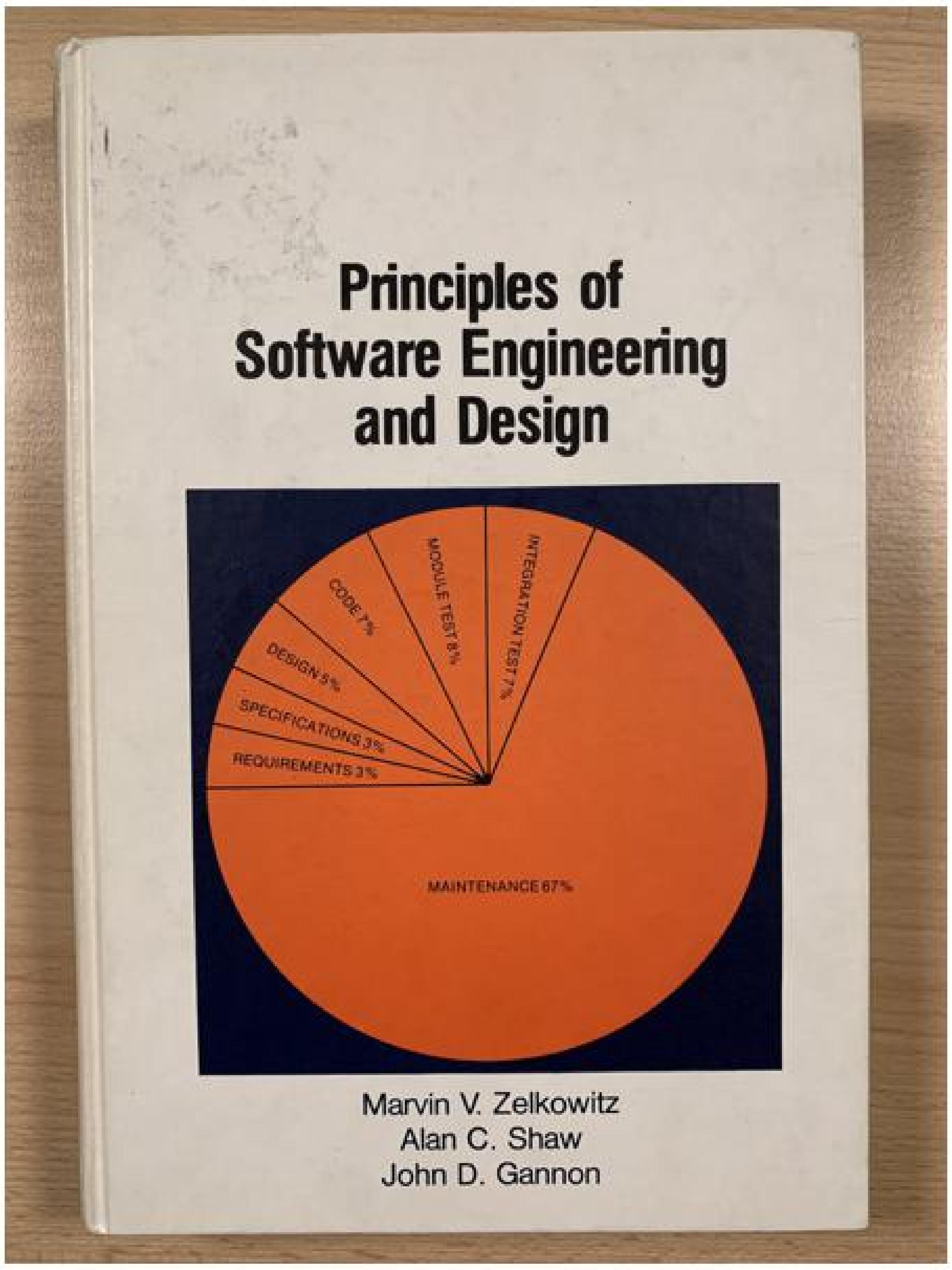
@girba

“Developers spend most of their time figuring the system out." Let’s dissect this a little. □ 1/

