Script for Interviews of API Designers

Version 6- December 2017

Contact: Brad A. Myers, Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue. Pittsburgh, PA 15213, bam@cs.cmu.edu, 412-268-5150

Is it OK if I record this interview?

Introduction

First, I need to read our standard introduction, as required by our protocol.

Our group at Carnegie Mellon University (CMU) has been doing research for many years on how to improve the quality and usability of Application Programming Interfaces (APIs) by using human-centered approaches. For example, we have done usability studies of finished APIs and API documentation to identify usability problems. By "API" we mean to include libraries, frameworks, toolkits, and software development kits (SDKs) created for use by other people, both for internal and external use. "Usability," as we are using it here, includes a variety of properties, not just learnability for developers unfamiliar with an API but also efficiency and correctness when used by experts. Usability also includes providing the appropriate functionality and understandable ways to access it. This property is sometimes called "DevX" or "DX", for developer experience, as an analogy with "UX," for user experience.

We recently received funding from Google and the USA's National Science Foundation (NSF) for a research project to try to better understand how API *designers* do design today, including what process is used for API design, what barriers and difficulties there are for this design, and what designers would like to know from the potential and actual users of their APIs.

We understand that you have done API design, and would like to interview you, which is expected to take about 90 minutes to 2 hours. This study is being performed by my group at CMU, and your participation is entirely voluntary, so whether or not you participate will have no bearing on your job. You may withdraw at any time. You must be 18 or older to participate in this study, and there will not be any compensation for participating. The interview will be conducted in person or by phone or through the computer (e.g., Hangouts or Skype). Since the interviews will be recorded, it is important that you be in a private room, and not in an open-space cubicle, for example. These recordings will be protected on computers at CMU. There are no expected risks or benefits to you for participating, beyond the benefits of helping improve the understanding of API design in general.

This study was approved by the Institutional Review Board (IRB) at CMU and the appropriate people at your company. Any public publications that come out of this research will be entirely anonymous and general, and no specifics about your answers or your APIs will be included in any external publications without your express written permission. To the extent that we want to report on differences in process among the companies, no company-specific identifying information will be released without specific written permission of the company. However, your exact answers may be shared with a designated person or people at your company, who will be identified to you in advance, so you will not necessarily be anonymous internally (but they will not be shared with any *other* company). However, if you want to keep something private even from people at your company, then please let me know if any of your particular answers should be also hidden.

Is that all OK?

Is it OK if I record this interview?

Background Questions

1. What are your undergraduate and any graduate degrees?
2. How long have you been doing software development?
3. What are the main programming languages that you are using recently?
4. What is your current job title?
5. How long have you been *designing* APIs?

6. How many different APIs have you helped develop?
   a. In this answer, how are you defining the boundary between APIs for your count? Is it clear what constitutes "one" API, or differentiates one API from another API?
7. Can you tell me the names of some or all of them?
8. Can you describe what kinds or types of APIs they are?
   a. For example, are they public REST APIs, web APIs for internal use, Java library APIs, etc.?
   b. Are they public, private or for trusted partners?
9. How large is the largest API that you have worked on? (For example, how many methods in the entire API, not just the part you worked on?)
10. How would you rate yourself in terms of your expertise in API design?
    a. Like from "novice API designer" to "expert API designer"

General Questions about API Design

11. From where did you learn to do API design?
    a. Did you have any classes in school that covered relevant topics?
    b. Are there other material or classes that you have used to learn about API design?
    c. Have any of the classes or materials mentioned usability of an API?

12. There are various books and blogs that discuss principles of API design. Are you familiar with any of these?
    a. Do you find them useful? If so, which ones?
    b. Do you disagree with any of the information in these? If so, can you give an example?
    c. What reading recommendations would you have for a new API designer?
13. Can you identify any characteristics or skills that differentiate a person who is a good API designer from a software developer?
    a. (If they don't understand the question) That is, if you were interviewing for a junior API Designer position, what would you look for in a candidate that might be different from what you would look for in a "regular" software engineer?
14. Are there some APIs that you think are particularly well designed?
    a. What about them do you admire?
15. If you were to write down your own "lessons learned" or "rules of thumb" or "best practices" about API design, what would it include?
    a. How did you learn these?
16. Can you relate some *mistakes* you made in previous APIs that you now know to avoid?
17. What are some critical things that you as an API designer need or want to know to support you in your API design and development?
18. Are there any *metrics* that you have found most useful for evaluating your APIs?
    a. Does this differ based on the stage of the API?
    b. What about different metrics for *usability* vs. *usage* vs. "*API health*"?
19. What do you consider to be the key decisions that you have to make when designing and modifying APIs
    a. What data do you need to help you make those decisions?
20. Does your company, division or group have **written API guidelines** that new APIs are supposed to follow?
    a. Can we see them?
    b. Who wrote them?
    c. How are they enforced or checked?
    d. Do you find them sufficient to cover most situations and design questions?
    e. Are there any unofficial guidelines that developers commonly use in your company?
    f. (If there are any unofficial guidelines/personal manifestos) How do new programmers learn of them/is there a repository where all employees can access them?
21. Do you have any **tools** that help check or enforce API quality?
    a. What kinds of things can the tools check or enforce?
    b. What cannot be checked or enforced by the tools?
22. What could future tools do to help with API design or evaluation?
23. Is there a standard **process** used to design and evaluate APIs at your company?
    a. Are there different processes for different parts of your company or different kinds of APIs?
        i. Who gets to decide what process to use?
    b. Is it formally proscribed for API designers, or just informally followed by various teams?

