FINAL ORAL EXAM QUESTIONS
SPRING 2006
Last Modified: 6 April 2006
Version 1.1

Updates:

v1.1 → 1.2:
• Correction to Fall, 2004: Dr. Smith, not Prof. Roach, taught the second section of CSCI 5200

v1.0 → 1.1:
• Questions added for CSCI 5230, Fall 2003 (unexpected last-minute material from Dr. Barrett)

v0.9.3 → 1.0:
• Questions added for CSCI 5227, Fall 2005

v0.9.2 → v0.9.3:
• Questions added for CSCI 5230, Fall 2005

v0.9.1 → v0.9.2:
• Questions added for CSCI 5300, Spring 2006

v0.9 → v0.9.1:
• Inappropriate questions removed from CSCI 5200 question set for 2004-2005.
• CSCI 5230 question set for 2002-2003 designated for 2004 as well

Note: you don’t have to study all of the questions on this sheet. You only need to study the set of questions that was created for the semester in which you took a class, and for the instructor from which you took that class.

If you took a class in a semester for which questions are not available, please choose the set of questions that seems like the best match with what you were taught. If there are no good matches between what you were taught and the questions in this exam, please contact Dr. Pfeiffer immediately (phil@etsu.edu), to discuss alternative strategies for managing the exam.

Courses for which people are normally responsible:

• ACS Majors: 5200, 5230, 5300, 5150, 5220, 5250, 5620
• IT Majors: 5200, 5230, 5300, 5360, 5460, 5710, 5720
1. Describe the general steps for Remote Procedure Calls to occur.

2. What kind of services does X.500 provide?

3. What characteristics should a process possess to be a good candidate for process migration?

4. What are the advantages of TCP over UDP? ... of UDP over TCP?

5. What is the ontology problem in agent systems, and why is it hard to solve?

6. Describe how the Bully Algorithm works for distributed coordinator election.

7. What advantage does a 2-tiered distributed system have over a 3-tiered system, and vice versa?

8. What are the performance advantages of kernel-level threads over user-level threads?

9. What resources does a Quality of Service system likely have control over?

10. Describe the end-to-end argument. Give an example of its use.

11. In general, describe the differences between a Distributed Operating System, a Network Operating System, and a Middleware System.

12. What is the purpose of message queueing systems?

13. What are Lamport time stamps used for?

14. Explain the difference between strict consistency and sequential consistency.
1. Critique and discuss the following statement: “Software engineering is simply the development of software.”

2. Consider these life-cycle models: Linear sequential, Prototyping, RAD, Spiral, Incremental, Component-based development, and Formal methods. For each situation below, which model best suits it? Elaborate.
   a. A large in-house project to replace an existing software application
   b. A 3D role playing PC based game.
   c. An aircraft navigational system that is mounted in the helmet of a f15 fighter pilot.

3. Explain what is wrong with the following set of requirements.
   a. When required by users, the system shall display information about a particular book.
   b. CMT generates reports for the administrator, which contains user information.
   c. If the user information is found then the system displays the first name, last name, and user-type (admin or user) in a dialog box.

4. During requirements elicitation, how would you ensure that the requirements are correct?

5. What is the major difference between requirements analysis and requirements validation?

6. Differentiate between static and dynamic requirement specifications techniques and give some examples.

7. Why might Joint Application Design or Rapid Application Design be better at achieving completeness of requirements than structured interviews?

8. Your new boss assigns you to write a set of use cases. Why should you ask her about each of the following issues?
   Use case style? Use case ceremony?

9. Statistical quality assurance activities are well understood in manufacturing, where measurement, tolerances, and timing are related to physical products. How can statistical methods work in software engineering? Give one example.

10. Why is getting the level right so important when writing use cases?

11. Domain knowledge is critical to identifying, specifying, and understanding requirements. Give two methods for handling the problem of domain knowledge.