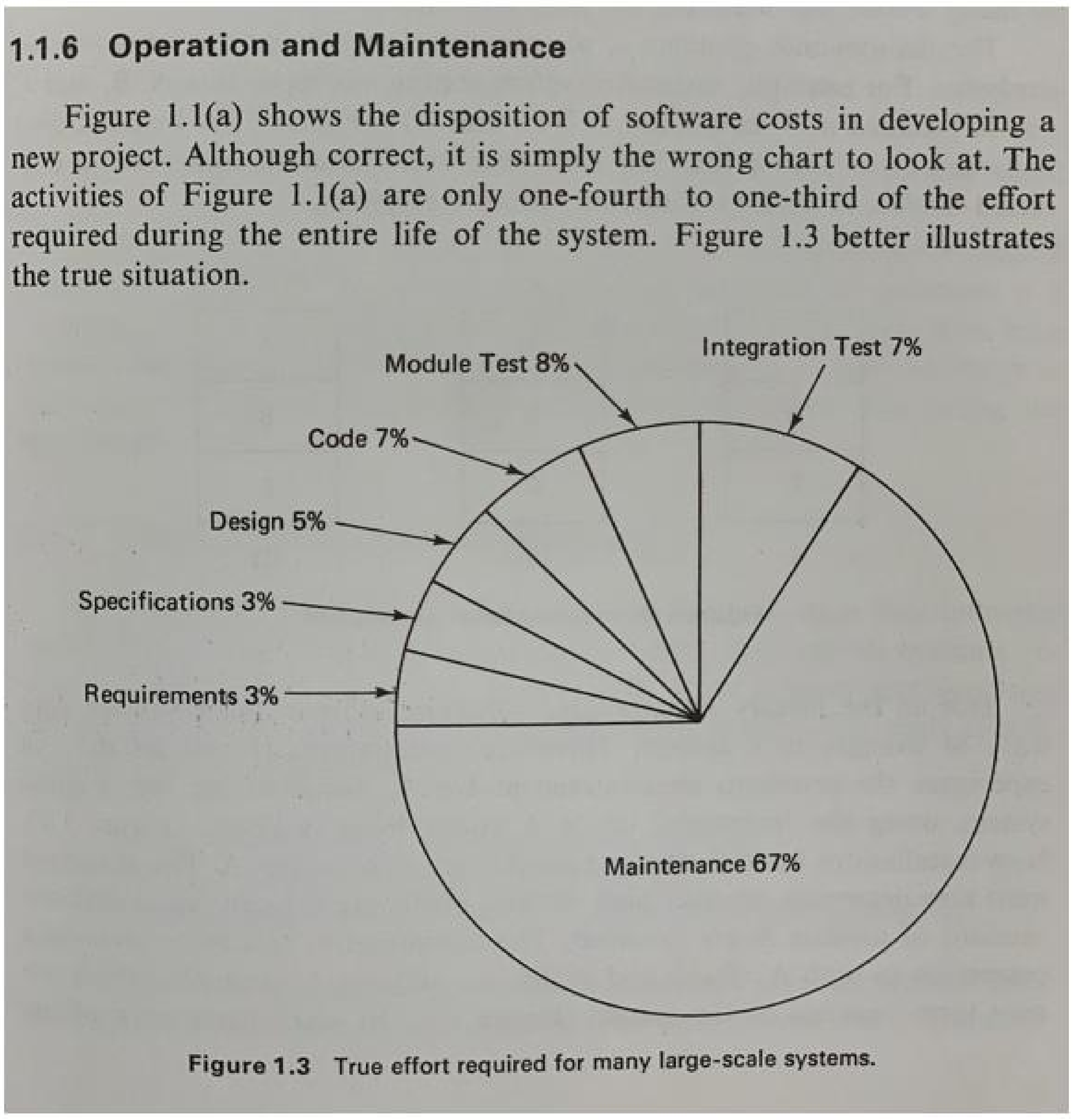
@girba

The oldest reference on the topic I know of dates back to 1979: Zelkowitz, Shaw, and Gannon. Principles of software engineering and design. Prentice-Hall Englewood Cliffs, 1979. 2/ https://t.co/6b44pymDKw



