Question or category	**Typical question text (if different from previous column)**	**Goals when asking question**	**Caution answer example(s)**	**Strong answer example(s)**
Think back to a system you worked on 5 years ago…	“Think back to a system you worked on 5 years ago. If you were in charge of re-developing that system today, what would you do differently?”	See if candidate tends to think in big picture terms. May lead to detection of defensiveness.	“I thought the app came out fine. I can’t really think of anything I’d change.”	“We didn’t realize that our service layer wasn’t flexible enough, and we didn’t do a good job of managing evolution of versions. I’d work harder to design a service layer that fit the needs of the rest of the application.” “I would talk to the users a lot more. We really didn’t understand their task flow, so we had to iterate the interface too many times. A lot of code got thrown away.”
What's up with the alphabet soup on your resume?	“I see you have listed fifty-eleven technologies on your resume. Are you an expert in all of them?”	Find out what technologies the candidate really knows well, vs. those they only know superficially.	"Yes, I know all fifty-eleven of them."	“The recruiter told me I should put all of those on there, but I don’t know some of them that well.” Can follow up by asking the candidate to break the list down into techs they know well vs ones they don’t.
What's the weirdest bug you've ever encountered?	“What's the weirdest bug you've ever encountered? How did you solve it?” *(can preface the question with story about your own wierdest bug.)*	Probe candidate's depth. See how the candidate thinks based on how they solved the bug.	Trivial bugs, or defensive answers about the bug.	The more interesting the bug, the better. Also a good answer will show some creative thinking in the process used to solve the bug.
Questions to drill down on object orientation	“Have you ever created an object hierarchy? If so, describe the scenario and why the hierarchy was needed.” “Have you ever created an abstract interface? How would you decide	Evaluate depth in object oriented concepts. Particularly applicable to positions that have responsibility for framework design and development, but also useful to gauge whether a	Vague or evasive answers that indicate lack of deep understanding of object orientation. Such an answer might not be a showstopper if the position is junior and/or only involves tools and platforms that don’t use object	Answers that show how object orientation helped them solve a challenging development problem. Answers that indicate depth in OO understanding.

	whether to use an object hierarchy or an interface?” “Have you ever used Reflection? Why did you need to use it?” “Have you ever solved a coding problem with generics? Tell me about the problem and why generics was a good solution.”	candidate could potentially take on additional responsibilities involving OO. *These questions should only be used if the interviewer is conversant with these OO concepts.*	oriented principles heavily, such as JavaScript.	
Suppose your app needed dynamic data validation…	“Suppose your app needed to validate data, but the rules for validation were not known until run time. How would you approach that problem?”	See if candidate can think through a problem that should not be solved by throwing code at it. Find out if candidate tends to over-emphasize code, and prefers to use brute-force coding to solve development problems.	Any answer that tries to shoehorn code into a solution to the problem. Since, by definition, the rules are not known at compile time, any code based solution will be brute force, with code enforcing rules needing constant maintenance.	Any answer that recognizes the need to separate the rules into some form external to the code, just as XML or database tables, and just have the code fetch and enforce the rules at runtime.
“Imagine you're having dinner with {industry luminary such as Satya Nadella}…”	“Imagine you’re having dinner with Satya Nadella. He asks you what you don’t like about Azure and what you would like Microsoft to fix. What are your top two or three items?”	Find out how deeply a candidate understands a platform. To give a strong critique of a platform, the candidate would both need significant experience with the platform plus the ability look at the platform from a “big picture” point of view.	An answer that highlights small features of a platform as the top items shows a candidate who is more focused on routine details of development. That’s OK for a junior position, but not for a senior or architect position. Example answer: “I think the screen for renewing an Azure subscription is hard to use and should be fixed.”	An answer that highlights high level flaws in the platform. For example, “Azure’s security options are not flexible enough, and they are confusing to use. Azure security needs serious overhaul.”

If you could acquire expertise by magic, what would it be?	“If you could acquire complete expertise in some leading-edge technology instantly by magic, which one would it be?”	No one can keep up with everything. But good developers are typically aware of important areas they have not yet mastered. This question helps see if they think about new technologies in a broad sense or a narrow sense. Good developers tend to have a more broad viewpoint.	“There’s this new JavaScript framework for working with NoSQL data objects. I’d really like to learn that.” An answer like this indicates that the developer is focused on routine, day-to-day technologies. OK for some positions, especially on large teams. Generally, any answer that involves a technology that could be learned in a week or two is not an ideal answer. A bad answer would be something like: “I don’t know, really. What sort of thing would be needed for your project?” That would indicate that they don’t really think about evolution of technology very deeply at all. They are probably only suitable for heads-down work.	“I’d like to be an expert in machine learning. I think it’s going to have a big impact on lots of applications.” Good answers involve technologies that are likely to have a significant impact on development in the future, and are difficult to master.
"What's your philosophy of user interface design?"		Some developers are apathetic towards users, or even contemptuous of them. They worry almost entirely about the code aspects of development, with no particular attention to whether an app fulfills the needs of the user. This question helps uncover such an attitude.	“I build something and show it to the users for feedback.” (Indicates that they don’t talk much to users before designing and development of apps. This approach is wasteful, and rarely achieves high user satisfaction.) “I don’t really care about that. I just want to write code.” (This is an explicit admission that they don’t care about user-oriented aspects of development at all. Only fit for positions involving purely back-end or system-level work.)	Any answer that walks through a process that involves talking to users before starting development, and then gathering feedback for changes during the development process. A really great answer is one that describes a process for user experience design that involves sketching and evaluation of designs before development.

