

## 101 Software Engineering Interview Questions

### **Goal and Introduction**

This Technical Report grew out of an interview I had with a wonderful group of software developers at Sprague Energy. We hit it off immediately and by the end of the technical interview were swapping favorite interview questions. It occurred to me that there are many lists of interview questions for software developers around the web, but most concentrate on the latest buzzword technology or methodology for programming rather than on the larger and (in my opinion) richer world of software engineering.

Technology is the easy part of software development: there are only a few things computers can actually *do*, mostly shuffling bits around, and most new technologies are just another way of making them do that in a slightly different way or a slightly different environment. With very few exceptions, UI is UI, data is data, and whether the UI is in *curses* or HTML5 with jQuery, the data in punched cards or in JSON, is a flavor of the month. Despite this, it is convenient to have concrete language to express a general concept. This list uses the Microsoft .NET environment as its vehicle for expression, because that is both popular and where my most recent experience lies.

Technology changes, and this is the way of the world: knapped stone was replaced by hardened copper which in turn was replaced by bronze and then iron and steel and CNC lasers and now by plasma cutters and in the future doubtless by infused nanomachines or something equally shiny and “revolutionary.” Despite the technological changes, cutting off your nose to spite your face remains a bad idea, measuring twice and cutting once remains a good one, a square cut remains a typical success criterion, and an ancient Egyptian pyramid builder would recognize the purpose of most construction tools used today.

Each question has a level indicated, a roman numeral corresponding to that in a “Software Engineer *x*” title. In rough translation to organizations that use other titles, a II is a developer, a III is a senior developer, a IV is a project lead, a V a program lead, and a VI a technical director. Typically a candidate’s “level” is fairly consistent from section to section, so earlier sections give clues as to where to start probing in the later sections.

Finally, there is no answer key. If you as an interviewer cannot expound on one of these questions for at least a few minutes, perhaps you should not ask it; and if you do not know what one of these question is referencing, perhaps you might enjoy finding out.

I hope you enjoy reading this as much as I have enjoyed writing it, and that it sparks creativity and insight in your hiring interviews.

### **Basic OO**

1. [I] Describe what are generally considered the fundamental attributes of object oriented design and development.
2. [II] Describe the differences between an interface and an abstract class in a class-based language. How does the introduction into C# of extension methods affect this difference?
3. [III] Describe the differences between a class-based OO language (such as C# and Java) and a prototype-based OO language (such as JavaScript and Ruby).
4. [IV] Class A includes a collection of elements of class B. Class B includes a collection of elements of class C. It is necessary to provide an `IEnumerable<C>`-valued property in class A that enumerates all elements of type C collected in the elements of type B collected in an instance of type A.

© 2014 Brian Hetrick. This document may be used and copied in accordance with the terms of the Creative Commons Attribution-ShareAlike 3.0 United States license. The text of this license is available at: <http://creativecommons.org/licenses/by-sa/3.0/us/>

The `IEnumerable<C>` provided by the property in A must publish `CollectionChanged` events whenever the sequence of C items changes. `IEnumerator<C>` objects produced by the `IEnumerable<C>` object must throw `System.InvalidOperationException` whenever a change to the sequence of items enumerated occurs between calls to `MoveNext` without an intervening call to `Reset`. Describe the approach you would take.

5. [V] Design a thread-safe queue class with a specified maximum number of elements which causes a thread invoking the `Dequeue` method to wait until a queue element is available, and a thread invoking the `Enqueue` method to wait until the queue contains fewer than its maximum number of elements. What synchronization primitives are required? Explain how you would handle an absence of queue readers. Explain how you would handle an absence of queue writers.

### Basic Computer Science

6. [I] What does big-O notation, for example  $O(n \log n)$ , mean? (Extra points for the formal definition.)
7. [II] Name the two hard things in computer science.
8. [III] Describe the mechanism of deadlock. Describe three possible interventions to a design that permits deadlock that provably prevent deadlock.
9. [IV] Describe the differences between an AVL tree and a B tree, and when each would be preferred.
10. [IV] Pronounce the names Edsger Dijkstra, C A R Hoare, Donald Knuth, John von Neumann, and Niklaus Wirth.
11. [V] How do heaps relate to A\* searches?
12. [V] Edsger Dijkstra famously said, "Computer Science is no more about computers than astronomy is about telescopes." Discuss.