12. Describe the role of pre- and post-conditions in a use case.

13. Some people have argued that the details of the user interface should not be included in the SRS. Support or deny that assertion.
1. In a Software Requirements Specification (SRS) document, the writer is given the choice of several styles for writing functional requirements. Describe one style and its advantages.

2. Why might Joint Application Design or Rapid Application Design be better at achieving completeness of requirements than structured interviews?

3. Your new boss assigns you to write a set of use cases. Why should you ask her about each of the following issues? Use case style? Use case ceremony?

4. Statistical quality assurance activities are well understood in manufacturing, where measurement, tolerances, and timing are related to physical products. How can statistical methods work in software engineering? Give one example.

5. Why is tracability so important to requirements engineering?

6. What is the purpose of the following UML models: sequence diagram, class diagram, package diagram?

7. Give one advantage of the continuous model (rate each process area separately) over the staged model (rate the overall organization on all process areas) in CMMI, then give one advantage of staged over continuous.

8. Identify any potential problems with the following non-functional requirements.
   a. The system shall perform a search in less than 0.5 seconds.
   b. The system shall be composed of reusable modules.
   c. The system shall be user friendly.

8. Why is getting the level right so important when writing use cases?

9. Domain knowledge is critical to identifying, specifying, and understanding requirements. Give two methods for handling the problem of domain knowledge.

10. Agile methods claim to fix many problems associated with more traditional lifecycle models. Give two things that agile methods do that are different than traditional methods, not counting Pair Programming.

11. Describe the role of pre- and post-conditions in a use case.

12. Critique these functional requirements:
   a. The File Menu, when displayed, shall show the choices “Open File”, “Save File”, “Close File”, and so on.
   b. Users often enter bad data, so the system shall prevent it.
   c. The system shall store employee records in a binary search tree based on employee salary.

13. Some people have argued that the details of the user interface should not be included in the SRS. Support or deny that assertion.
1. Critique and discuss the following statement: “Software engineering is simply the development of software.”

2. Explain what is wrong with the following set of requirements.
   a. When required by users, the system shall display information about a particular book.
   b. CMT generates reports for the administrator, which contains user information.
   c. If the user information is found then the system displays the first name, last name, and user-type (admin or user) in a dialog box.

3. During requirements elicitation, how would you ensure that the requirements are correct?

4. What is the major difference between requirements analysis and requirements validation?

5. For the structured paradigm and the object-oriented paradigm, explain the role of analysis models in the requirements engineering process including what they are, what they communicate, their objectives, and their focus.

6. Why might Joint Application Design be better at achieving completeness of requirements than structured interviews?

7. Why is getting the level right so important when writing use cases?

8. Critique and discuss the following statement: "Domain knowledge is critical to identifying, specifying, and understanding requirements." Include in your discussion ways to handle the problem of domain knowledge.

9. Describe the role of pre- and post-conditions in a use case.
1. What are Archive Logs? Why would you want to enable Archive Logs? What are the advantages/disadvantages?

2. Explain Rollback Segments. What are they? When are they generated? When are they used?

3. Describe the lost update problem in concurrency control.

4. Define deadlock, and state a method commonly used by DBMS to deal with the problem.

5. When accessing information from an unsorted data file, why would it be beneficial use multi-level indexes? Explain your answer.
1. Some would argue that software is inherently different than other products, making software quality harder to achieve than for other products. Give one such inherent property, tell why it would (if true) make software quality harder to achieve, then give one argument why this might not be the case.

2. What makes usability difficult (but not impossible) to specify?

3. Explain Beizer’s Pesticide Paradox.

4. Why do some people claim that quality is “free”?

5. What role should SQA play in dealing with subcontractors?

6. What aren’t generic review checklists (e.g. with items like, “use cases should always list pre-conditions”) enough to base reviews on?