Questions on comparing platforms	“What do you see as the key advantages and disadvantages of web technologies vs desktop technologies vs. mobile technologies?” (Potentially with actual technologies included in the question, such as HTML5/JavaScript, Windows Forms, etc. Can also have variations of the question using back-end technologies)	Discover if candidate tends to be emotionally attached to a technology or two. Such developers can be a problem if you expect them to be a part of your team through a platform change.	“HTML and Javascript is fine for any business application. I don’t really see a need for those desktop platforms. And for mobile, we can just do Progressive Web Apps.”	“Web technologies fit a lot of typical business apps, but native technologies have some big advantages for mobile apps. I feel like native apps on a phone are more responsive and offer better experience to the user.” “Relational databases are the standard for most routine apps, and I prefer to use them. But there are some apps where entities have attributes that are defined on the fly. I understand that NoSQL based platforms handle that kind of data better.”
"What are the pros and cons of agile methodologies?"		Discover if candidate is fanatically attached to some particular flavor of agile. Such developers can be disruptive to a team that uses a different agile methodology.	“I think every software development project should use Scrum. I don’t see how anything gets done using anything else.”	“Agile is good for managing feedback and accountability. It helps keep the project moving. Sometimes, though, it’s hard to see opportunities to develop functionality that works across the whole application because we tend to be focused on smaller use cases.”
"How do you become familiar with new technologies?"		Since it’s inevitable that developers will need to learn new technologies, this question helps see how this candidate does it.	“I expect to be sent to a training class for any new technology.”	Any thoughtful answer that outlines resources the candidate finds helpful, such as video courses, books, online sample projects, etc.
Questions about n-tier architectures	“Diagram a typical n-tier architecture on the whiteboard, and explain the reasons for each layer.” “Has your thinking about n-tier changed in the last few years? How? What changes do you expect in n-tier architecture	Discern depth of understanding in n-tier architectures, which are needed for almost all business applications. Especially important for and candidates for a position with architectural responsibilities.	An answer that emphasizes trivial details of the n-tier architecture they use. Vague or evasive answers, which may indicate a developer who only works on the front end consuming n-tier interfaces, and doesn’t really know what’s going on behind the scenes.	An answer that shows depth of understanding of n-tier, often by showing they know what each layer is for in terms of a real application. For the last question, a good answer from a real project would be a strong indicator of real-world experience with n-tier concepts.

	because of the move to cloud-based computing?” “Have you been involved in a project where the wrong architecture caused major difficulties? What did you learn from that?”	The last of the questions can also uncover defensiveness in candidates that are supposedly architect level.		
The Kobayashi Maru question	“Suppose you are in a meeting that was called to tell you about a new module or new application you need to develop. As the meeting goes along, you start getting an intuitive estimate that it will take at least 8 months to develop. The executive running the meeting reaches the end, and says ‘We’ve promised the customer that we’ll deliver this to them in two months.’ What do you do?”	As in the Star Trek reference for the type of question, there is no right answer. It’s a test of character. Mainly, this question helps you see of a developer has the ability to deliver bad news, and ideally as soon as possible.	“Well, I would see if we could cut features, or add developers…” (Bad answer, because there’s not time to add developers. Also, with such an answer, it’s dodging the tough circumstances. So I follow up with “the decision maker says all the features are essential.”) “I’d tell them I’d do the best I could and get to work.” (Not forthright enough. Implicitly accepts that the work might be completed on time.)	“I would express my firm belief that the deadline is unrealistic. If the manager still insisted, I would tell them that I’ll get to work but I’m not promising that I’ll meet that deadline.”
Team vs. individual work	If I sense that the candidate is fudging how much work they did on something vs others on the team: "Did you do that or did the team do that?"	Can cut through resume fluffing. Also a good question to probe for defensiveness.	“I kinda worked on the whole thing, but a lot of people contributed to it.” (This answer is vague and evasive.)	“I did most of that sub-system. I got some assistance from one of our data experts to fix up the schema.”