Interview introduction... #00:01:16-1#

I: Q1 #00:01:22-6#

B: (...) #00:02:56-9#

I: Sorry, who do the programming? #00:03:02-4#

B: Programming team is in India. #00:03:05-7#

I: Q2 #00:03:15-4#

B: the first I consider is how the logic written, like the part I always check is the common part of the code. Do we have code duplication or not. In case we have code duplication then I want the code to be refactored. Then I check the logic; how it's written, how many loops are there, how complex is the code. Third part: the comments; are the comments really readable. If tomorrow any some other person and the developer is not available. If other developer want to review the code or do some changes in the code, can he easily understand it. And fourth: the variable, which he used, no one use global variable in normal practice. #00:04:14-6#

I: Q3 #00:04:22-8#

B: Possible I check the logic behind the code, how positive is it written and can I think of a better way to implement it. That's my first parameter. (explain about the version/commit number...). Once it is done, I review the code: I go from function to function, from file to file. What part of logic is written, what file is the logic in, optimum enough, if is all the expectation of the code discovered. (give example...) #00:06:40-8#

I: Q4 #00:06:56-6#

B: First of all if it is a low quality code, we can get improved right away. Because we check it and get it improved. before it goes to production. then we bare something goes wrong. And then it will help us, reduce the big number of defects because what X also sees the committed code, that have been written, how many defects are we getting. It will reduce them. like if we were getting defect so it goes to sudden process/person and that costs a lot amount of time. #00:07:48-5#

I: anything else? #00:07:51-3#

B: No. #00:07:53-1#

I: Q5 #00:08:16-1#

B: We have one technical architect, which is a part of the Off-shore team. also we have different team from 6 to 7 people. #00:08:32-3#

I: and your team is how many people? #00:08:33-9#

B: we have 7 people in my team. #00:08:36-4#

I: yes.. #00:08:36-4#

B: We are 7, the 7 are developers. and one is the technical Architect. the role of him is review the code once it done, help the team find the optimal solutions. but to answer your question, I will keep it short. That first it goes to the technical arch. in off-shore team. Once he is done with the CR, then it will come to me, as onshore as SME (abkurzung für …), my role is SME. Then the code comes to me to review it, once I reviewed the code, I give comments and then we are good to go. #00:09:25-4#

I: Okay. so you work with who? #00:09:39-3#

B: they send me the code, i am not a developer, I am a SME. #00:09:56-7#

I: You send them the code changes? #00:10:01-9#

B: yes, we use JIRA tasks for it. And I commented their, like: " I don't like this" "please Change" "Please Change" and then they will do it. #00:10:17-6#

I: The 7 Developers in your team are all in India and the other in Germany? #00:10:21-9#

B: I am in Stuttgart right now, so there is 7 developers in India and there is also in India a Business analyst which also is a team lead. #00:10:43-4#

I: ya.. #00:10:44-8#

B: ya. #00:10:47-2#

I: and you review the code of these 7 developers. #00:11:01-7#

B: ya all of them. Suppose of have a change request. So we get the change request from other "Customer" and then for these change request I do the review. and then we get defects from the existing code from the "customer". Once they are done and then only the tech arch. will do the code review and not me. #00:11:35-4#

I: Q6 #00:11:42-7#

I: yes, there is always favourite developers, from whom which I duple check the code. #00:11:49-5#

I: can you tell me about them. #00:11:55-9#

I: yes. Because sometimes I found a responsible behaviour and then they didn't check it properly, they didn't understand the requirement correctly. and sometime the JUint are are not written properly. because some developers write really good Junit and some do not write so good one. That is why I want always to check with some developers. #00:12:35-1#

I: Q6 experience with them #00:12:42-2#

I: yes. the last week we had code review regarding a CR and then the Junit was not so good. It was like no matter what it will always pass. that is about Junit. I asked him to change it and he change it. #00:13:20-9#

I: okay… and I don't get it. Why do you like this one? #00:13:29-9#

B: aah I don't like this one. I am taking when I do not like. #00:13:31-2#

I: And the one I like? #00:13:36-8#

B: hmmm *he got confused* there it was, the was code refactoring problem there was a comment X which I found, other than that, everything was ok. It was not so good. #00:14:04-0#

I: Q7 #00:14:27-6#

B: do you have some points in mind? #00:14:48-0#

I: no whatever occurs to your mind, but not from the technical perspective. #00:14:48-0#

B: Can you give me an example? #00:15:23-5#

I: Explain... #00:15:26-4#

B: from the first code review. it was with good developers and my technical architect has already done the reviews and hmm I think the problem I faced when in my first code review was like I was not sure that the code, which he has written, will affects the other parts of the application. Like we remove something and then we are not sure, I was not sure. That the problem I faced. Other than that Junits were fine, code refactoring was ok, code quality was good. and there is nth I can complain about. #00:16:06-7#

I: and how did you solve the problem you faced? #00:16:17-0#

B: I checked the more experienced colleague and I asked him to help me. And then he helped me out and we found it. #00:16:29-4#

I: Q8 #00:16:50-0#

B: the benefit is the code quality which deliver to the customer is always good, not always but it is better than without the code review. The number of defects is less; we will always get defects but the number is always less. and the developer as he knows his code will be checked then he will also write a good quality code not anything he wants. #00:17:31-3#

