# Follow up interview

## The need for it

There were some interesting answers from the survey, and there’s need to confirm that people understood the questions as they were intended. The questions may have to be rephrased for the next survey.

State: Today’s date is X. I am currently at Qlik’s offices in Lund, conducting this interview. Would you please state your name?

You took this survey about two weeks ago. Since the survey was anonymous, I have asked NAME to answer some of the questions again before this interview.

1. In the survey, the term “Tools and frameworks” was used. When you read, what sort of software did you think of?
2. In part two, we defined what we meant by “Tools and frameworks”. Did you read this definition?
3. If your read through the questions, were there any of them that caused you confusion, or where you were unable to answer truthfully?
4. Are you in a position to make software usage decisions for a group of people?
   1. If yes: Did you put yourself in context of making decisions for a group, or for yourself?
5. When you answered the questions, did you put yourself in the context of working on a hobby project or working professionally for your company?
6. For the question “Which of these traits or aspects do you usually consider when deciding if you want to TRY a new tool or framework?”, was the emphasis on TRY important? Would your answers have been different if the question was phrased as “Which of these traits or aspects do you usually consider when deciding to use a new tool or framework?”?

Go through questions and ask them to explain why they answered the way they did:

1. How quickly do you usually decide if the tool or framework is for you?
   1. Is the question too broad to answer truthfully? Is every case unique, or is there an average time for you?
2. Ask about contradicting answers:

Contrdicting answers:

1. I can have working code quickly – It takes a long time to get started
2. API
   1. The API has code examples – The API has poor or no code examples
   2. The API gives thorough explanations – The API is poorly explained
3. The documentation doesn’t assume any prior expertise – The documentation assumes prior expertise of the software
4. The documentation uses inconsistent language – The documentation has consistent language
5. The documentation is easy to navigate – The website for the documentation is hard to navigate
6. The release- and change notes are thorough - The release notes are poorly written
7. Community
   1. There exists an active online community around the software – The online community is dead or has little activity
   2. The online community around the software is unappealing
8. The creators behind the software are not transparent – The creator of the software has high transparency with its issues, ways of working, future plans, etc