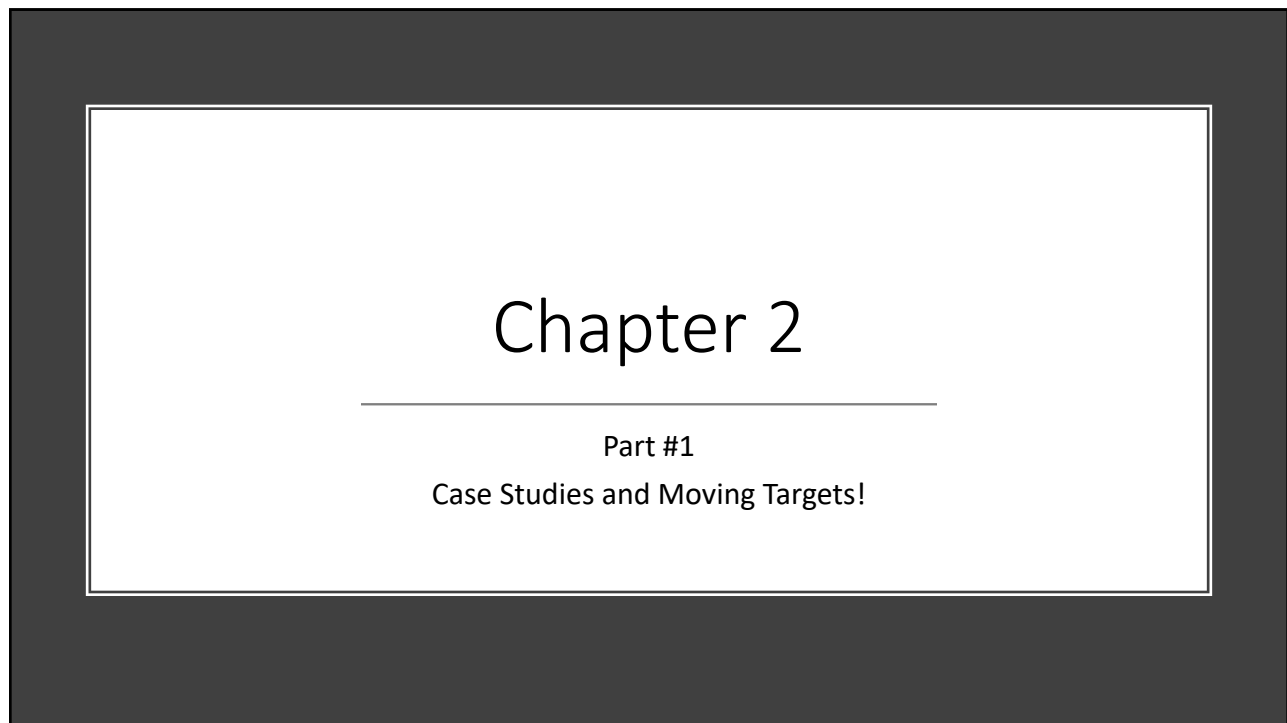


1



2

2.1 Software Development in Theory

- Ideally, software is developed as described in Chapter 1
 - Linear
 - Starting from scratch

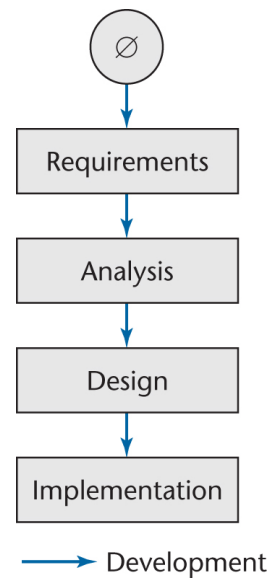


Figure 2.1

3

Software Development in Practice

- In the real world, software development is totally different
 - We make mistakes
 - The client's requirements change while the software product is being developed

4



2.2 Winburg Mini Case Study

- **Episode 1:** The first version is implemented
- **Episode 2:** A fault is found
 - The product is too slow because of an implementation fault
 - Changes to the implementation are begun
- **Episode 3:** A new design is adopted
 - A faster algorithm is used
- **Episode 4:** The requirements change
 - Accuracy has to be increased
- **Epilogue:** A few years later, these problems recur

5

Evolution- Tree Model

Winburg Mini Case Study

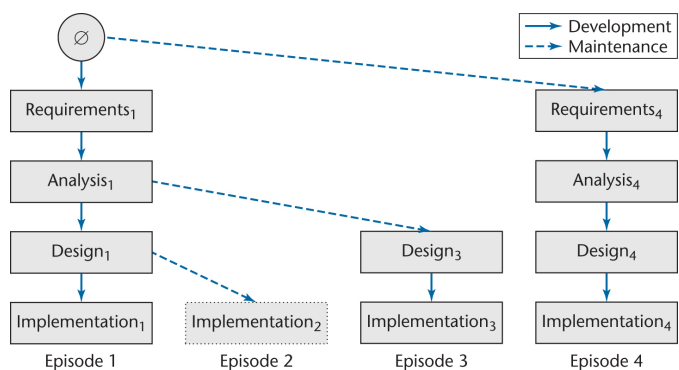
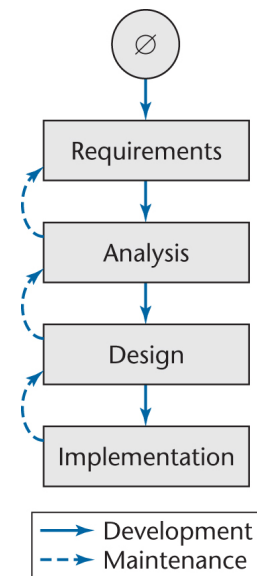


Figure 2.2

6

Waterfall Model

- The linear life cycle model with feedback loops
 - The waterfall model cannot show the order of events



7

Return to the Evolution-Tree Model

- The explicit order of events is shown
- At the end of each episode
 - We have a *baseline*, a complete set of *artifacts* (constituent components)
- Example:
 - Baseline at the end of Episode 3:
 - Requirements₁, Analysis₁, Design₃, Implementation₃

8

2.3 Lessons of the Winburg Mini Case Study

- In the real world, software development is more chaotic than the Winburg mini case study
- Changes are always needed
 - A software product is a model of the real world, which is continually changing
 - Software professionals are human, and therefore make mistakes

9



2.4 Teal Tractors Mini Case Study

- While the Teal Tractors software product is being constructed, the requirements change
- The company is expanding into Canada
- Changes needed include:
 - Additional sales regions must be added
 - The product must be able to handle Canadian taxes and other business aspects that are handled differently
 - Third, the product must be extended to handle two different currencies, USD and CAD

10

Teal Tractors Mini Case Study (contd)

- These changes may be
 - Great for the company; but
 - Disastrous for the software product

11

Moving Target Problem

- A change in the requirements while the software product is being developed
- Even if the reasons for the change are good, the software product can be adversely impacted
 - Dependencies will be induced

12

Moving Target Problem (contd)

- Any change made to a software product can potentially cause a *regression fault*
 - A fault in an apparently unrelated part of the software
 - Or a brand new fault in the same area
- If there are too many changes
 - The entire product may have to be redesigned and reimplemented
 - It won't be

13

Moving Target Problem

- Change is inevitable
 - Growing companies are always going to change
 - If the individual calling for changes has sufficient clout, nothing can be done about it
- There is no solution to the moving target problem

14