I: and for you? how do you benefit from that, as an IT? #00:17:34-2#

B: for me, I do not see any benefit, for me as a person I do not see anything there. But as the project, it will improve the project quality. #00:17:52-2#

I: okay. Q9 #00:18:02-0#

B: yes. Once after all the code review we still get defects that it is not 100% secure even if you do code reviews, then are also defects and sometime the code fail. and sometime there are undesired behaviour which are not expected. #00:18:42-3#

I: Did you have a bad experience in code review, an unpleasant experience. (explain it ...) #00:19:39-2#

B: no I do not think so. It is part of my job and doesn't make me feel annoyed. #00:20:01-3#

I: do you like this process? #00:20:06-0#

B: I think it is a really good process. I don't know about the onshore team but for this model it is a very good practice #00:20:16-8#

I: ok. Q10 #00:20:47-2#

B: no that’s never happened. #00:20:54-0#

I: do you discuss the code changes? #00:21:06-7#

B: yes, the developer opens his code and we go through it and I said I do not think it is a right way. we can do it like this then we have discussion about it. #00:21:15-8#

I: is it always like that or only in special cases? #00:21:23-5#

B: always. #00:21:23-5#

I: You discuss it per skype #00:21:23-5#

B: yes #00:21:25-0#

I: and Q11. #00:21:58-9#

B: (explain something ...) I wrote the Junit can have a better JUnits, the current JUnits do always always pass. It was like writing if (1==1) so this case will always pass so please improve it. #00:22:48-9#

I: okay.. #00:22:53-4#

B: and also the code was not written properly, I could see lots of common code functions which can be avoided. So I recommend a refactoring of code. #00:23:16-7#

I: when do you comment the solutions? #00:23:23-5#

B: I think there are always some comment. Sometime there is nothing I can write it " do this" and like the solution. #00:23:32-3#

I: okay.. Q11 #00:23:53-2#

B: (explain testing process ...) #00:24:37-1#

I: and from the collaboration process and /explain... #00:24:54-1#

B: /explain the difference in functionality.... #00:25:28-0#

I: /explain #00:25:40-6#

B: ... #00:25:48-0#

I: Back to the question about the your team member, can you tell me 3 things about everyone of them? #00:26:06-2#

B: Every one of them three things! #00:26:01-8#

I: yes or three words #00:26:09-0#

B: 7 people! #00:26:13-2#

I: yes #00:26:14-2#

B: so starting with my team lead. She is good, responsible and take honorship. Technical Arch. is good but something, I think, he can improve. and what else... hmm and his "technical" is also good. *silence* and regarding, two developers are really good, the code quality that I get is also good, the changes are always of good quality. but when I talk about the other one they are like are freshes, first of all, so I can understand that they do not have much experience with the project also. So I can say the code quality is okay. sometimes the code refactoring, this part, is missing. especially the JUints are always not so good and sometime not written at all. so I think something which can be improved. #00:27:59-3#

I: and do u like to review the code from the first two? the good ones? #00:28:00-3#

B: sort of *laugh* I think it is part of my job. I know when the more experienced one give me a code to review so I understand it is ok and there is less chance of mistakes and I can do it easily but when the other two give it to me I can do it "poorly". So I ask question and then "why this" and "why not this" "why this". to understand that that ok what the "X" like. #00:28:28-7#

I: and the fresh one? are they improving? #00:29:13-2#

B: they do not repeat the mistakes #00:29:22-2#

I: and did it happened that a developer felt offended by the comments? #00:29:35-7#

B: in other team, I heard something like this. that the developer felt offended by the comments because he thought he have a lot of experience with that and with the work and (some other few words) and how dare someone give him comments on his code. But in my team it didn’t have such an experience #00:30:05-8#

I: how the comments were written #00:30:23-9#

B: it was more like a conv. and then it was like we have this and we do this... #00:30:39-0#

I: so how do you avoid that in your comments? #00:30:50-3#

B: it more like a diplomatic way how to improve things. pointing them to mistakes rather than saying how to improve it? #00:31:06-7#

I: okay.. Q8 #00:31:21-9#

B: there is a learning part from CR. I get to see different style of coding and maybe I will see different ideas. and a person if have a bug I can bug in only one domain and one media but with code review I can see many area, different part of the app. so there is always some learning involved. maybe when I code I will code I one pattern but when you see how other people code then you get different view. #00:31:53-3#

I: okay. Since when do u work with this team. #00:32:05-7#

B: 10months #00:32:11-6#

I: do u all start together? #00:32:16-4#

B: there is more experienced, two or three, they were already here when I joined this project. and there is no one who came after me. #00:32:39-9#

I: Q7 #00:32:59-5#

B: yes definitely. I think I have more knowledge now and more better confident when writing the comments before I was new to the projects. I now know the code better and the project better so only then colleagues start respecting u respecting ur opinion. Because when u r fresher and new to the project they think if he is new to the project how can he gives the comments. and once they know ya he also knows something and we could use his comments constructively. Then I think we are already "X". #00:33:57-7#

I: okay so thank you ... #00:34:05-7#