@girba

It said that most of the development time was spent on maintenance (67% in the book). Granted, the book does not specify how the figure was obtained. Still, it



1.1.6 Operation and Maintenance

Figure 1.1(a) shows the disposition of software costs in developing a new project. Although correct, it is simply the wrong chart to look at. The activities of Figure 1.1(a) are only one-fourth to one-third of the effort required during the entire life of the system. Figure 1.3 better illustrates the true situation.

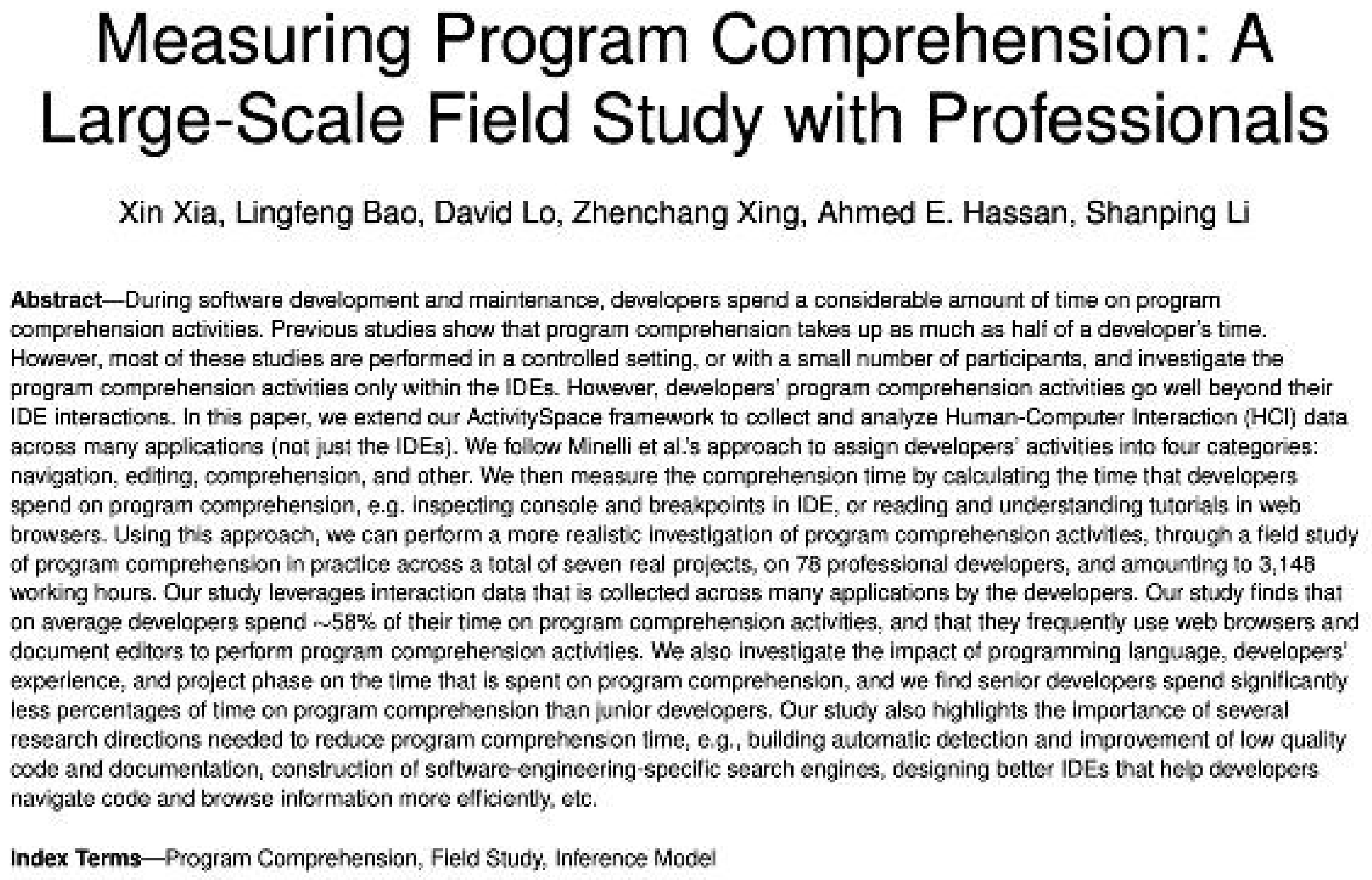
Module Test 8%
Integration Test 7%
Code 7%
Design 5%
Specifications 3%
Requirements 3%
Maintenance 67%