### C#

13. [I] When are instance constructors called?
14. [I] Name three categories of classes you cannot instantiate.
15. [II] When are class (static) constructors called?
16. [II] When are object finalizers guaranteed to have been called?

17. [III] What methods of `System.Object` are typically overridden when defining a new class? Why does class definition typically not involve operator overriding?
18. [III] What does the `System.FlagsAttribute` class do?
19. [III] Describe the difference, the mechanism by which the difference is imposed, and the impacts on use of the declared item, between the class member declarations:

```
public System.EventHandler EventName;
and
public event System.EventHandler EventName;
```

In the above, is `EventName` a value object or a reference object? What is its type? What is its parent type?

20. [III] What are the differences between the `unsafe` and `unchecked` keywords, and when would you use each?
21. [III] Explain the differences between value objects and reference objects, and how these affect their usage.
22. [III] Explain the differences between `ref` and `out` parameters, at both the call site and the implementing method. Why is there no `in` parameter passing keyword?
23. [IV] Describe the differences between a `public const` value and a `public static readonly` value. Explain how these affect their usage at compile time and at run time.
24. [IV] Explain the differences among an anonymous delegate, a lambda expression, and a method used as a delegate. Explain how these affect their usage.
25. [IV] Explain the difference between a method in a generic class and a generic method in a class.
26. [V] Explain contravariance and covariance in generic type arguments. How are these indicated in a declaration?

### Database

27. [I] Explain the differences between optimistic and pessimistic locking. Explain the design idiom typically used to implement optimistic locking at the

## 101 Software Engineering Interview Questions

application level. Explain the NHibernate or Entity Framework implementation of optimistic locking. Explain how an application can recover from an optimistic locking failure.

28. [I] What is an inner join? What is a left join? What is a view?
29. [II] Explain the differences between a one-to-zero-or-one, one-to-one, one-to-many, and many-to-many relationships. Explain how each is implemented in a relational database.
30. [II] What is the effect of `SET QUOTED_IDENTIFIER ON`?
31. [II] Explain the concept of referential integrity. Describe the relational database features typically involved in enforcing referential integrity.
32. [III] What is the effect of `SET IDENTITY_INSERT tablename ON`?
33. [III] What is ISO-8601 date format, and in what circumstances do you need to use it?
34. [III] Describe how to represent a rooted tree in a relational database. Describe how to represent a general graph in a relational database.
35. [IV] A database contains two tables A and B with primary key columns AKEY and BKEY respectively. There is a many-to-many relationship between the tables implemented by a linking table R with columns AKEY and BKEY referring to the primary keys of rows in the relationship to one another.
  - a) How would you write a query to return the rows of A which are not in relationship with any row of B?
  - b) How would you write a query to return the rows of A which are in relationship with a specific row *b'* of B?
  - c) How would you write a query to return the rows of A which are in relationship with at least one row of B?
  - d) How would you write a query to return the rows of A which are in relationship to all existing rows of B?
36. [IV] It is necessary to programmatically copy a SQL Server database, including all metadata such as keys, IDENTITY attributes, indices, and constraints, from one server instance to another.
  - a) How can the program discover the required metadata?
  - b) How can the program retain the values of IDENTITY columns?
  - c) Which types of constraints must be imposed before copying table contents and which types of constraints must be imposed after copying table contents?

