Interviews Analysis and Coding Examples

Hamdy Michael Ayas, Philipp Leitner, Regina Hebig

Snapshots of coded interviews

….

Researcher: In your personal opinion, what do you consider as microservices based architectures? So, what defines it? What are you trying to get out of it?
Interviewee: From what I read initially, the main idea, the base of that is in my opinion the decoupling of the modules that exist in a product. And afterwards you can talk about different projects in different environments, different databases and that stuff. I believe that depends on the type of the product on what is the amount of time you have, how big is the product, what did you have, I think it depends on many factors #as-is_dependent_degree_of_MS_design. So depending on what you are doing exactly, how that microservices type thing will be.
Researcher: Can you give me some examples of microservices types as you describe them?
Interviewee: Yeah, for example, at the company that I am now the discussion that we had ah, about how we proceed with our system. Initially, we were talking about the practices that are described, for example, in every blogpost that you'll read about microservices #debating_MS_principles_practices. So, it's a project which is independent from the rest. Let's say you have a system which has different modules. Ideally, every module would be a different project, different deployment, different databases etc. #different_projects_per_MS. But that I think it depends on the size of the modules. If you have models of that very small, it can be one project. If you have a module that is very big, it can be two or three microservices, and we were initially discussing that. But afterwards we saw that because of time, because we had a deadline, because of the additional effort #time-effort_dependent_decomposition we decided to do one API and one microservice #single_MS_decomposition. But in that project, we had separated and decoupled our model. So mainly because of the effort. Because when you're saying that you will do each module a different project #different_projects_per_MS, that means that we will have a different deployment #different_deployment. Then that means that someone should prepare a different github let's say, for every module and you have a lot of modules. Someone should prepare a database for each module. Uh, okay, you do that. You spend that effort and that time to do that #big_effort_upfront. Afterwards, are you sure that you should have one database for each module? #debating_MS_principles_practices But then you say I have a database that every module should see that. So, then you have decided to have a common database for all of them. But you also have different databases for each module #different_DB_along_shared_DB. You have these considerations and these discussions. In general, because of the extra effort of all these and the complications of all these, we decided to go with a single project, but still keep the decoupling of the modules #cost_benefit_analysis. We said that maybe in the future, in the way that we did that we can easily add more #design_for_short_and_long_term, if we decide to make every module a different API, it will be easier from the way that we build it now

#design_for_update_and_extend. Because in the same project, everything is a different package let's say, a different DLL, a different built #different_projects_per_MS. So, it is easy to have them separated.

…..

Researcher: So, can you describe the journey of migrating?
Interviewee: Yeah, basically, we started it with, I guess the wrong way. We tried to take the monolithic and break the code actually into many smaller components #started_with_breaking_code. But then we said, let's drop that one. It is easier to start from scratch #MSA_from_scratch and basically just migrate the business logic, I mean the design and not the actual code #migrate_design_not_code. So, basically what we did is, uh, we implemented the code again on a new framework that is basically optimized for this kind of architectures for microservices #reimplimentation_with_specialized_technologies. And we did some design changes as well. So, more or less it wasn't really a migration, it was a re-write more or less.
Researcher: Maybe a re-factoring or restructuring then maybe?
Interviewee: Yeah, em no, it was a re-implementation. We implement again the code; we didn't copy pasted any code #scrapping_old_code. (we used) Just the design, just the business logic #design_reuse.

…

Researcher: Do you think of any other tool that you think could help you with this? How would it be useful for you to use such knowledge?
Interviewee: When you say tool you mean a technical tool?
Researcher: It could be both technical or something that you would use in the organization to convey a message, for example, or to align people. Any of any of those.
Interviewee: I think the tools and frameworks and guidelines they are all in place. I mean, you can find them everywhere on the internet. You just have to use them and apply them #tools_availability. The difficulty is to apply them #challenge_applying_tools. And as I mentioned before, make all the people of the company change their mindset. And start thinking in a more let's say Agile and microservices way #challenging_mindset_change. Because another part of the microservices is also that you are developing a methodology let's say. You need to be scrum or kanban. You need to follow these principles #intertwined_with_methodology. But to follow these, to follow scrum, you need to change your mindset. So basically, I think I guess the tools are in place more or less, they are everywhere on the internet. You just need to follow them and change your culture and your mindset to comply with these #cultural_change.

…

Researcher: What defines microservices for you and what are you trying to get out of it?
Interviewee: Okay. What we try to get out from it it's ah, just to become like easier for us to maintain our product #product_maintainability or easier to set up and configure. Because whenever you have, like, a new system it is taking from you like more time to set up and configure for each client #faster_delivery_config. And then maybe you'll find yourselves that

okay, you do an installation for some table, which has never been used by this specific client. So the reason is why I need to give them some code that they are never going to use it #delivering_garbage_code as that code every time, whenever I'm going to update it from my core system, I need to do like a QA and wait for them just to be sure that it will not be impacting anything on them #effort_QA_on_garbage_code. So, by using microservices, we are going to be sure that we're going to save some time and effort from testing and on the same for debugging and finding issues whenever we do, like any upgrade #scoping_efforts_with_MS.

Researcher: Yeah. So, what is microservice for you?

Interviewee: Yes, for us microservices now it's like will be more easy for us to maintain and setup, and gain more clients. For example, say now was one huge system. I cannot say, OK, I'm going to deliver it for you in one or two days. But when I do like microservices, if you are like small organization, which you need to use, like my system, I can do it for you in one or two days #shortened_time_to_client_delivery. Depends on what your requirement. So, I can deliver it for you, only the part that you are interested about. I don't need to give you a huge system and I don't need to spend time from my side and from client side to do that installation and delivery and then testing and training for them, for all that system #value_engineering_with_features. Because whenever you talk with a client about huge system, they are going to think long time about it, if they need it or no and it will be impacting a lot of stuff from their end as well and from their third-party system. But whenever you give them like microservices, you are sure that this is the only part that you need to give to them #lower_leap_to_sell. So, I'm not going to spend more time to deliver it for them, I will not spend more time to configure it for them. Then that means that I will gain more clients in the market #shortened_selling_cycle. It will be less costs for the client. But at the same time, I have to achieve that I can get more client at the same time #enabling_economy_of_scale.

…