Figure 1.3 True effort required for many large-scale systems.

@girba

So, how far are we now, more than 4 decades later? Let’s look at this recent paper: Xia, Bao, Lo, Xing, Hassan, &amp; Li. Measuring Program Comprehension: A Large-Scale Field

[redacted]



Measuring Program Comprehension: A Large-Scale Field Study with Professionals

Xin Xia, Lingfeng Bao, David Lo, Zhenchang Xing, Ahmed E. Hassan, Shanping Li

Abstract—During software development and maintenance, developers spend a considerable amount of time on program comprehension activities. Previous studies show that program comprehension takes up as much as half of a developer's time. However, most of these studies are performed in a controlled setting, or with a small number of participants, and investigate the program comprehension activities only within the IDEs. However, developers' program comprehension activities go well beyond their IDE interactions. In this paper, we extend our ActivitySpace framework to collect and analyze Human-Computer Interaction (HCI) data across many applications (not just the IDEs). We follow Minelli et al.'s approach to assign developers' activities into four categories: navigation, editing, comprehension, and other. We then measure the comprehension time by calculating the time that developers spend on program comprehension, e.g. inspecting console and breakpoints in IDE, or reading and understanding tutorials in web browsers. Using this approach, we can perform a more realistic investigation of program comprehension activities, through a field study of program comprehension in practice across a total of seven real projects, on 78 professional developers, and amounting to 3,148 working hours. Our study leverages interaction data that is collected across many applications by the developers. Our study finds that on average developers spend ~58% of their time on program comprehension activities, and that they frequently use web browsers and document editors to perform program comprehension activities. We also investigate the impact of programming language, developers' experience, and project phase on the time that is spent on program comprehension, and we find senior developers spend significantly less percentages of time on program comprehension than junior developers. Our study also highlights the importance of several research directions needed to reduce program comprehension time, e.g., building automatic detection and improvement of low quality code and documentation, construction of software-engineering-specific search engines, designing better IDEs that help developers navigate code and browse information more efficiently, etc.

Index Terms—Program Comprehension, Field Study, Inference Model

@girba

This paper describes in great details how the figures are obtained. And it says that Comprehension took on average ~58%. Now, take a closer look at the third column

[redacted]

