

## Midterm Key

Information and Computer Science 121

Professor Debra J. Richardson  
Spring 2000

Name: \_\_\_\_\_

Signature : \_\_\_\_\_

Student ID: \_\_\_\_\_

### Instructions

- Exams are in the distribution center.
- Students may look at their exam in the presence of the distribution center staff, but may not modify the exam in any way.
- Students can take their exam, but once taken from the distribution center, no regrading of answers will occur (only recounting, if the grade was added incorrectly).
- Students should be aware that if they want a regrade, I reserve the right to regrade the entire exam.

Professor Debra J. Richardson

question type	T/F, p. 2	SA, p. 3	SA, p 3	SA, p. 4	SA, p. 5
# of questions	1-10	11-17	18-25	26-33	34-40
worth (%)	10	16	16	29	29
grade (%)	10	16	16	29	29

total grade	100 %
-------------	-------

**True/false questions** (1 points each. Circle true or false in the left margin.)

- false    1.    Verification and validation should be a separate phase of the software development process.
- false    2.    Verification and validation techniques are all focused on code execution.
- false    3.    Test plan development should begin at the end of the design phase.
- false    4.    A cognitive walkthrough can only be performed after a system is implemented.
- true     5.    The spiral model is only applicable to large-scale software.
- false    6.    The primary activity in Software Engineering is coding.
- true     7.    Requirements analysis and specification addresses what is to be built.
- true     8.    A data flow diagram is a function-oriented system model.
- true     9.    An entity-relationship diagram is a data-oriented system model.
- true     10.   A disadvantage of using natural language to specify requirements is that it often results in ambiguous requirements.

**Short answers** (2 points each, unless otherwise indicated in parentheses after the question.)

11. What three problems in software production are indicative of the *software crisis*? (3)

- *Software is typically late, over-budget, and faulty (low quality, erroneous, buggy)*

12. In which two phases of software development are the most faults typically introduced.

- *Specification (requirements) and Design*

13. Reliability measures how well a system provides expected service over \_\_\_\_\_? (1)

- *Time*

14. Why is build and fix a bad approach to software development?

- *It is costly to make changes*
- *It is difficult to maintain*
- *Doesn't work well for large complex systems*
- *Difficult to test*

15. Requirements should be *consistent*. Name three (3) more desirable characteristics of a requirements specifications. (3)

- *Abstract,*
- *Complete,*
- *Unambiguous,*
- *Precise,*
- *Testable, (verifiable)*
- *Concise, (clear)*
- *Feasible,*
- *Even,*
- *Modifiable,*
- *No Implementation Bias*
- *Reference Tool (readable by all stakeholders)*

16. Ambiguity is one common problem with requirements specifications. Name three (3) other problems. (3)

- *Incompleteness*
- *Inconsistency*
- *Imprecision*
- *Infeasibility*
- *Unevenness*
- *doesn't address user needs*
- *implementation biased*
- *not verifiable*

17. Give one reason why should prototypes be discarded?

- *Too easy to fall into build and fix*
- *Can't be used as a legal contract*

- *Poor design*
- *Expensive to maintain*

18. What are you testing when performing a cognitive walkthrough?

- *Usability of the interface*

19. What is a system model?

- *An abstract description of the system to be developed.*

20. What type of activity is done during perfective maintenance?

- *Changes to improve the software product*

21. Why is it important to try to find faults early on in development?

- *It is less costly, more difficult to fix faults*

22. Learnability is one usability attribute. Name two (2) more.

- *Efficiency*
- *Memorizability*
- *Errors*
- *Satisfaction*
- *Robustness*
- *(not ease of use or usability or usefulness or understandability because learnability is given)*

23. Define abstraction.

- *Identify important aspects and ignore details*

24. How can a developer check a requirements specification for faults?

- *Reviews (walkthroughs or inspections, but not cognitive walkthroughs)*

25. What does the spiral model add to each phase of the software lifecycle?

- *Risk assessment*

26. What does Brooks mean by the terms *essence* and *accident*? Why are they important in the study of software engineering?(4)

- *Essence deals with difficulties inherent in the nature of software - unsolvable problems, there will be no silver bullet.*
- *Accident deals with non-inherent difficulties - amenable to breakthroughs*
- *This is important to software engineering because it distinguishes between difficulties that can be solved and those that [probably] never will.*

27. What are the four (4) standard phases of the software lifecycle? (4)

- *Requirements*
- *Design*
- *Implementation (and Integration)*
- *Maintenance*

28. Which phase of software development accounts for the highest percentage cost of overall development? What is the approximate percentage? Why does this phase account for such high percentages? (4)