### **.NET Framework**

37. [I] Characterize the .NET versions which support extension methods.
38. [II] What are the methods and properties exposed by an enumerator?
39. [III] What does the `System.Runtime.CompilerServices.MethodImplOptions.Synchronized` method attribute do?
40. [III] Explain why invoking a delegate (absent other constructs) fails to provide the semantics defined for a .NET event publication. Describe the standard idiom for publishing a .NET event that meets the semantics defined.
41. [III] Explain how and why one would implement the `System.ComponentModel.INotifyPropertyChanged` interface.
42. [III] What is the `System.Nullable<T>` type used for?
43. [III] What standard idioms can be replaced by the contents of the `System.Diagnostics.Contracts` namespace?
44. [III] Compare and contrast LINQ to SQL and LINQ to Entities.
45. [III] To what .NET namespace would you look for support in developing a RESTful web service?
46. [III] To what .NET namespace would you look for support in developing a SOAP web service?
47. [IV] Typically, inability to complete an operation is signaled with an exception. Identify standard mechanisms in .NET which might be used to provide invitations for higher-level or more context-aware code to intervene when difficulty is encountered, allowing processing to continue if corrective action was taken, where otherwise execution would fail with an exception. Identify portions of

the .NET framework that already use these mechanisms in this way, if they exist.

48. [IV] Describe the interactions among type inheritance, object disposition, object finalization, and garbage collection. How does this affect the design of a type inheritance hierarchy? What design idiom is typically used to implement the `System.IDisposable` interface, and why?
  49. [IV] What is an ambient transaction? How are these useful in a SDI UI? What related constructs would be needed for the same purpose in an MDI UI, and why?
  50. [IV] Why would you use the `System.Messaging` namespace?
  51. [IV] To what .NET edition would you look for support in developing a Windows Phone application?
  52. [V] To what .NET edition would you look for support in developing a NetDuino application?
  53. [V] Describe the differences among binary serialization, general XML serialization, and data contract serialization. For each of them, describe a circumstance that requires that type of serialization to the exclusion of the others.
  54. [V] Describe the asynchronous component design pattern. Describe the fundamental misconception of the Microsoft recommended usage pattern preventing the .NET `System.ComponentModel.AsyncOperation` class from achieving its purpose. Describe an alternate usage pattern in which `AsyncOperation` achieves that purpose.
  55. [V] What is the `System.AppDomain` class, and when would you create an instance of it?
  56. [V] How many months are in the current year? (Hint: "12" is incorrect.)
- Software Engineering**
57. [I] What is the air speed velocity of an unladen swallow?
  58. [I] Describe the differences between change control and source code control.
  59. [I] Describe the three goals of data security.
  60. [III] Discuss the use of string resources in internationalization.
  61. [III] Describe the differences between a unit test and an integration test.
  62. [III] Discuss the role of the Story Review in the Agile development process.
  63. [III] How do you unit test an abstract class?
  64. [III] Describe the goal of Humphries' Personal Software Process. Does it achieve it?
  65. [III] Describe an Eulerian tour (or cycle or path).
  66. [III] What is two-factor authentication? What is its role in security risk avoidance? What is its role in compromise mitigation?
  67. [III] Discuss the numerical stability of Romberg integration.
  68. [III] Compare and contrast the tuning of PID process controllers and of fuzzy logic process controllers.
  69. [III] Describe function points. To what phases of the software development life cycle are they applicable?
  70. [III] Describe the overall process of numerically solving a set of simultaneous differential equations with boundary conditions.
  71. [III] What is dependency injection, why is it useful, and how can it be implemented in the absence of factory classes?
  72. [III] A data link between two structures a kilometer apart is to be established. Discuss the data security trade offs (in all three goals) among 20 mA current loop, TIA/EIA-485, optical fiber, free space optical, and 2.4 GHz radio.
  73. [III] Discuss the implications of typical accessibility aids to desktop software design and implementation.
  74. [III] Describe the three types of calendar in common use throughout the world. Which type is the Gregorian calendar?
  75. [III] A microcontroller with a 3.2 MHz system clock needs to send and receive standard full duplex asynchronous serial data at 19.2 kbps (166.67 clocks per bit) by bit banging GPIO lines. The microcontroller's timer module increments a timer count value every clock up to a programmable modulus value, at which point the timer count returns to zero. The timer module can be pro-

## 101 Software Engineering Interview Questions

grammed to interrupt on timer count return to zero, and can also be programmed to interrupt when the timer count equals either of two comparator values A or B. The count return to zero, count equals value A, and count equals value B interrupt vectors are distinct from one another. Each of the microcontroller's GPIO lines can be programmed to interrupt on either positive going or negative going edges; the GPIO lines share a single level change interrupt vector, but this vector is different than those of the timer module. There is no RTOS involved.