    c. Which roles or people at your company are involved in the design of the API?

    d. Are there multiple levels of authority? For example, who gets to decide requirements for the API? The architecture of the API? The specific method names?

    e. Do you create a design document or specification?

    f. Is there a separate review and approval process before APIs are released?

        i. If so, at what point in the design?

        ii. Who is involved in the review?

    g. What sort of testing does the API go through during / after development (beyond reviews)?

        i. Is there any usability testing?

24. There is currently a lot of externally available data relating to APIs, such as star ratings on documentation, GitHub issues, Stack Overflow posts, etc. Do you look at any of these as appropriate proxies for an API's usability or quality?

    a. If there are docs star ratings, is that just as good a proxy as Customer Satisfaction (CSAT) surveys?

    b. Have you ever relied on this kind of data? If so, for what?

25. Do your API users have a communication channel back to the API developers?

    a. Do they suggest features, etc.?

    b. Has this been useful?

26. Are there any *other* kinds of data that you collect about your APIs that hasn't been mentioned yet?

    a. How do you use these data?

27. After APIs have been released, how do you decide whether it is acceptable to make non-backward compatible changes that will break API users' code?

    a. Do you find it to be straightforward to tell if a change is breaking or not?

28. Do you know how the documentation for your APIs is written?

    a. Are you personally involved in writing or reviewing the documentation for your APIs?

        i. Are there technical writers involved?

    b. What kind of documentation is available for your API?

        i. Specifications? Tutorials? Example code? FAQs?

    c. How do you decide what to include?

    d. What do you think are the greatest challenges users face when using your API's documentation?

    e. Have you or anyone else ever evaluated how discoverable, readable or easy to use your API documentation is? If so, how?

29. Have you had any experience with APIs that are **automatically generated** from a high-level specification? What were they?

    a. What about automatically generated **libraries or SDKs** for the APIs?

        i. Are there usability problems with automatically generated APIs, libraries or SDKs?

    b. Some studies have shown usability problems with these. Do you have an opinion about what aspects of the design are most relevant to the usability problems of automatically generated APIs?

    c. How can automatically generated libraries be improved?

30. In general, how much do you think API design affects product adoption?
    a. Why?
    b. Do you know of any examples where your company lost out to a competitor due to the customer preferring the competitor's APIs to yours?

31. What do you feel has the biggest impact on the usability of APIs?
    a. (if they don't understand the question, maybe ask specifically about: Architecture? Client Library (if a web API)? Documentation?)

32. Have you seen any examples of the *misuse* of an API that have resulted from users misunderstanding the API or the API designers misunderstanding the users' needs?
    a. What would have been a way these kinds of situations could be avoided?

33. Can you tell me about some compromises in the design of an API? What were the tradeoffs that had to be made where usability was given lower priority than some other requirement (e.g., performance, release time or backwards compatibility)?
    a. If so what was the tradeoff and how was this decided?

34. Now we are getting near the end, and we have a few more general questions.
    What's the biggest thing that you struggle with for API design that we haven't talked about yet?
    a. What do you really wish there was an authoritative answer for about API Design in general?
    b. For example, if there was an article published that answered one question that you've had about API design, what question would it be answering?
    c. Have you figured out anything that you think other companies could make use of?
        i. From your company or you personally?
    d. Our group does usability studies on various design patterns of API design. Can you think of any studies we could do that would be helpful for future designs?

35. Do you feel like API Usability is recognized in your company as important?
    a. Are there resources dedicated to API usability? How much
        i. Is this enough?
    b. What should your company do to put more focus on API usability?
    c. Do you feel like API Usability is recognized as important generally in the community?
        i. If not, can think of anything to do about this?

36. Any other thoughts on API Usability that we haven't covered?

Thank you very much! These have been very useful!