



# Technical Interviewer's Handbook

Concise guide to conducting technical interviews

Ivan Koshelev

# Table of Contents

<b>Before the Interview</b>	3
Understanding technology	3
Planning	4
<b>During the Interview</b>	6
Questions and answers	7
Dealing with wrong answers	9
Ending the interview	10
<b>After the Interview</b>	12

# Before the Interview

Make sure that you are qualified to verify candidate's technical knowledge in technologies and concepts required by the position list you were given by HR.

Don't try to verify candidate's knowledge in technologies you don't have enough experience with. But, also, don't be too zealous declining an interview if you don't have experience with more exotic of the technologies required (unless you know for a fact that a more qualified interviewer is available for this particular case).

Your job is to verify candidate's level of understanding of 'core' technologies, English communication level and overall attitude. Most interviews will be followed by a second round conducted by the representatives of particular projects, who have a better understanding of project priorities in regards to narrow technical specializations and "exotic" technologies.

## Understanding technology

Understanding a technology means knowing its key architecture principles (some call it "philosophy") and patterns it is based on, technical implementation details, intended usage scenarios, limitations and typical problems. The interview is based on the premise that you, the interviewer, have this understanding.

"Core" technologies include:

- programming languages
- SQL databases and other data stores
- mainstream ORM frameworks
- mainstream desktop and mobile UI frameworks
- mainstream Web server frameworks
- build and continuous integration tools
- testing libraries, tools and patterns (unit, integration, E2E)
- software design patterns and principles like SOLID (not a technology, but quite important).

A given position will require a subset of this. **Tech lead** is expected to have a working knowledge of all of them, **senior** will have an idea about all and working knowledge of most,

**regular** will have working knowledge of 2-4. When estimating candidate's level of general seniority, you will have to weigh candidate's stronger and weaker sides. In an enterprise environment, knowledge of "exotic" tech rarely compensates lack of knowledge of "mainstream", unless position description explicitly states otherwise.

When you are deciding, whether a given technology can be considered "mainstream" or "exotic", your best bet is statistics. Amount of mentions on stackoverflow (<https://stackoverflow.com/tags>) and how often does this technology come up in CVs and position descriptions. You can also assess its maturity by how many books and courses are offered for it by Apress, O'Reilly, Packtpub, Safari Books, Pluralsight, etc. The more mature a technology, the deeper understanding of it would be considered "good" (though, don't forget, that there is always a ceiling of "good enough").

Also, when estimating candidate's fitness for a particular position, you should prioritize knowledge that takes time to obtain. If a candidate fits position by major technologies, but lacks knowledge of 1-2 libraries that can be obtained in less than a month — it is reasonable to give approval for a position with mention of knowledge-debt in detailed feedback and advise the candidate, which technologies they should research. Diligent candidates will often already obtain considerable knowledge by second round of interviews.

## Planning

It is important to have a plan for the interview. I recommend it to be a list of questions with short answer reference for each. Having a reference answer is important, even if you know it by heart by now. This is due to the fact, that at the end of the day both you and the candidate are just a pair of talking people.

Many candidates demonstrate such convincing confidence in their wrong answers, that it is hard not to doubt your own knowledge, especially considering that someone's employment opportunity may hinge on it. In this case, a reference answer taken from a trustworthy source is the best way to settle your doubt.

It falls within your responsibility to only use a reliable source in such cases, preferably official documentation for a given technology (i.e.: MSDN for C#, Mozilla Developer Network for JS).

This gives a good guarantee that the information is not skewed and also provides you with a terminologically correct answer. This means that it uses terminology intended by the creator and/or generally accepted by the community. Use of proper terminology is very important in day-to-day communication of technical specialists, as it eliminates ambiguity and provides brevity. Attention to this is a mark of a technical lead, and it is up to you to keep an eye for it.

A person describing a thing like:

"You set up the descriptions on those mvc classes and they route requests based on their string"

instead of:

"You add attributes to controller classes and the router will choose a corresponding controller based on the URL of the request"

may have the right idea in their head, but will have trouble explaining it to the team or making comprehensible documentation for the project.

The questions list should also be organized for flexibility. Questions should be grouped into major section i.e.: "core language", "desktop development", "web development", etc. which you can freely skip when something is not required by a position. In each section, questions should follow order from basics to advanced/complex concepts and be grouped by relation. For example, a good order and grouping in C# would be "What is a type?", "What are primitive types?", "What is a common base type for all concrete types?", "What is the most important subdivision of types?".

I recommend you to also leave space for notes near each question, where you will mark correct and wrong answers and more detailed feedback for yourself. It is best to print a fresh copy for each new candidate and fill it thought interview process.

# During the Interview

Make sure the candidate feels comfortable, this will help him or her demonstrate their best sides. Always introduce yourself to avoid being “a disembodied voice”. Make the candidate feel that they are talking to a person and are treated as equals. Before starting the interview, confirm the purpose of the call with the candidate in just a few sentences, i.e.:

“Good day, Jane, my name is Ivan Koshelev, I’m an employee of Luxoft company. I’ve been asked to conduct a technical C#/Java/JavaScript/etc. interview with you.”