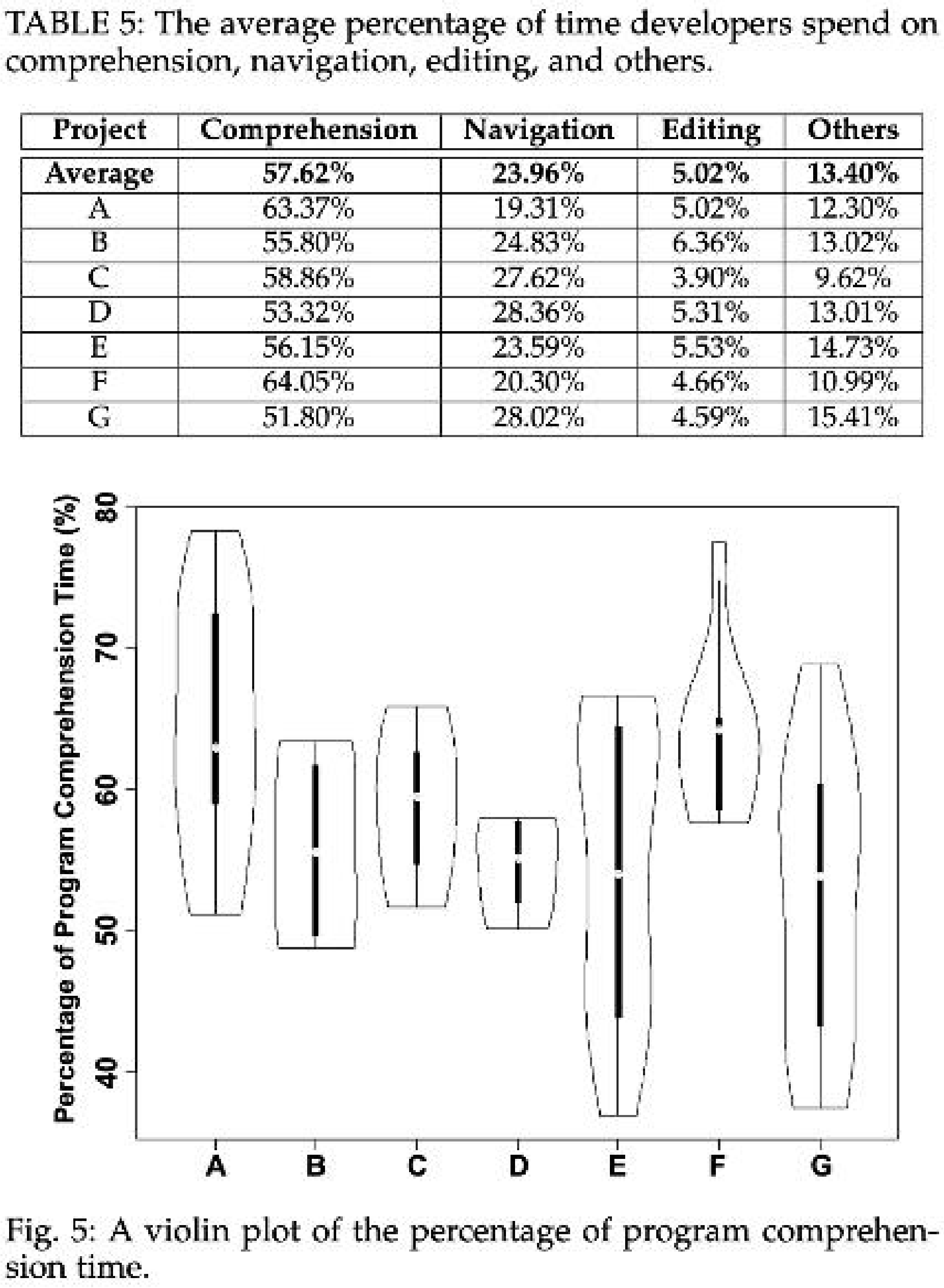


TABLE 5: The average percentage of time developers spend on comprehension, navigation, editing, and others.

Project	Comprehension	Navigation	Editing	Others
Average	**57.62%**	**23.96%**	**5.02%**	**13.40%**
A	63.37%	19.31%	5.02%	12.30%
B	55.80%	24.83%	6.36%	13.02%
C	58.86%	27.62%	3.90%	9.62%
D	53.32%	28.36%	5.31%	13.01%
E	56.15%	23.59%	5.53%	14.73%
F	64.05%	20.30%	4.66%	10.99%
G	51.80%	28.02%	4.59%	15.41%

Fig. 5: A violin plot of the percentage of program comprehension time.