Requirements

1. Can you name a number of non-functional (or quality) requirements?
2. What is your advice when a customer wants high performance, high usability and high security?
3. Can you name a number of different techniques for specifying requirements? What works best in which case?
4. What is requirements tracing? What is backward tracing vs. forward tracing?
5. Which tools do you like to use for keeping track of requirements?
6. How do you treat changing requirements? Are they good or bad? Why?
7. How do you search and find requirements? What are possible sources?
8. How do you prioritize requirements? Do you know different techniques?
9. Can you name the responsibilities of the user, the customer and the developer in the requirements process?
10. What do you do with requirements that are incomplete or incomprehensible?

Functional Design

1. What are metaphors used for in functional design? Can you name some successful examples?
2. How can you reduce the user's perception of waiting when some functions take a lot of time?
3. Which controls would you use when a user must select multiple items from a big list, in a minimal amount of space?
4. Can you name different measures to guarantee correctness of data entry?
5. Can you name different techniques for prototyping an application?
6. Can you name examples of how an application can anticipate user behavior?
7. Can you name different ways of designing access to a large and complex list of features?
8. How would you design editing twenty fields for a list of 10 items? And editing 3 fields for a list of 1000 items?
9. What is the problem of using different colors when highlighting pieces of a text?
10. Can you name some limitations of a web environment vs. a Windows environment?

Technical Design

1. What do low coupling and high cohesion mean? What does the principle of encapsulation mean?
2. How do you manage conflicts in a web application when different people are editing the same data?
3. Do you know about design patterns? Which design patterns have you used, and in what situations?
4. Do you know what a stateless business layer is? Where do long-running transactions fit into that picture?
5. What kinds of diagrams have you used in designing parts of an architecture, or a technical design?
6. Can you name the different tiers and responsibilities in an N-tier architecture?
7. Can you name different measures to guarantee correctness and robustness of data in an architecture?
8. Can you name any differences between object-oriented design and component-based design?
9. How would you model user authorization, user profiles and permissions in a database?
10. How would you model the animal kingdom (with species and their behavior) as a class system?

Construction

1. How do you make sure that your code can handle different kinds of error situations?
2. Can you explain what Test-Driven Development is? Can you name some principles of Extreme Programming?
3. What do you care about most when reviewing somebody else's code?
4. When do you use an abstract class and when do you use an interface?
5. Apart from the IDE, which other favorite tools do you use that you think are essential to you?
6. How do you make sure that your code is both safe and fast?
7. When do you use polymorphism and when do you use delegates?
8. When would you use a class with static members and when would you use a Singleton class?
9. Can you name examples of anticipating changing requirements in your code?

10. Can you describe the process you use for writing a piece of code, from requirements to delivery?

Algorithms

1. How do you find out if a number is a power of 2? And how do you know if it is an odd number?
2. How do you find the middle item in a linked list?
3. How would you change the format of all the phone numbers in 10,000 static html web pages?
4. Can you name an example of a recursive solution that you created?
5. Which is faster: finding an item in a hashtable or in a sorted list?
6. What is the last thing you learned about algorithms from a book, magazine or web site?
7. How would you write a function to reverse a string? And can you do that without a temporary string?
8. What type of language do you prefer for writing complex algorithms?
9. In an array with integers between 1 and 1,000,000 one value is in the array twice. How do you determine which one?
10. Do you know about the Traveling Salesman Problem?

Data Structures

1. How would you implement the structure of the London underground in a computer's memory?
2. How would you store the value of a color in a database, as efficiently as possible?
3. What is the difference between a queue and a stack?
4. What is the difference between storing data on the heap vs. on the stack?
5. How would you store a vector in N dimensions in a datatable?
6. What type of language do you prefer for writing complex data structures?
7. What is the number 21 in binary format? And in hex?
8. What is the last thing you learned about data structures from a book, magazine or web site?
9. How would you store the results of a soccer/football competition (with teams and scores) in an XML document?
10. Can you name some different text file formats for storing unicode characters?

Testing

1. Do you know what a regression test is? How do you verify that new changes have not broken existing features?
2. How can you implement unit testing when there are dependencies between a business layer and a data layer?
3. Which tools are essential to you for testing the quality of your code?
4. What types of problems have you encountered most often in your products after deployment?
5. Do you know what code coverage is? What types of code coverage are there?
6. Do you know the difference between functional testing and exploratory testing? How would you test a web site?
7. What is the difference between a test suite, a test case and a test plan? How would you organize testing?
8. What kind of tests would you include for a smoke test of an ecommerce web site?
9. What can you do reduce the chance that a customer finds things that he doesn't like during acceptance testing?
10. Can you tell me something that you have learned about testing and quality assurance in the last year?

Maintenance

1. What kind of tools are important to you for monitoring a product during maintenance?
2. What is important when updating a product that is in production and is being used?
3. How do you find an error in a large file with code that you cannot step through?
4. How can you make sure that changes in code will not affect any other parts of the product?
5. How do you create technical documentation for your products?
6. What measures have you taken to make your software products more easily maintainable?
7. How can you debug a system in a production environment, while it is being used?

8. Do you know what load balancing is? Can you name different types of load balancing?
9. Can you name reasons why maintenance of software is the biggest/most expensive part of an application's life cycle?
10. What is the difference between re-engineering and reverse engineering?

Configuration Management

1. Do you know what a baseline is in configuration management? How do you freeze an important moment in a project?
2. Which items do you normally place under version control?
3. How can you make sure that team members know who changed what in a software project?
4. Do you know the differences between tags and branches? When do you use which?
5. How would you manage changes to technical documentation, like the architecture of a product?
6. Which tools do you need to manage the state of all digital information in a project? Which tools do you like best?
7. How do you deal with changes that a customer wants in a released product?
8. Are there differences in managing versions and releases?
9. What is the difference between managing changes in text files vs. managing changes in binary files?
10. How would you treat simultaneous development of multiple RfC's or increments and maintenance issues?

Project Management

1. How many of the three variables scope, time and cost can be fixed by the customer?
2. Who should make estimates for the effort of a project? Who is allowed to set the deadline?
3. Do you prefer minimization of the number of releases or minimization of the amount of work-in-progress?
4. Which kind of diagrams do you use to track progress in a project?
5. What is the difference between an iteration and an increment?
6. Can you explain the practice of risk management? How should risks be managed?
7. Do you prefer a work breakdown structure or a rolling wave planning?

8. What do you need to be able to determine if a project is on time and within budget?
9. Can you name some differences between DSDM, Prince2 and Scrum?
10. How do you agree on scope and time with the customer, when the customer wants too much?