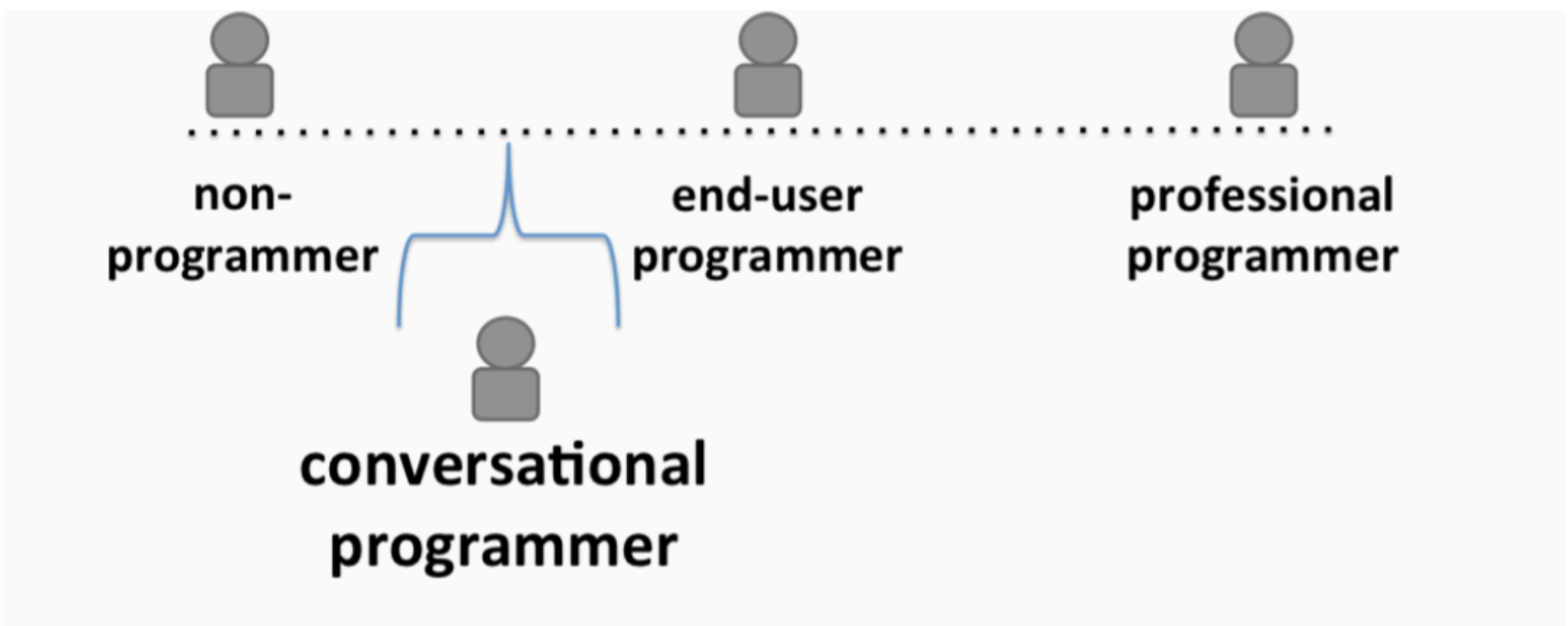
Investigating Learning Strategies of Conversational Programmers

April Y. Wang, Parmit K. Chilana @ Computing Science, Simon Fraser University

Introduction

The overall goal of our research is to shed light on the learning needs and strategies of conversational programmers and how they differ from artifact-creation needs of end-user programmers and professional developers.

Conversational programmers are people who do not have formal training in computer science and are working in non-engineering roles but try to learn programming to engage more effectively in technical conversations or to improve their job marketability [1]. Their goal is not necessarily to create artifacts or solve computational problems, but to use programming literacy as a mechanism for establishing common ground in technical conversations.