7. List one way in which audits are more formal than inspections.

8. In model testing, what is external consistency? Give an example.

9. What are the implications for black box testing when using Design-by-Contract?

10. Choose the order in which you would test the following kinds of functions for a class, and justify your choices; assume that getters are trusted. {overloaded constructor, conversion-from-class, destructor, default constructor, > operator}

11. Describe two usability test cases (note: not functionality tests) for the “Open file” options in Microsoft Word.

12. Briefly describe Test-Driven Development.

13. Why do XP-ers claim that refactoring can be trusted more easily in XP than in traditional development?

14. What makes a user interface “intuitive”?

15. Components tend to be more self-contained than objects, which should make them easier to test. Give one reason why this might actually make components *harder* to test.

16. Under what conditions might it be difficult to reproduce an error seen in a test script?

17. How can error metrics be used to improve process?

18. What are two ways to control the exponential nature of collaboration testing?

19. Why is a reduction in the number of reported errors over some time period *not* a good indicator of the quality of the system?
1. Given that no test technique (e.g. equivalence class partitioning, test factor analysis etc…) can assure the test team they are choosing test cases that will find all the bugs in the software, why do we use them?

2. Can you expect to more efficiently create a high quality product if you have better developers or better testers?

3. What makes usability difficult (but not impossible) to specify?

4. Explain Beizer’s Pesticide Paradox.

5. Why do some people claim that quality is “free”?

6. What is the main benefit to the test team when a Risk Analysis is done?

7. List one way in which audits are more formal than inspections

8. Describe two usability test cases (note: not functionality tests) for the “Open file” options in Microsoft Word.

9. Briefly describe Test-Driven Development.

10. Why do XP-ers claim that refactoring can be trusted more easily in XP than in traditional development?

11. What makes a user interface “intuitive”?

12. How can error metrics be used to improve process?

13. What are two ways to control the exponential nature of test combinations when multiple factors are involved?

14. Why is a reduction in the number of reported errors over some time period *not* a good indicator of the quality of the system?

15. What is the ultimate goal of the test plan?

16. What are the primary advantages of Software Test Automation?

17. Why does the use of Capture Replay as an automated testing method tend to fail?

18. What is the worse thing that can happen in the execution of an automated test case (and is more likely to happen if the testware is poorly written?)

19. What is the goal of threat modeling as it pertains to security testing?

20. What is Grey-Box testing and why is it more common with web-site testing
1. Name, and briefly describe, Watts Humphrey’s 5 levels of process improvement.

2. Describe the key process areas to move from level two CMM to level three CMM. (This question can be asked for the relation between any two adjacent levels)

3. Describe some difficulties with the CMM.

4. Boehm identifies Parkinson, Price-to-win, Analogy, Top-down, Bottom-Up, Expert Judgment, and Algorithmic Strategies as the seven major techniques for estimating software costs in use today. Describe N of these strategies giving their strengths and weaknesses.

5. In what ways, if any, are algorithmic models subjective, rather than objective?

6. Humphrey talks of Wide band Delphi as an improvement over expert judgment. Describe the steps in Wide Band Delphi.

7. Explain the difference between SLOC, weighted SLOC, and function points.

8. Discuss one strength and one criticism of each of the following models of project estimation: SLOC, COCOMO, function points.

9. What is a Rayleigh curve and how is it used in project estimation?

10. Pick three criteria that characterize "good project estimation plans". For each of these criteria, discuss, briefly, one thing that could go wrong if the criterion you pick is not observed.

11. What three factors are considered in the COCOMO model of project estimation?

12-14. Describe the process / environmental/ personnel adjustments used to model impacts on productivity in COCOMO.

15. What is meant by the notion of "organic" and "embedded" projects? Which did COCOMO consider more difficult, and why?

16. COCOMO is an exponential model of project estimation. What was the rationale for defining the model as an exponential one?

17. What 2 weaknesses of COCOMO is COCOMO II designed to address?

18. Name and describe each of the elements that are counted in an FP analysis.

19. What are the basic steps of Risk Management according to Boehm?

20. What is Risk exposure?

21. What is Risk Leverage and how is it calculated?

22. What is an isolation layer and what risks does it mitigate?

23. How does risk mitigation differ from risk transfer? Give examples of each.
24. What risks does Cockburn associate with relational databases? What strategies does he suggest using to overcome these risks?