- *Maintenance*
- *67% (60-70%)*
- *poor development; personnel turnover; every fix is more costly later in the lifecycle;*

29. Name a problem with team programming and a benefit. (4)

- *Problem: interface problems between components or communication problems*
- *Benefit: products too complex to develop alone; multiple perspectives; faster time to market*

30. Name two (2) benefits of separating concerns.(4)

Any two of:

- *Separation of responsibility*
- *Mastering of complexity*
- *Allows concentration on individual qualities*

31. Why is it beneficial to identify potential changes early in the software development process.(3)

- *Enables planning so that future changes are less costly and/or easier to implement*

32. Name three (3) reasons for doing a lifecycle reviews. (3)

Any distinct three of:

- *Verify that inspected artifact conforms with requirements*
- *Detect points of non-conformance between current artifact and previous one*
- *Early detection of defects in the product*
- *Discover [all] defects currently present in the product under development*
- *Detect defects in a artifact's representation*
- *Evaluate techniques and tools*
- *Measure development process*
- *Feedback for specifiers to improve*
- *Feedforward for process and quality control*
- *Detect defects in the requirements specification*
- *Measure product quality*
- *Measure the development process*
- *Visibility enhancement*
- *Early identification of infeasibility*

33. One of the reasons software development is unique is because software is intangible. Name three (3) other properties of software that make it unique compared to other products. (3)

Any distinct three of:

- *S/w is Malleable; S/w is conformable to the environment (rather than other way round)*
- *S/w applications horizons expand with h/w capabilities*
- *S/w is unprecedentedly complex*
- *S/w construction is human-intensive*
- *S/w solutions require unusual rigor*
- *S/w has discontinuous operational nature*
- *S/w is extremely portable*
- *S/w is constantly changing*
- *S/w is inherently unverifiable*
- *S/w engineering is a young discipline*  
*[No points for faulty (doesn't have to be), abstract, invisible (too close to intangible)]*

34. What is the primary purpose of using mockups (scenarios and prototypes)? (2)

- *Knowledge Acquisition and Requirements Elicitation*
- *Early requirements validation*  
[1 point for validating user interface]

35. One goal of performing lifecycle reviews is measuring product quality. Name three (3) more.  
(3)

See # 32

[1 point for listing specific qualities that can be measured]

36. One product of a requirements specification is a "Refinement of Customer Needs". Name two (2) other products contained within a requirements specification.

Any two of:

- *Documentation of all requirements and constraints (functional and non-functional)*
- *Contract between customer and developer*
- *Guide for development and/or verification*
- *Lifecycle considerations*
- *Acceptance Test Plan*

37. Name a benefit and a drawback of using formal/mathematical notations to define requirements?(4)

- *Benefit - unambiguous*
- *Drawback - difficult to understand*

38. What does Fredrick Brooks mean by the term "mythical man month"?(3)

- *Men and months are not interchangeable. Adding more men does not achieve an anticipated gain in productivity.*

39. In "Calling the shots" Brooks discusses estimating how long a system programming job can take. He claims that one can not take data from a small programming project and extrapolate it to a large programming project. According to Brooks, what must be added to the time estimate of a large programming project that is not included in a small programming project?(3pts)

Any subset of:

- *Planning*
- *Documentation*
- *Communication/Team meetings*
- *Testing*
- *System integration*
- *Training times*

40. One software quality is Correctness. Define this quality and list eight (8) other software qualities and define four (4) of them. (12)

- *Correctness - the software behaves according to the specification*
- *Reliability - the probability that software will operate as expected over specified period of time*
- *Robustness - it behaves reasonable under unanticipated circumstances*
- *Efficiency - it uses available resources economically*
- *Usability - It is easy to use*
- *Understandability - it is easy for developers to understand the produced artifacts*
- *Verifiability/Testability - if satisfaction of desired properties can be easily determined*
- *Maintainability - if it can be easily modified*
- *Repairability - defect correction is easy*
- *Evolvability/Flexibility - if facilitates addition of functionality or modification of existing functions*
- *Reusability - it can be reused with minor modification to create another product*
- *Portability - It can run in different environments with little or no effort.*
- *Interoperability - it can coexist and cooperate with other systems*
- *Productivity - measures the efficiency of the development process*
- *Timeliness - measures the ability to deliver s/w on time*
- *Visibility - all steps and its current status are documented clearly*