Research Questions

1. How do conversational programmers perceive the effectiveness of different programming learning resources?
2. To what extent does learning programming help conversational programmers' on-the-job?
3. What are conversational programmers' attitudes towards learning programming?

Methodology

Semi-Structured Interviews

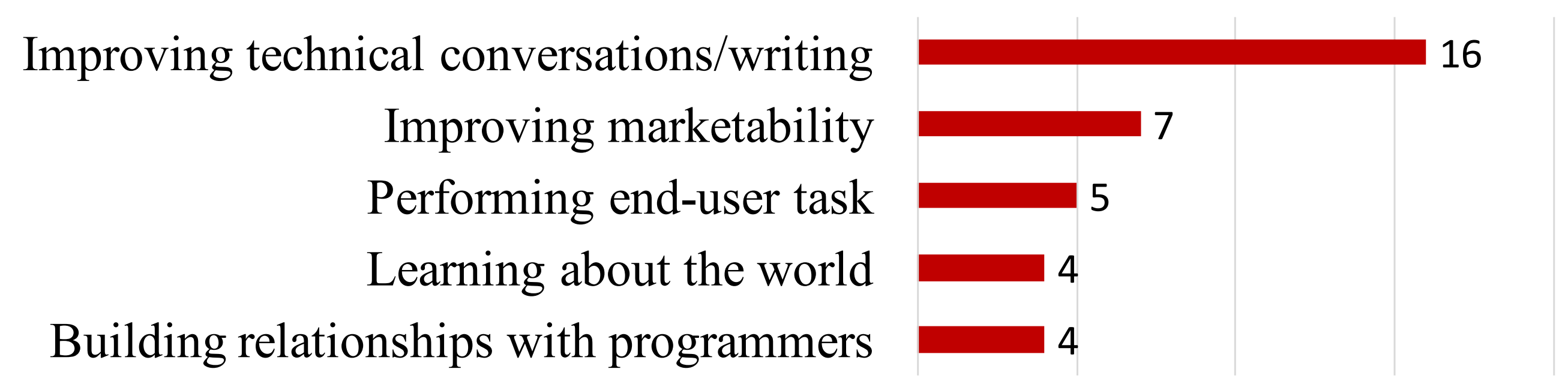
To better understand the needs of conversational programmers, we have conducted semi-structured interviews with 23 conversational programmers who have recently made attempts to learn programming on their own and who come from a variety of non-CS backgrounds and job roles (e.g., entrepreneur, HR coordinator, product manager, event manager, marketing assistant, designer, archivist).

ID	Occupation	ID	Occupation
P1(31-40F)	entrepreneur	P2(19-30M)	visual designer
P3(41-50F)	bank clerk	P4(41-50F)	HR coordinator
P5(19-30M)	intern as helpdesk support	P6(51-60F)	retired pharmacist
P7(19-30M)	business professionals	P8(19-30M)	event and marketing coordinator
P9(19-30F)	digital marketer	P10(31-40F)	researcher in Health Sciences
P11(19-30F)	library archivist	P12(19-30M)	intern as business assistant
P13(19-30M)	program coordinator	P14(19-30F)	HR coordinator
P15(19-30F)	program coordinator	P16(19-30M)	intern as marketing assistant
P17(41-50M)	product manager	P18(31-40F)	researcher in Humanities
P19(19-30F)	researcher in Arts	P20(31-40F)	digital communication manager
P21(19-30M)	student in business	P22(51-60F)	medical instructor
P23(31-40F)	researcher in Psychology		