25. What role do thresholds play in Hall's model of risk management? What is the risk of not using thresholds?

26. What are the elements of a risk action plan?

27. Describe the form and discuss the use of the following work visualization tools in project planning.
   - Work Breakdown Structure (WBS)
   - Gantt chart
   - PERT chart

28. What is a critical path and what is a weakness of critical path analysis?

29. Give one argument for, and one argument against, detailed tracking of software projects.

30. It is important to manage change to control its bad features. What are the bad features of change CM tries to control?

31. McConnell and other describe a five step procedure for handling change control. What are those steps?

32. Describe 3 factors used in evaluating a change request-

33. What is the difference in auditing/ measuring progress in an XP project and a spiral model project?

34. Name and describe the application, functional, matrix, and project structures for project management, including their respective strengths and weaknesses as organizations for SPM.

35. What is the difference between a team structures in an XP project and an typical chief programmer team, a democratic team?
1. Name, and briefly describe, Watts Humphrey’s 5 levels of process improvement.

2. Describe the key process areas to move from level two CMM to level three CMM. (This question can be asked for the relation between any two adjacent levels)

3. Describe some difficulties with the CMM.

4. Boehm identifies Parkinson, Price-to-win, Analogy, Top-down, Bottom-Up, Expert Judgment, and Algorithmic Strategies as the seven major techniques for estimating software costs in use today. Describe N of these strategies giving their strengths and weaknesses.

5. Discuss one strength and one criticism of each of the following models of project estimation: SLOC, COCOMO, function points.

6. Describe, in general, the COCOMO model of project estimation.

7. Describe, in general, the function point model of project estimation.

8. What are the basic steps of Risk Management according to Boehm?

9. Describe two risk mitigation techniques.

10. What are the basic steps for change control?

11. Describe and critique the XP Planning Game.

12. What are the goals of a project retrospective?

13. Give one measurement that will help answer the question, “Are our project’s code reviews effective?”

14. Give an example of a project “information radiator”. Of what use is one?

15. Give one advantage and one disadvantage of a group that makes decisions by consensus.

16. Name three attributes of software projects that, according to Cockburn, tend to decrease the need for ceremony (or project weight).

17. Describe the form and discuss the use of the following work visualization tools in project planning.
   - Work Breakdown Structure (WBS)
   - Gantt chart
   - PERT chart

18. What is a critical path and what is a weakness of critical path analysis?

19. Give one argument for, and one argument against, detailed tracking of software projects.

20. It is important to manage change to control its bad features. What are the bad features of change CM tries to control?
1. Describe briefly the four phases of Software Project Management as discussed by Frank Tsui.

2. What kind of projects is best suited for agile METHODS?

3. Describe briefly any four of the following agile methodologies: Scrum, Dynamic Systems Development Method (DSDM), Crystal Methods, Feature Driven Development (FDD), Lean Development (LD), Extreme Programming (XP), or Adaptive Software Development (ASD).

4. What are the elements of a quick estimate?

5. Name and describe N (N>7) project deliverables that project managers might be asked to deliver.

6. Give a brief explanation of a work breakdown structure (WBS)?

7. What is a project milestone?

8. What is G/Q/M (Goal/Question/Metric)?

9. Differentiate between metric and measurement.

10. Differentiate between a hierarchical organization and a matrix organization.

11. Differentiate between software process and software methodology. Use the design process as an example to further clarify the difference.

12. What are the three main risk analysis and planning activities? Give a very short explanation of each.

13. Give a brief explanation of a function point.

14. Name and describe each of the elements that are counted in an FP analysis.

15. What is the general process of using COCOMO?

16. Name the four team maintenance activities.

17. How should punishment be handled? How does this contrast to rewarding?

18. Briefly explain the idea of “chaordic.”

19. Comment on Highsmith’s view that “working software is the primary measure of progress.”

20. According to Highsmith “Agile approaches shine when speed, uncertainty, and change characterize the project environment.” Name the six key areas that are to be adopted when utilizing an agile methodology (according to Highsmith).
1. How does the database design process differ from the software design process?

