

☐ Recording Device      ☐ Email List of Tools      ☐ Extra Paper      ☐ Eight Copies of Page 7

Start Time \_\_\_\_\_

I would like to record this conversation so that I can refer back to it later. May I? ☐ Start Recording

Do you consent to participate in this interview? ☐

### ***Tool Adoption Interview***

This interview will have four parts. In the first part, I'll ask you a questionnaire about your programming experience. In the second part, I'll ask you some open ended questions about how you learn about new programming tools. In the third part, I'll ask you some more direction questions about learning. And in the last part I'll ask about some of your opinions. On to the first part.

The following questionnaire is intended for me to get an idea of what your programming background is. Your answers in no way affect the rest of the interview, it simply gives me a context for interpreting the results.

What's your job title?	
About how many years have you been programming professionally?	
What's your age?	
Over the last year, about how many hours per week would you say you spend programming, on average?	

In the past year, which editors or programming environments do you use regularly? Could you estimate about what relative percentage of time you use each?		How about languages you've used in the past year?	
IDE/Editor	Percentage	Language	Percentage

Can you describe the environment that you usually work in, such as how many other programmers you work with, how often you interact with them, etc?

---

---

---

---

---

---

---

---

---

---

In this interview, I'm interested in your experience regarding learning about new tools. By "tool," I mean something that helps you perform a task, such as an Outline View, auto-complete, Tasks View, Quick-Fixes, Google Code Search, and so on. I've sent you a visual list to help jog your memory. Go ahead and take a look at it, and feel free to refer to the list at any time. When you're satisfied, please let me know.

Now, the second part of the interview. Can you pick 3 tools off of that list, and then tell me how you discovered and learned each?


So there are a bunch of ways that you can learn about new tools. Based on your experience, I'd like you to estimate how effective each method of learning is on your likeliness to use a tool again. I'm going to describe 7 learning methods; if you can think of more, please let me know. So, please pick two methods that, in your experience, have been most effective.

Best	Worst	Peer Observation, where you observe someone else use a tool while programming that you didn't know about
		Peer Recommendation, where someone observes you programming and suggests the new tool
Best	Worst	Tool Encounter, where you just happen to find the tool by exploring the user interface of your development environment
Best	Worst	Tutorial, where you are reading or watching a tutorial that mentions a new tool
Best	Worst	Written Description, where you notice that a tool is mentioned on a website or publication
Best	Worst	Twitter or RSS Feed, where you learn about tool from someone or some site that you are following
Best	Worst	Discussion Thread, where you learn about a new tool after reading it on list of comments, forum, or email discussion

Now, how about two that, in your experience, is least effective?

Now, the third part of the interview. I'd like you to think about a situation when you were observing someone programming, and you noticed them use a tool, and you learned about the tool based on what they did.

Can you think of any such occasions?

Let's call the person who used the tool that you didn't know about previously "the teacher."

Now let's flip it around; try to think about a situation when you were programming, and you learned about a new tool because someone suggested that you use the tool.

Can you think of any such occasions?

Again, we'll call the person who made the suggestion "the teacher."

About how often do you think that you learn about a tool this way?	daily, weekly, monthly, quarterly, yearly
About how many tools do you estimate that you've heard about this way?	
About how often do you think that you teach a tool this way?	daily, weekly, monthly, quarterly, yearly
Is it your experience that learning about a tool this way is efficient?	
Can you think of an instance where tool transfer should have happened, but didn't?	
Are there any blogs that you read regularly?	
Do you use twitter? For what purpose?	

## *Tool Creation*

Now the last part of the interview. Part of reason that I'm doing this interview is that I'm looking at ways to automate the process of learning about new tools. Specifically, I'm working on tools that tell programmers when they didn't use a tool but could have.

Do you think that such a tool would be useful?

Do you think that it could be as effective as other kinds of tool learning? Why or why not?

That's it! Thank you so much for your participation. Talking with and observing real developers is the most important research method we have, so I really appreciate your help. Before we hang up, do you have any questions for me?

Thanks again!

End Time \_\_\_\_\_

Jot down any tool implications.

Initiated By    Learner   ☐                      Teacher   ☐

Subject Was    Learner   ☐                      Teacher   ☐

What was the <b>tool</b> ?	
About how <b>long</b> ago?	
What was the human <b>relationship</b> ?	supervisor, subordinate, peer
What was the <b>programming task</b> ?	
How <b>receptive</b> was the learner/teacher?	
When was the <b>next time</b> that the tool was practiced by the learner?	immediately, next time I programmed
Did the learner <b>ask</b> how the tool is used?	
If so, when?	immediately, long after
How?	
Did the teacher <b>tell/show</b> how the tool is used?	
If so, when?	immediately, long after
How?	
How often is the tool <b>used now</b> ?	

Which **other** tools have you found out about in this way?