1. **Successful software engineering:** The use of other software engineering methods such as reviews and inspections, structured methods, configuration management and information hiding in software design and development processes have resulted in improvements in software quality. People who suggested that the only way to improve software quality was by using formal methods were clearly wrong.

**Market changes:** In the 1980s, software quality was seen as the key software engineering problem. However, since then, the critical issue for many classes of software development is not quality but time-to-market. Software must be developed quickly, and customers are sometimes willing to accept software with some faults if rapid delivery can be achieved. Techniques for rapid software development do not work effectively with formal specifications.


4. It had the goal of extending and adapting the SEI maturity model to make it applicable across a wider range of companies (i.e., more applicable to companies that are NOT US defense contractors).

5. G, L, D, A, O, M, B, I, E, J

6. d

7. When things go wrong – e.g., changing client priorities, approval methods, business rules and requirements – and this results in delays, false starts, and re-work, contractors are likely to seek process shortcuts if the ability of the contractor to absorb cost overruns is greatly exceeded.

   This can result in product risks such as reliability issues, performance problems, usability issues, reduced maintainability, etc.

8. e

9. a

10. a. true; b. false; c. false; d. false, e. true

11. c

12. pre-condition: \{ N\geq 1 \}
   post-condition: \{ [ (1\leq {LAST}\leq N \land A[\text{LAST}]=K \land \text{for every } \text{LAST} < i \leq N, A[i]\neq K) \lor (\text{LAST}=0 \land \text{for every } 1 \leq i \leq N, A[i]\neq K)] \land \text{UNCH}(A) \}

13. \( f = (x>0 \rightarrow x,y := 0,y+x | \text{true} \rightarrow I) \)
   Also acceptable, but unintended: \( f = (x>0 \rightarrow f \circ (x,y := x-1,y+1) | \text{true} \rightarrow I) \)

14. d
15. a. false; b. true; c. false

16. \{P\} S \{I\}, \{I \land \neg b\} S \{I\}, (I \land b) \Rightarrow Q

\{P\} repeat S until b \{Q\}

17. initialization: \(P \Rightarrow I\)

Does \((x \geq 0 \land d = y \land k = 0) \Rightarrow d = y - k\)?
Yes, since \(d = d - 0\).

preservation: \(\{I \land b\} \Rightarrow \{I\}\)

\(\{d = y - k \land k \neq x\}\)
\(d := d - 1\)
\(\{d + 1 = y - k \land k \neq x\}\)
\(k := k + 1\)
\(\{d + 1 = y - (k - 1) \land (k - 1) \neq x\} = \{d = y - k \land (k - 1) \neq x\} \Rightarrow I\)

finalization: \((I \land \neg b) \Rightarrow Q\)

Does \((d = y - k \land k = x) \Rightarrow d = y - x\)? Yes, clearly.

18. We need to show \(P \Rightarrow \wp(S, Q)\).

\(\wp(S, Q) = \wp(\text{if } x < 0 \text{ then } x := x + 10 \text{ end_if}, x = 8)\)

\(= (x < 0 \land \wp(x := x + 10, x = 8) \lor (x \geq 0 \land x = 8)\)

\(= (x < 0 \land x + 10 = 8) \lor (x = 8)\)

\(= (x = -2) \lor (x = 8)\)

Does \((x = 8) \Rightarrow (x = -2) \lor (x = 8)\)? Yes, clearly. Therefore, the assertion holds.

19. The other three are: maintainability, survivability, and error tolerance.

20. The need for fault/error tolerance means that dependable systems have to include redundant code to help them monitor themselves, detect erroneous states, and recover from faults before failures occur. This affects the performance of systems, as additional checking is required each time the system executes.

21. a. RELIABILITY: The probability of failure-free operation over a specified time period (in a given environment, for a specific purpose).

   AVAILABILITY: The probability that a system, at a point in time, will be operational (and able to deliver the requested services).

b. He argues that a telephone exchange switch that routes phone calls is an example of a system where availability is more important than reliability. Users expect to hear a dial tone when they pick up a phone (indicating switch availability), and faults that occur while a connection is being set up are often corrected so quickly by the system that users may not even notice them. Furthermore, even if a call is interrupted, the consequences are usually not serious.

23. a. false; b. true; c. false; d. true; e. false; f. true; g. false, h. false