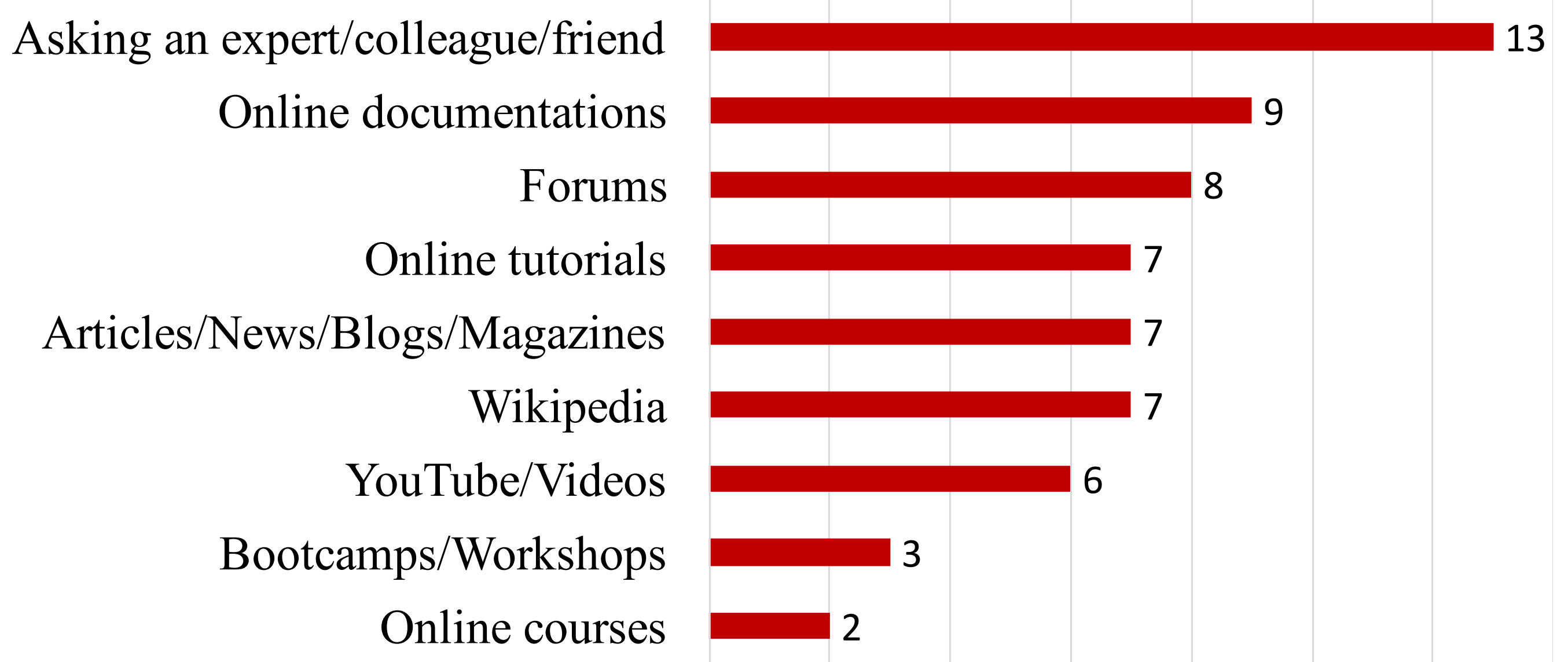
Initial Results

Using a grounded theory analysis approach, we are synthesizing key findings from the interviews to better understand the motivations of these diverse conversational programmers, the challenges they face in technical conversations, how learning programming helps (or does not help) their work, and their perceptions of advantages and disadvantages of using different learning resources (e.g., online tutorials, courses, videos, forums) vs. relying on colleagues or experts.

1. Why do conversational programmers want to learn programming?



2. What resources do conversational programmers use?



3. What factors would conversational programmers look for when evaluating a resource?

Type of Content	What content does the resource provide? (Conceptual; Syntax; Technical trends)
Efficiency	How much time has to be committed when using certain resources?
Complexity	How difficult is it to understand the content?
Reliability	How accurate and current is the content?
Engagement	How engaging is the content?