2. What mechanisms can be used to map a relationship in ER to a relational DB or an OO Model?

3. Name some of the major problems encountered when trying to find data requirements in a natural language document.

4. Describe 1st and Boyce Codd Normal Forms:

5. Describe the problems caused by tables not in BCNF:

6. Given an object-oriented model with what approaches could you use to map a super class and set of subclasses to tables in a relational model?

7. How would you chose among the three approaches (Localize, Link and Collapse) in translating a super type and set of subtypes to a relational model?

8. Contrast the top down, bottom up and inside out (oil stain) approaches to database design.

9. When merging two sub-schemas together items that store information about the same real world things fall into three classes. Name the classes and how to deal with them.

10. How do Object-Relational databases complicate design problems?
1-16. Define and explain the significance of the following terms.

1. stamp coupling
2. refactoring
3. future (i.e., in the context of nonblocking systems design)
4. constructor method
5. iterator class
6. façade
7. broker
8. content coupling
9. control coupling
10. model-view-controller
11. temporal cohesion
12. logical cohesion
13. sequential cohesion
14. opaque reference
15. procedural cohesion
16. publish-subscribe

17. Page-Jones distinguishes between four kinds of classes—foundation classes, architectural classes, business classes, and application classes. What practical rationale does Page-Jones cite for this distinction? And what does this ordering among these four sets of classes reflect?

18. Is RTTI, as a rule, a sign of a good design? Justify your answer. If you say yes, state one strategy for promoting RTTI; if not, one strategy for removing it.

19. State the conditions that the principle of substitutability imposes on the relationship between a class FOO and a derived class BAR, relative to a public method in FOO, FOO::M.

20. State the principle of once and only once, as it relates to software design—and briefly explain its importance.

21. Is high fan-in a sign of a good design? What about high fan-out? In each case, justify your answer.

22. Name and describe the five defining characteristics of an object.

23. How does the use of floating point impact low-level design?
1. If you were given a program’s source code to evaluate based on correctness, robustness, flexibility, readability, and reusability, what are some the questions you would ask when evaluating the code? [The answer is acceptable if the student can identify at least 2 questions for each criterion.]

(2 – 4) Define and explain the significance of the following terms.

2. Refactoring

3. Iterator class

4. Model-view-controller (or MVC)

5. State in a single sentence or phrase the purpose of creational design patterns, structural design patterns, and behavioral design patterns.

6. When software components are reused within an environment, the environment needs to find out the details of the modules (meta-data) that make up the components. What is the process of obtaining the meta-data called?

7. Differentiate between distributed and centralized storage architecture.

8. When would you typically use repository architectures?
1. Assume a user types in the following URL to their browser: http://www.yahoo.com. In as much detail as possible, describe what follows up until the page is rendered. You must refer to at least these protocols: ARP, DHCP, DNS, Ethernet, HTTP, IP, TCP, UDP; you must also provide a careful description of the nature of any network traffic related to the browsing with respect to these protocols.

2. Assume ETSU suffered a total loss of server and network systems, but that you have full backups available. List the first 5 services you would get up and running (in order) and explain why you have that order -- you should assume power has been restored before the Physical Plant turns control over to you.

3. Assume you are hired as the OIT helpdesk manager. What statistics would you want to see and why? Be sure to explain how they will be useful and how you will use them.

4. Assume that you are in OIT and have to prioritize the following projects for the summer. Perform a resource estimate, along with a magic quadrant analysis, and then a project schedule (taking into account priorities, resources, and dependencies). You should give reasonable time estimates for each of these tasks and assume a resource mix of 2 full-time student workers and 1 full-time senior System Administrator

1- Switch out 4 labs of 25 computers each
2- Produce a new image for Fall
3- Replace 100 faculty desktops
4- Upgrade Blackboard to a new (minor release) version
5- Upgrade the hardware on imail
1-8. Define and explain the significance of the following terms.