- a) Describe an overall approach to implementing the serial communications, noting particularly which operations occur during interrupt handling and which do not.
  - b) The microcontroller does not expose run/wait state information or interrupt/normal state information on its pins. Describe an approach to instrumenting the solution so that overall remaining CPU capacity can be estimated.
  - c) Discuss an approach you would explore if the microcontroller sets flags which must be polled rather than generating an interrupt, with particular attention to jitter in the serial signals.
76. [IV] Describe the Iron Triangle (Triple Constraint, Project Management Triangle). Compare and contrast its impact on waterfall project management and Agile project management.
  77. [IV] Describe the Sprint Review, Sprint Demo, and Sprint Retrospective in the Agile development process. Which are validation and which verification processes? What is each validating or verifying?
  78. [IV] Define the McCabe complexity metric. Define the Halstead volume metric. Discuss why the ratio of these metrics is a proxy measure of reliability.
  79. [IV] Discuss typical approaches to packet routing in a mesh network.
  80. [IV] Describe the four most common human perceptual and learning styles, and the prevalence of each in the population. Discuss the implications for both product UI design and business presentations.
  81. [IV] Describe the major components of risk management. Discuss the similarities and differences between risk avoidance and materialized risk (event) mitigation. Discuss the role of artifact and process metrics in risk materialization recognition.
  82. [IV] Describe Formal Inspection, sometimes called Fagan Inspection. What development process artifacts are involved? What are observed effects on immediate and total lifecycle project costs?
  83. [IV] Describe afferent coupling, efferent coupling, abstractness, and module or object instability. Explain why these are proxy measures of adaptability.
  84. [IV] Describe the various categories of module or class cohesion. Explain why these are proxy measures of maintainability.
  85. [IV] In some ORMs, mapping can be done either by applying attributes to classes and class members, or by creating separate mapping files. Which approach is quicker to implement, and why? Which approach is more reliable, and why? Which approach is more maintainable, and why?
  86. [IV] Discuss the trade-offs between authenticating a client/server HTTPS connection with HTTP basic authentication and an alternative of X.509 client certificates. Consider risks to both the organization and the user; typical attacks against both data at rest and data in motion; typical attacks against both client and server; the ability to detect both attacks and actual compromise; and both present and future implications of compromise.
  87. [IV] A *blackboard* is a global data structure used by cooperative problem solving agents called *knowledge sources*. Each knowledge source recognizes when a set of data on the blackboard represents a configuration to which it is applicable, applies itself, and finally places its *findings* on the blackboard. Knowledge source findings are typically the results of inferences or computations. The findings are then used, along with the initial data, by other knowledge sources to make further inferences or computations.
    - a) Compare and contrast this arrangement with a typical data flow architecture such as a spreadsheet.
    - b) Consider the implementation of a distributed virtual blackboard using a message queue

