Do you believe that the taxonomy corresponds to the challenges faced by developers when reviewing refactoring?

Expert: Yes

How often do you encounter these challenges in your company as an author or reviewer?

Integration: 3 / 4

Expert: We definitely see configuration issues, merge conflict, build failure, API management is not popular). So it is 3 out 4 I see them personally I guess.

Objective: 3/6

Expert: I mean you cannot submit code reviews without an unclear goal or unknown benefit., it will get rejected. You should know what is the benefit and your goal. You cannot create PR if you don't have it approved before. In an open source project, you just contribute to the project, so there is no restriction of what you should contribute. But closed source projects, no, there is restriction of what you should contribute, so if you refactor, it should be approved by the program manager before you are able to provide code for it. So, there is an unclear goal in the company. Again, it is way different when you talk about open source projects and closed source projects. When you have open source project, definitely you will have unclear goal and benefit because there are just people, there are not restricted of they should do, they can do whatever they want.

Expert: Why did you add bugfix as part of refactoring? I mean based on what I worked on. I mean when you work on fixing bugs, you should do a minimum change. Just change to fix the bug. You cannot just do any refactoring. Again, I am talking about industry, it is way different than open source. So, I am telling you what I know based on what I have. So, bug Fixing, when you fix a bug, you can only fix bug, you cannot even change a variable name unless you have to change it. This is very highly unlikely. Bug fix you fix issues where something does not perform correctly, so you just need to keep. I mean you can change the code, you can do minimum change to have a minimum possibility of introducing a regression. That is how we fix a bug. Is there applied everywhere? No, when I was in my previous company, there are some people they fixed bugs and they changed 5 files. But, it is different, every place has a different guideline. So, for objective, 3 of them I definitely see them. I see potential side effects, scope change, feature support. I definitely see them. For bug fixing, I think this forbid you cannot do it because you need to do a minimum change. And for the objective, unclear goal or unknown benefit, you just cannot do it. You cannot just submit a PR based on just what you think. Again, this is closed source project. In open source project, they can do it. For me, I saw only 3, and I told you why the other 3 I don't see them.

Testing: 3/3

Expert: For testing, I guess 3 of them I see them.

Quality: 5/7

Expert: I don’t know. I just forgot the name of technical debt. I forget what it means.

Author: So, technical debt means when you do something but it is not perfect, it means you do like a patch without necessarily paying attention to all the issues. Like you choose the algorithm. Like when you write a temporary code. You try something, but you are not sure whether you want to keep it or not or you want to keep it as is for now. The definition of technical debt means anything you do but it is not perfect and you know it is not perfect. Then you are in debt to the client to write a perfect code, because this is what the client expects, but you because you are in a rush or you are not certain or because you have other constraints. You write a code that is temporary, and you let it go. It is like when you write something and you add temp or TODO or fixme later.

Expert: So just again, my answer to the question is based on the guideline that I have currently. Is this something you want, you want something that I believe it. For me, I believe it is fine to put TODO, but do it later on. But in my PR, I just submitted 3 days ago. One of the line I put in TODO, they told me, don’t put TODO. Leave the task alone without TODO. I mean there is a difference between what I want and what people want. I finished all that they asked, but there is one line that I wrote TODO I am going to do a separate task for this. They told me not to go and do the other task, and then come and finish this one.

Author, so, this is the idea. We saw discussion like that. I will do that later, like todo, and I will fix this. And they depend on the review. In your company, they say no, maybe in other places like open source, they cannot force them to say no for example.

Expert, that is what I am saying, it is just open source I think.
I did not understand what internal quality attributes were. I think you can do an interview with a guy from open source so you know how they think.

Author: I will show you an example. Here is it: “think there is still an inheritance problem with the Base2 class (see inline) and the way python handles multiple inheritance
means we have to be careful when creating the UTF8 versions of the unittest classes.”

Expert: Okay, I understand.

Author: it is a design level problem.

Expert: yeah, I mean I saw code smell, I saw internal, I saw external. Technical debt, I just did not. Design pattern they did. Coding convention, what do you mean by coding convention? lack of documentation definitely did.
Coding convention, what do you mean by that?

Author: Let me show you an example. Here we go, “We've seen how difficult it is to arrive at a single naming convention} that works for all OpenStack projects, but as Rally expands beyond benchmarking OpenStack alone, it is quite simply impossible to create a single name format that will work for literally everything.”

Expert: Yes definitely, you should have it, so yes there is a naming convention, so you should follow them. So, it is everywhere I guess.

Author: So, when someone does not follow that naming convention, we saw reviewers say rename this because it is not following the convention.