“Good day, I’ve been waiting for your call.”

Sometimes candidate will inform you, that interview cannot be conducted for some reason. You must never question candidate on his motivations, and gently inform them, that it is okay, and if they want to reschedule it — they should reach out to the Recruitment specialist who originally scheduled the interview. If candidate informs you, even briefly, that they are in a situation where a conversation could be distracting and pose danger (i.e.: driving), you should immediately ask them to call you back when they are not distracted. You must never agree to carry out an interview in such circumstances.

At the start, make sure that the interview is a two-way communication process, not an interrogation. Always give the candidate an opportunity to ask you questions:

“Before we begin, do you have any questions **about the process of the interview?**”

Remember: this invitation is only for questions about the interview process, such as how long it will last, what are the main questionnaire sections, which technological knowledge is being verified. Many candidates ask questions about the projects they are interviewing for, but this information is not for you to handle. You should mention that a second round of interviews with project representatives should be expected by the candidates who pass the first round and point the candidate to Recruitment team.

# Questions and answers

During discussion, do not discourage the candidate, no matter what. You should never express negative opinion about candidate's knowledge, personal opinions, attitude or give out other signs of negative impression.

Don't forget, that communication is more than words — sighs and tone of voice can discourage the candidate and hinder his or her further performance. Over-friendliness and encouragement should also be avoided. You should stay friendly but neutral.

Remember: in this conversation you represent the company — you must maintain proper professional conduct.

Technical questions should not contain the answer within the question, or they should give multiple choices with only one or few variants being correct. Candidate should not be able to guess the correct answers.

If the candidate struggles with answering, you may give him a hint or ask a question about practical application of a theoretical concept. Hints are especially okay if position requires only regular level of knowledge. At the end of the day, your goal to verify that:

- candidate understands a given aspect/rule/behavior
- candidate has experience with its practical application
- candidate will not make mistakes associated with it
- candidate can communicate about it with other developers.

It is up to you ultimately, how much you will have to discuss a given question with the candidate to determine to what extent each statement applies to candidate's answer.

A "good" answer usually has some of the following qualities:

Answer is technically correct	Answer makes sense and information given is correct, but answer is not full or key details are missing. In this case, you can ask additional theoretical questions or questions about practical application.
Candidate understands practical application of theory	Candidate gave an example of practical application of a concept on his own accord or after an additional question. The question may be about the limits of application, i.e. ask the candidate about the result of a normal sounding operation, which would

	throw an exception, where this outcome is not obvious.
Answer covers important aspects, to the point where no additional questions are needed	Answer already covers the range of additional questions that you usually ask.
Answer is concise	Candidate covers important aspects and stops there — he does not make redundant or meaningless albeit correct statements. Exact required minimum of words and/or actions is the mark of a master.
Candidate rejects wrong answer proposals	You can make a wrong statement that sounds right and urge the candidate to agree. This tactic tests candidate's confidence as much as knowledge, so it is to be used sparsely, because the correct behavior in a doubtful situation would be to check documentation, which is cumbersome during an interview.
Answer is terminologically correct	Terminology used in the answer is appropriate (see its importance <a href="#">above</a> ).
Candidate understands, where the question is coming from or what is a likely follow-up question, and elaborates <b>pre-emptively</b>	Candidate demonstrates understanding of the relationships between different aspects of software development or technology. After giving an answer about a given concept, he does not wait for the next question and proceeds to describe concepts linked to or based on it. Alternatively, when asked a question about a problem, he describes it, then describes existing solutions, their own problems and further solutions.
Can name corner cases/exceptions	Candidate mentions important exceptions to the rules.

Some candidates try to apply concepts they have learned in one technology to other technologies, which they deem similar.



Examples include Java and C#, JavaScript and PHP, etc. This is generally a sign of a candidate not paying due attention to the description of a position or underestimating required technologies (especially prevalent for JavaScript).

## Dealing with wrong answers

It is up to you, whether to inform the candidate about incorrect answers. Normally, it makes sense to do so only if you think that resulting discussion will actually help you test the extent of their knowledge (usually when the answer is only partially wrong, or the candidate mixed over the concepts and may yet give a correct answer). Otherwise, confirm candidate's answers with a neutral "Ok" or "Yes" and move on to the next question.

Sometimes candidate will ask directly, whether the answer was correct and what the correct one is. In this case it is best to tell the truth and give a short, terminologically correct answer from your reference. If candidate asks such a question more than 2-3 times, you should inform them that explanations will drain time of the interview, which will harm your final estimation of their abilities, but if the candidate still insists — it is best to give them the correct answer.

In regards to wrong answers, it also makes sense to skip advanced question for topics where candidate failed in the basics. For example, if a candidate does not know what are the dangers of concurrency in computing, it makes no sense to ask them about different types of locking.

You should never argue with a candidate about correctness of an answer, even when you are 100% sure. In this case, tell the candidate "Ok, I'll have to check documentation after the interview". If you are actually not sure about the correctness — take note of the answer and verify information given.

