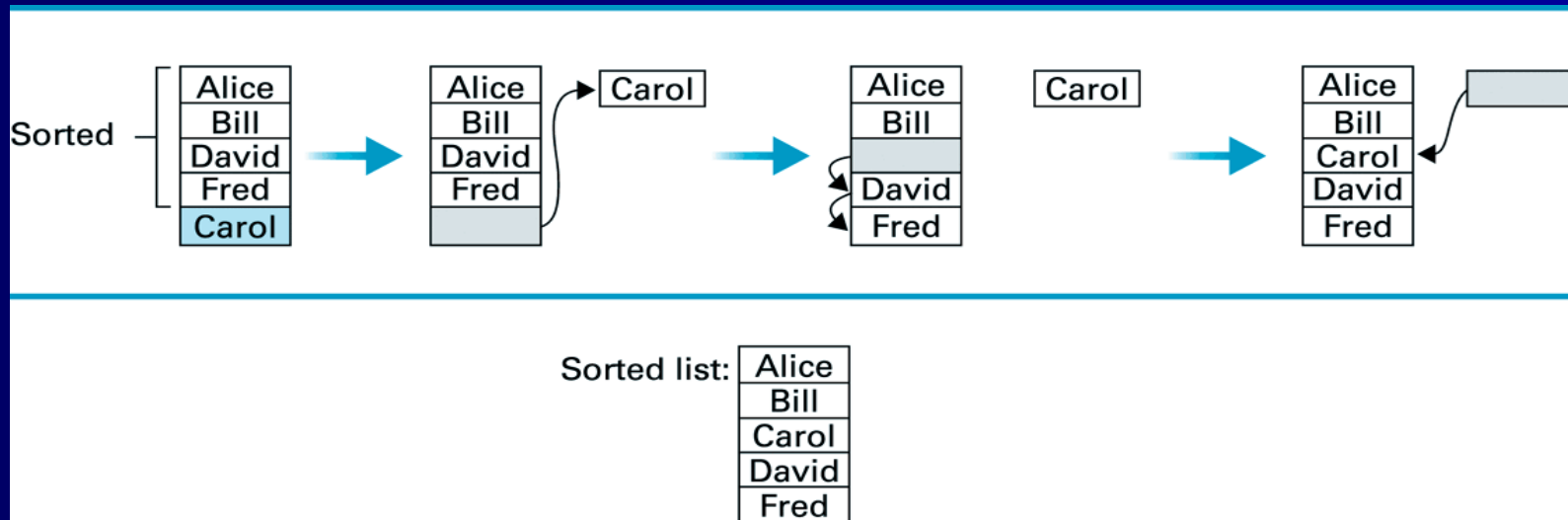
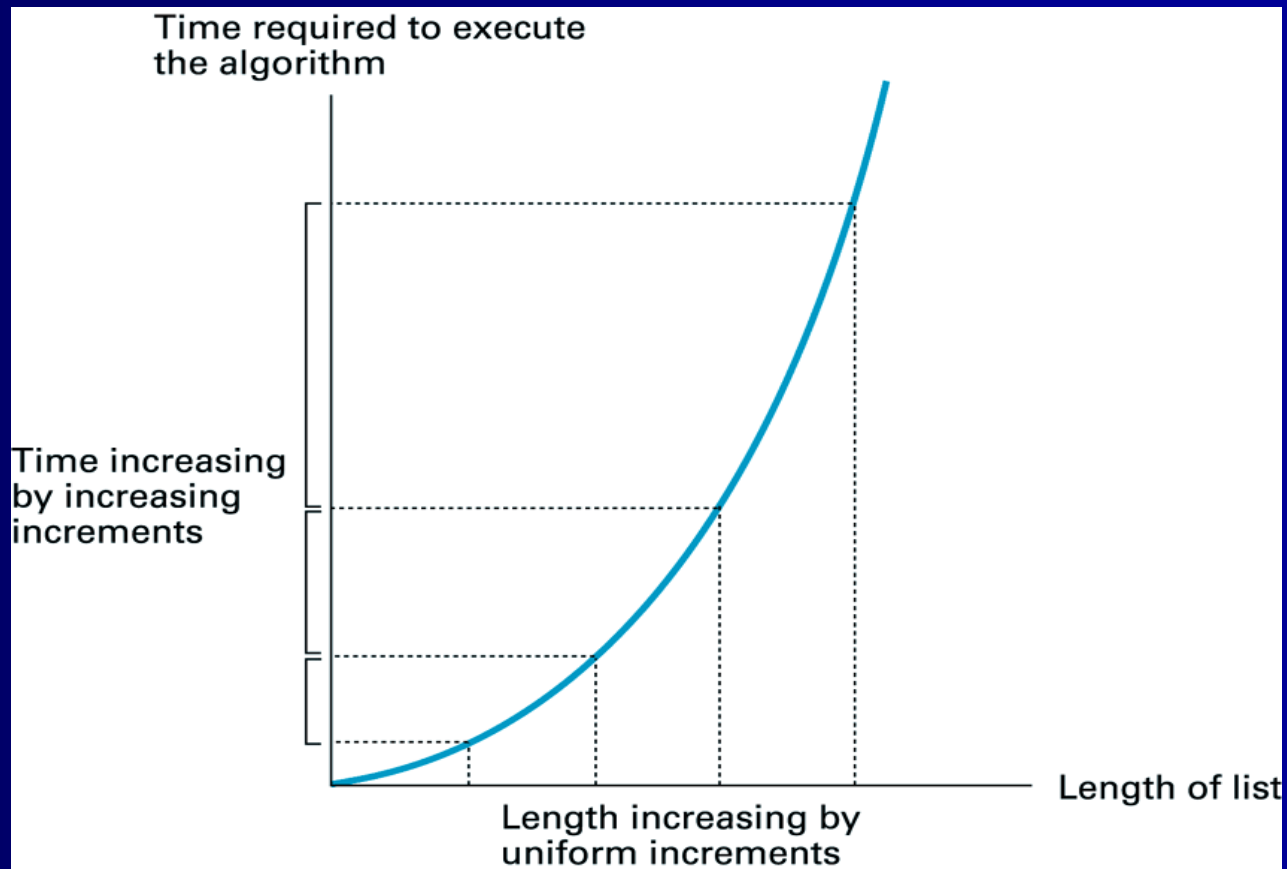


Figure 4.11: The insertion sort algorithm expressed in pseudocode

```
procedure Sort (List)
N ← 2;
while (the value of N does not exceed the length of List) do
  (Select the Nth entry in List as the pivot entry;
  Move the pivot entry to a temporary location leaving a hole in List;
  while (there is a name above the hole and that name is greater than the pivot) do
    (move the name above the hole down into the hole leaving a hole above the name)
  Move the pivot entry into the hole in List;
  N ← N + 1
)
```

Figure 4.19: Graph of the worst-case analysis of the insertion sort algorithm



Licenses

- Buy the software for unlimited use
- Lease the software as a service
- GPL (General Public License) and other free licenses