Expert: yes we do this during code review.
So, 1,2 , 3, 4, 5, so 5 out of 7.
The things that I did not see is design pattern and technical debt. I mean why design pattern you see the design pattern in the refactoring. You mean someone who implements a design pattern.

Author: Yes, they are saying for example, you are doing a decorator or factory method design pattern, and they are saying can you please modify it because it is not following the appropriate way.

Author: It does not follow the appropriate way of design pattern. You are doing a design pattern, but you are not doing the design pattern correctly.

Expert: It's very possible but I did not see it.

Author: You are saying 5 out of 7, right?

Expert: 5 out of 7. Again, I am telling you what I have seen and I did not see. It is different from what is correct and what is not correct. Design pattern is definitely correct, but did I see it? Did I see people using it? No. The code I see is way more complex than the design pattern.

Refactoring: 3 / 4

Expert: What does domain constraint mean?

Author: It is related to OpenStack that is based on cloud infrastructure service. Here is an example: if we want to refactor barbicanclient in this way, we should not create a client instance successfully if a non-existed version is
Specifie. “

Expert: it is not clear to me.

Author: I don’t have the link to the review to show you. I don’t have another example to show now.

Author: I think the author is trying to say that they are refactoring based on specific technology.

Author: Yes, but It is also linked to the cloud. Maybe the name is not descriptive.

Author: it is good to know that, because if it is not clear to the expert, it won't be clear to the reader. We will take that into account, expert, thank you. Let's keep going.

Expert: For this: 3 out of 4.

Management: 4/4

Expert: For the management, I say no ongoing review, forgotten review, change dependency, review guideline violation. This is very good. I see all of them. No review activity is very common. And this is happening for a reason. They get rejected, and it stays open. People don't close it, so some people don't close it. It is everywhere. Forgotten review is very often. People forget them. Change dependency usually people from different teams, you just add them in the code review, and they will. I mean based on what I am doing, when I put change a file from this team, it automatically will add people from that team to a code review, and it becomes a required reviewer. You know. Again, I am talking about big stuff compared to open source. But based on what I do, If I change a code, a file that you own it in your tem, it will automatically add you in pull request when I worked on pull request. I will not add you, the system will add you. I am not going to wait for someone else. I don't know if you want to rename it or stay it as it.

Author: you mean rename "change dependency", right?

Expert: yes, just make it more clear. Here you mean change depends on someone else or other team. Sometimes, the other team becomes part of the PR, the pull request you create. So, it is there, you can say 4 out of 4.

When considering its future implications, how important do you think these challenges are?

Expert: I think it is very important in general. All categories are important.

Are there any actions taken to prevent these challenges in your company?

Expert: Yes

If yes, can you please mention these actions?

Expert: All of these has bots. There is a bot to detect if there is a merge conflict, there is a bot to detect build failure, there is a bot for testing. For objective, I don't think there is a bot. Quality stuff, I don't think there is anything there. Refactoring, no bot for that. No review activity there is bot for that.

Expert: So the integration, testing, management, all of these have bots to detect these.

Author: So, let me the expert understands this. So, these bits you run them yourself, or you run them automatically.

Expert: they are running automatically.

Author: And this is only for the company you are worked on right now, not before?

Expert: yes, before I don't think we have it.

Author: so, it is not generalizable.

Author: I think bots are a good idea, some of them are automatically detected, some of them are not.

Do you have any recommendation in refining the above taxonomy?

Expert: Yes

If yes, can you briefly mention these recommendations?

Author: I think the expert did already tell us what things make sense to him and what things do not make sense to him. 3 subcategories does not make sense to him.

Author: And there are two categories that need to be renamed as well.

Expert: Again, it does not make sense to me because these are open source. It does not mean it does not make sense if you go to someone from other company.

Author: Yes, this is your survey. It is about you. So, it is based on your experience. That is what is valuable to us. Okay, 3 of them do not make sense because we will report this, 3 of them do not make sense to the expert.

Are there any other challenges that you have faced during reviewing refactoring changes? if yes, can you briefly describe them?

Author: Can you remember a case in which you review refactoring or you submit a change for review, and someone mentions something that we did not mention? Or we did not capture in this category?

Expert: Let me read it back. What about if you do refactoring, most of the time, you want to make the code easy to maintain? Where is that?

Author: Yes, this is part of the external. External quality attribute means maintainability, easy to read, easy to maintain, and easy to understand, these stuff.

Expert: Yes, so I think there is nothing in my mind.

Author: Good, it means there are no challenges that we are missing.

How many years do you have in the software industry?

Expert: 8 years

For how many years have you been familiar with refactoring?

Expert: 8 years

How many years of code review experience do you have?

Expert: 6 years