1. nonrepudiation
2. Rijndael
3. ICMP directed broadcast
4. Chain of custody
5. TEMPEST
6. Diffie-Hellman
7. rootkits
8. need to know

9.-22. Explain the similarity and difference between the following pairs of related terms.

9. authentication and authorization
10. anonymity and pseudonymity.
11. message authentication codes (MAC) and digital signatures (explain this one carefully)
12. symmetric and asymmetric encryption
13. statistical and rule-based IDSes
14. worms and viruses
15. viruses and Trojan horses
16. Trojan horses and worms
17. backdoors and Trojan horses
18. steganography and cryptography
19. accounts and community strings
20. connect scan and SYN scan.
21. a nonce and a salt

23. Name and define the three basic requirements for system security.

24. Describe the three basic approaches for doing authentication, and give one example of each approach.

25. Explain the operation, and discuss the relative frequency, of stack smashing attacks.

26. Are methods for session-oriented cryptography (e.g., SSL) private or public key cryptosystems? Explain, and give the rationale behind the standard strategies for doing real-time encryption.

27. Let
   - M be a message
   - RSA-Transform( Key, M ) denote the transformation of message M, using key K
   - Key Abbott-public be Abbot’s public key in an RSA encryption scheme
   - Key Abbott-private be Abbot’s private key in an RSA encryption scheme
   - Key Costello-public be Costello’s public key in an RSA encryption scheme
   - Key Costello-private be Costello’s private key in an RSA encryption scheme
Using the notation and the transformations provided above, show the minimum amount of work that Abbott must do to send a message to Costello that
   - Costello can confidently assume came from Abbott
   - Costello can confidently assume came from Abbott, and that remained secret during transmission
In each case, show what Costello must do to decrypt Abbott’s messages. Note: assume that all private keys are kept private, and that Abbott and Costello have reliably exchanged public keys.
1. Given an example of a reflexive, symmetric and transitive relation on the integers

2. Show the sum from 1 to N of 4i is equal to 2N*(N+1)

3. Define big O, big Omega, big Theta of function F,

4. Explain the relationship between a recursive function and a recurrence relationship.

5. Explain the use of pre and post conditions on the analysis of an algorithm and how the concept relates to software testing.

6. Explain these three types of recursion
   a. Chip and Conquer
   b. Chip and be conquered
   c. Divide and conquer

7. Explain why probabilities play a part in determining average case performance calculations
1. Describe the two main steps of induction; the three steps required to prove a loop correct; and how the two processes are related.

2. Explain the relationship among the following concepts: divide and conquer, recursion, recurrence, and masters method.

3. What is the main advantage of dynamic programming over divide and conquer?

4. How does amortized analysis differ from worst case running time? When is amortized analysis used? How do these concepts relate to aggregate analysis and the accounting method?
1. Explain SSL and what is it used for.

2. List basic components required in E-Commerce (such as session control).

3. Explain session hijacking

4. Explain the role of the merchant account in E-Commerce

5. Explain how sessions are used and give an example of an E-Commerce component that uses sessions.

6. What are the differences between a load balancer and a proxy server?

7. What are ways that companies attempt to determine the ROI of their web sites?

8. How does public/private key encryption work? (How would Joe send Jane an encrypted message and how would Jane decrypt it?)
1. Explain (or define) E-Business and its components

2. Explain what a "Business Plan" is and why it is important for success.

3. How can portals enhance workers productivity?

4. Explain how CRM (customer relationship management) is enhanced through collaboration tools.

5. Explain one of the roles that KM (knowledge management) plays in E-Business.

6. What is the meaning of pervasive computing and how does this impact E-Business?

7. Describe (or draw) a multi-tier architecture and talk about the different components. [Consider logical vs physical tiers]

8. How can XML play a part in “Enterprise E-Business?”