- with a message broker. Discuss the categories of message filter predicates this would require for subscriptions to the queue by knowledge sources.
- c) In the canonical arrangement, knowledge sources compute an *activation level* and wait for further direction before applying themselves. The activation level is an estimate of how much the result would advance the overall effort, and is used to allocate resources among the candidate applications of knowledge sources. Discuss why the result of this arrangement does not result in an A\* search of the problem space, and what modifications would be needed to make it an A\* search.
  - d) Discuss whether a relational database is an appropriate store for activation level/knowledge source candidate application pairs for resource allocation.
88. [V] Discuss the role of earned value analysis in software project management. Describe how earned value analysis can be gamed. Describe countermeasures to such gaming.
  89. [V] Describe the characteristics of each stage in the CMMI.
  90. [V] Give the observed relationship of total development cost to project schedule, including both compression and extension from the minimum overall cost schedule. (Extra points for knowing where the formulas come from.)
  91. [V] The Agile development process has been described as “Six Sigma for software.” Compare and contrast the goals, processes, and methods (including typical use of process and quality control techniques) of Agile with those of Six Sigma to assess the validity of this claim.
  92. [V] One approach to software construction is to build each assembly in a product against the lowest version of the .NET framework the assembly requires. Another approach is to build all assemblies in a project against a common version of the .NET framework. Discuss the advantages and disadvantages of each approach.
  93. [V] Compare and contrast the semantics of call by name and of the `IEnumerable<T>` interface.
  94. [V] Describe each of the subject matter areas in SWEBOK.
  95. [VI] Describe each of the subject matter areas in PMBOK.
  96. [VI] Longitudinal code base analysis uses checkin deltas to determine code added, modified, removed, and moved (possibly with whitespace changes). Discuss the patterns of these metrics expected for each of waterfall with architectural decomposition, waterfall with functional decomposition, and Agile development methodologies. Discuss how unexpected patterns can signal typical development process risk materialization events.
  97. [VI] A project will take 3 persons 18 months to perform, but line management has presented it to senior management as 3 persons for 12 months to get the project funded. Management imposes scope creep events with exponentially distributed inter-arrival times at a rate of  $\lambda$  per week and with impacts of mean  $\mu > 0$  and standard deviation  $\sigma$  labor weeks per event. Senior management will deem the project failed and ritually sacrifice the individual contributors when the project labor cost reaches 200% of the initially presented estimate, regardless of scope creep.
    - a) Explain why the cumulative impact of scope creep at time  $t$  weeks can be modeled as a normal distribution, regardless of the actual scope creep impact distribution.
    - b) Derive the project labor content distribution function as a function of  $\lambda$ ,  $\mu$ ,  $\sigma$ , and elapsed time  $t$  in weeks. (Hint: if events are generated with exponentially distributed inter-arrival times at a rate of  $\lambda$  per unit time, the number of events observed over a time  $t$  has a Poisson distribution with parameter  $\lambda t$ .) Characterize the relationship among these parameters that implies with (asymptotic) probability 1 the project labor content will increase faster than labor is applied at the described staffing level.
    - c) What rate of unpaid overtime, as a function of  $\lambda$ ,  $\mu$ , and  $\sigma$ , is required for the individual contributors to be 95% confident of surviving to project completion? Characterize the relationship among the parameters for which the necessary unpaid overtime exceeds the amount of paid time.

## 101 Software Engineering Interview Questions

- d) Assuming management perceives  $p$  units of progress (or completion) for each actual unit of progress (or completion), where  $0.5 < p < 2.0$ , what rate of unpaid overtime, as a function of  $\lambda$ ,  $\mu$ ,  $\sigma$ , and  $p$ , is required for the individual contributors to have 95% confidence that line management (aware of scope creep events and their impact) will believe at 18 months into the project that the project is 99% complete and permit the individual contributors to leave their positions with positive recommendations?

### **Critical Thinking**

98. [I] What is the difference between a duck?
99. [II] (Western culture specific) Which is better literature, *The Iliad* or *The Odyssey*, and why?
- 100.[III] (Native English speaker specific) Which is more correct when spoken by an alter of a multiple personality, where all alters have psychiatric illnesses: "None of me is sane," or "None of me are sane"? Why? Do these answers change between New York and London?
- 101.[IV] Describe the development of the character Albus Dumbledore in the Harry Potter stories.

### **Copyright and Grant of License**

This Application Note is Copyright © 2014 Brian Hetrick. It may be used and copied in accordance with the terms of the Creative Commons AttributionShareAlike 3.0 United States license. The text of this license is available at: <http://creativecommons.org/licenses/by-sa/3.0/us/> .

You might not wish to comply with the terms of the licenses granted to you above. Contact the author to make arrangements for licensing under other terms.