The fact that you should not confront the candidate, does not mean you should not pay attention to the issue. Pressing the wrong answer is a sign of either delusion or bad attitude, both of which should be noted in feedback if the behavior is pronounced or prevalent.

Normally you should look out for signs of candidate googling things during the interview or trying to cheat in a different way. These include audible keystrokes or sudden disappearance of background noise indicating microphone being set on mute to mask activity (but do leave a chance for the mic auto-muting itself or other benign explanation).

That being said, google is only bad if the candidate tries to cheat with it. If a candidate asks permission to consult reference material, does not refer to it too often and comes up with the right answer quickly — that is a sign of good researching skill. After all, candidate will have access to documentation during actual work and ability to find things quickly helps a lot.

How much googling is healthy is up to your discretion.

Remember, that the interview is limited in time. Giving a hint or additional question is okay, but spending 5 minutes on a topic trying to dig deeply is usually not. There are always more questions than there is time, spending 5 minutes on one question and then having to drop two others completely is not helping the candidate — a skipped question is almost as bad as a wrong answer for final estimation. This also means, you should gently interrupt the candidate if they keep talking about a concept past the point where it is useful for the purposes of the interview.

If the candidate asks you to skip a question due to not knowing/not remembering the answer, provided that question is not fundamental, it is also a good sign. Realizing the limits of one's knowledge is beneficial. Trying to guess or come up with a wrong, but plausibly sounding answer is bad.

From experience, 80% of correct answers at technical interview conducted over the phone is a good result for a software developer. This field yields itself better to written word and you should account for this.

## Ending the interview

At the end of the interview, you should avoid giving the candidate any preliminary estimations of the result and/or any promises of future prospects.

If the candidate insists, you should tell that you provide your feedback to the Recruitment team, which compiles feedback from all interviewers and does the comparison (and remember, this is the actual process, thus, even if candidate is the best you personally have seen for a given position, you don't know the level of candidates screened by other interviewers). In any case, you should advise candidate to keep contact with the Recruitment team for further information.

Sometimes a candidate directly asks for your estimation of his level of knowledge. In this case, you should avoid giving a direct answer, and instead point candidate to important gaps you've noticed (this will keep your answer meaningful). If candidate's level is low — you can advise a good fundamental book which you have personal confidence in (i.e. for C# that would be "CLR via C# 4th Edition" by Jeffrey Richter).

Otherwise, stick to advices like: "You can use more knowledge about Unit Testing, enterprise development always requires it". The person on the other side of the phone has invested time into this interview and it is okay to help them improve, so long as your advice is solicited and you are in a position to give it. Only when a candidate has a really good level of knowledge with no gaps, you can say: "**I personally** think your knowledge is good".

It goes without saying, but you should not allow any discrimination in the process of the interview. You must stick to estimation of technical skill, attitude and language level. No preference or disfavor is to be given to candidates based on their gender, nationality or citizenship, beliefs, etc.

# After the Interview

Remember: information about the candidate obtained during the interview is to be used only for the purposes of providing feedback to Recruitment team.

It is best to compose your feedback ASAP, while your memory of the candidates answer is fresh. Feedback form consists of your estimation of the candidate's general level (i.e. regular, senior etc.), opinion on their fitness for particular positions (yes/no/hold) and detailed descriptive feedback.

When should you decide on "hold" for a given position? Usually it means that candidate can potentially fill the position, but either just barely (a better candidate is likely to be found) or conditionally (additional knowledge obtainable before the second round of interviews is required by the candidate). Make sure to note the reason for hold in detailed feedback section.

Keep your detailed feedback concise and descriptive, include a sentence or two on the level of knowledge for each of the core technologies ([listed above](#)) that were tested and exotic techs specifically requested by position description.

Also, include estimation of level of English, especially technical English. If you think that candidate will have communication problems for some reasons — mention them (just remember, that sometimes it is a poor connection and not the candidate to blame for communication problems).

Most candidates have a good attitude, so you don't even have to mention it. A few, however, will demonstrate unprofessional conduct, which may include swearing, expressing negative **emotional** opinions about technologies without being able to provide technical justification (usually technologies they have no experience or knowledge of), refusing to cooperate and/or berating the interviewer, demonstrating discrimination. Such behavior is to be noted in the feedback. This includes any behavior that will likely negatively affect candidate's future team, other company employees, clients or image.

You may have a special opinion about a candidate, i.e.:

- they do not have enough knowledge to fill in a senior position, but have enough for a regular and show a lot of potential for growth
- they have very good knowledge in a field that is not key for current position, but would make them a valuable addition to different kind of project

- they have generally good knowledge, lack expertise in just a few key technologies required, but are likely to gain it in less than a month
- they demonstrate knowledge that is exceptional and are likely to help developers around them advance (possibly demonstrate mentoring experience or participation in development of tools and libraries used by others).

Make sure to note it. It is your goal as an interviewer to help Recruitment team with finding the best candidates for the company and matching them with projects where they would be most effective.

Good luck!