4. How do conversational programmers' needs different from traditional learners of programming?

Care less about

- Implementing the code about technology X

Care about

- Limitations and benefits of using technology X
- Connections between different terminologies
- Software engineering processes and development structures
- Communication skills

5. To what extent do conversational programmers attempt learning programming syntax and logic?

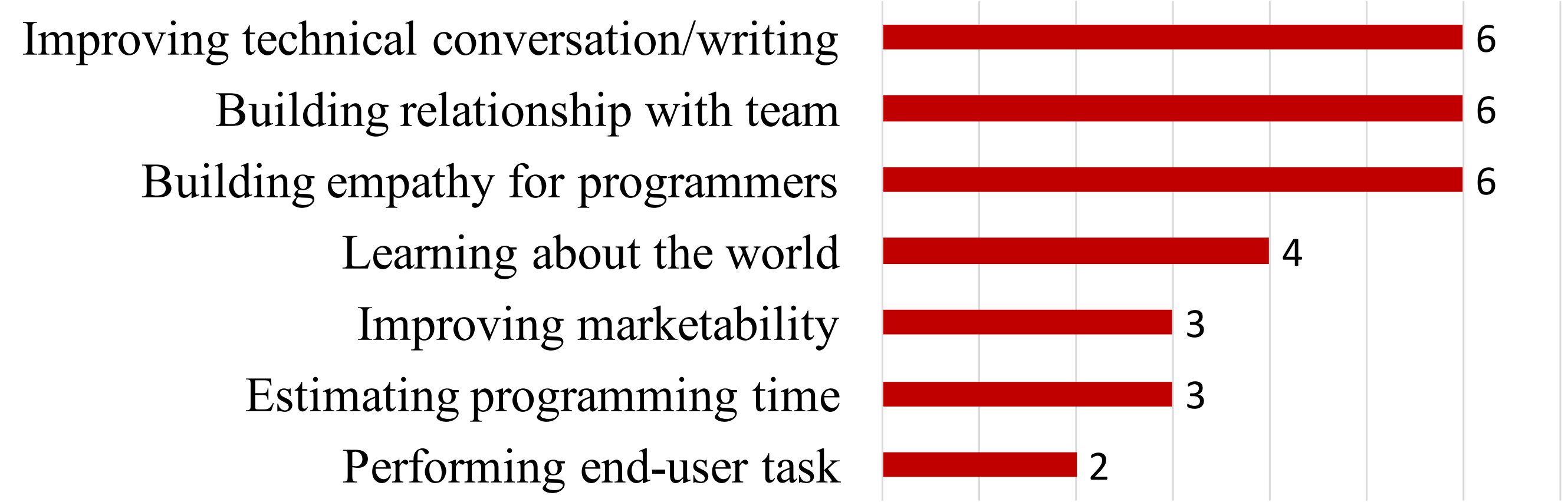
- In initial learning attempts, 18/23 participants said that they invested in learning the syntax and logic of a language
- BUT, only 9/23 participants felt that it was useful and necessary to have a syntax-level understanding when learning programming

6. What are conversational programmers' *perceptions* of challenges when learning programming?

- Programming knowledge is one language is not transferrable
- Learning curve is steep
- Have to go out of one's comfort zone
- Hard to phrase questions when asking for help
- Requires extra time commitment
- Trial and error does not work in conversations

7. To what extent does learning programming help vs. not help conversational programmers on-the-job?

Most conversational programmers in our study said that learning programming at the syntax level did not directly help them with their main motivation of improving technical conversations. However, the majority of participants (19/23) still had a positive attitude towards programming literacy and listed other ways how it helps them on-the-job.



Discussion/ Future work

1. How can we better understand the mismatch between conversational programmers' expectations and their actual learning achievements?

2. How can we design learning resources and tools that better support the unique learning needs and strategies of conversational programmers?

3. Why do conversational programmers still have positive attitudes towards learning programming even when it does not help them with technical conversations?

4. How should we think about educating conversational programmers?

References

1. Parmit K. Chilana, Rishabh Singh, and Philip J. Guo. 2016. Understanding Conversational Programmers: A Perspective from the Software Industry. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 1462-1472.