Effects of Tutorials on End-User Programmer Feature Usage and Engagement in TouchDevelop** (temporary title)

Sergii and Irwin and Austin
Department of Electrical Engineering & Computer Science
Oregon State University
Corvallis, Oregon, 97330 Email: sergii,kwan@eecs.oregonstate.edu

***Abstract*—As programming becomes an increasingly valuable skillset in the modern world, there are many methods available to learn how to program. One method gaining popularity with more complex web apps is that of the interactive tutorial. We examine users published scripts on the TouchDevelop platform and compare interactive tutorials to traditional, copy-and-paste tutorial methods in terms of user outcomes. To accomplish this, we build a tool to facilitate large-scale data capture and analysis of TouchDevelop users and scripts. This tool is also capable of facilitating a much broader scope of research questions concerning the TouchDevelop population.**

I. Introduction

Programming has, over the last 60 years, become an increasingly popular discipline, both as a primary profession and as a hobby or additional job skill. Traditionally most self-taught programmers learn from a book or a website that intersperses writing with code snippets that can be copied and pasted into a source code editor. However, in recent years the advent of rich web apps that are able to respond to the user dynamically have begun to change how programming is taught. The website Codecademy [4], for example, presents all its tutorials via a web app that also functions as an IDE, complete with progress indicators, hints, and tests to make sure the users code is correct. The site Code School [5] advertises by saying Enjoy an education in the comfort of your browser. Another new source of interactive programming tutorials is TouchDevelop [9]. This platform is intended to teach and enable end-user programmers on touch-based devices such as smartphones or tablets to create their own scripts. As both a new programming paradigm and a new programming language, it necessarily involves a large amount of teaching users new to programming, TouchDevelop, or both.

- RQ1: Do end-user programmers even complete tutorials?
- RQ2: What effect does completing tutorials have on end user programmers' engagement?
- RQ3: What effect does completing tutorials have on end-user programmers' learning?

II. Background

A. TouchDevelop

TouchDevelop is an experimental programming platform for end-user programming on touch-screen devices. It is also designed a research tool, meaning that published projects are automatically open sourced, and a wealth of data on users and published scripts is made public (see [3]) on a growing userbase. However, while the JSON-based APIs are very useful for looking at specific scripts or examining scripts one at a time, large-scale relational-database style queries are unavailable in the TouchDevelop API.

As a result, studies of the TouchDevelop population such as Athreya et al. [1] have had to rely on a random sample of scripts, classified by hand. First and most importantly, this greatly limits the size of the sample. Secondly, it introduces personal bias and error into the process - Athreya et al.s study [1] describes the process of negotiation when researchers inevitably disagreed on classification of scripts.

One particular case of this was the category of No Meaningful Functionality scripts in [Balajis paper]. This made up a significant portion of the TouchDevelop script base, and included tutorials, among other published scripts that would not be useful to anyone besides the author. However, due undoubtedly to a desire to not waste time, this category of scripts was not detailed further.

Li et al. [2] studied the behavior of users in TouchDevelop. They found that 68.3% of TouchDevelop users published one or two scripts and then never returned or produced more content. This begs the question: why? This is an especially important question, as TouchDevelop is a novel approach to programming and is intended for end user programmers, many of whom may have had no previous experience programming at all, and thus do not even know basic concepts such as variables and functions.

B. Tutorials and Community Engagement

****Programming languages are difficult to learn, and many resources have been devoted to increasing retention and interest in end-user programming languages.** A new programming language is difficult to learn, especially for an end-user developer. These end-user developers often need to learn on their own, without the support of peers.

Analysis of end-user developer repositories [1], [2], [3], [4] has shown that reuse between scripts is very low. In addition, user retention is also very low. In what ways can we improve user retention?

However, little research has been done specifically investigating what effect making training material, such as tutorials, available to end-user programmers has on their engagement and adoption of the platform.

**Need argument here that we should understand effects of tutorials on user engagement Open-source projects have their own ways of maintaining documentation [5] (cite other papers about the use of documentation for learning APIs, libraries, and programming). However, Carroll and Rosson [6] identified that tutorials were also not appropriate for "active users", who were more concerned about completing tasks than learning.

****The TouchDevelop environment had interactive tutorials and non-interactive tutorials and they were different in a few ways** The TouchDevelop environment has interactive tutorials and non-interactive tutorials. From a learning standpoint, interactive tutorials seem superior; for example ... Stencils (Kelleher), LemonAid (Andy Ko). There are reasons to consider non-interactive tutorials. For instance, a study of *opportunistic programmers* identified them as quickly copying and pasting snippets from tutorials to learn new skills and approaches, which would not be possible with an interactive tutorial [7].

**Strangely enough, the tutorials don't have any accompanying plain English language text with them. Some of them have documentation, some do not.

There are two types of TouchDevelop tutorials available: interactive and non-interactive. Interactive tutorials engage the user into tutorial completion more actively than non-interactive tutorials by providing frequent direct suggestions on what user is supposed to do next. On the other hand, non-interactive tutorials provide tutorial instructions and source code snippets without guiding the user directly.

III. Method

To conduct our study, we used the public TouchDevelop API to extract 35000 scripts which we stored in a MySQL database.

A. Analysis

We used SQL queries and statistical methods to examine our data.

**Please verify To identify a script as a tutorial, we... (describe how we identified the initial tutorial script. Was it that we went to the web site and then looked at the site? Or, based on naming? Or something else?)

Sergii please verify Scripts (that users create as a result of following a tutorial**???**) are tagged with `#doc`, `#tutorials`, or `#interactiveTutorial` in the case of an interactive tutorial.

We identified that tutorials were tagged with the #doc hashtag and that there are a total of XX tutorials.

****When appropriate, we removed some usernames from our analysis.** We contacted the TouchDevelop team and learned further that certain author names, including "TouchDevelop Samples" and "TouchDevelop Documentation", were operated by Microsoft employees; in our author data analysis we remove these from the analysis. **Be really specific when we actually do this. we might have to rewrite this statement

**There are some group accounts based on the account name and activity. We cannot be certain, however, we can only guess and this has to be a threat to validity.

Description	Number
Published scripts	40464
Authors	18153
Features (as labelled by TouchDevelop)	11789
Tutorials**what is this exactly? check	472

TABLE I. Summary of Dataset

TABLE II. Per-author statistics

IV. Results

****We analyzed this amount of data and found these summary statistics.** We analyzed **40464 scripts composed by 18028 authors. Of this data, **num scripts were identified as created as a result of following a tutorial.

(**Identify probably three groups of authors: those who wrote 0 tutorials, those who wrote 1, and those who wrote more than 1. Compare them.) **Plot the distribution of authors who wrote [0, 1, 1+] tutorials.

A. RQ1: Do end-user programmers even complete tutorials?

In general, TouchDevelop script authors are only somewhat willing to attempt tutorials. Of the 18022 users who have published at least one script, we observed that only 2981 (16.5%) attempted a tutorial and 2513 (14%) of all users completed one or more tutorials.

**Implication? Despite this finding, our data still shows that a large number of users have successfully published scripts. This suggests that many TouchDevelop users are either familiar with programming and did not require tutorials or that they learned or were taught TouchDevelop offline (for example, as part of an outreach effort or by friends). There is also the possibility that users generally avoided the official tutorials and used other resources, but non-Microsoft tutorials are rather rare. **One way to explore this further is to examine the features of scripts that people who didn't use tutorials had. If they are filled with advanced constructs then we can assume that they have some prior knowledge

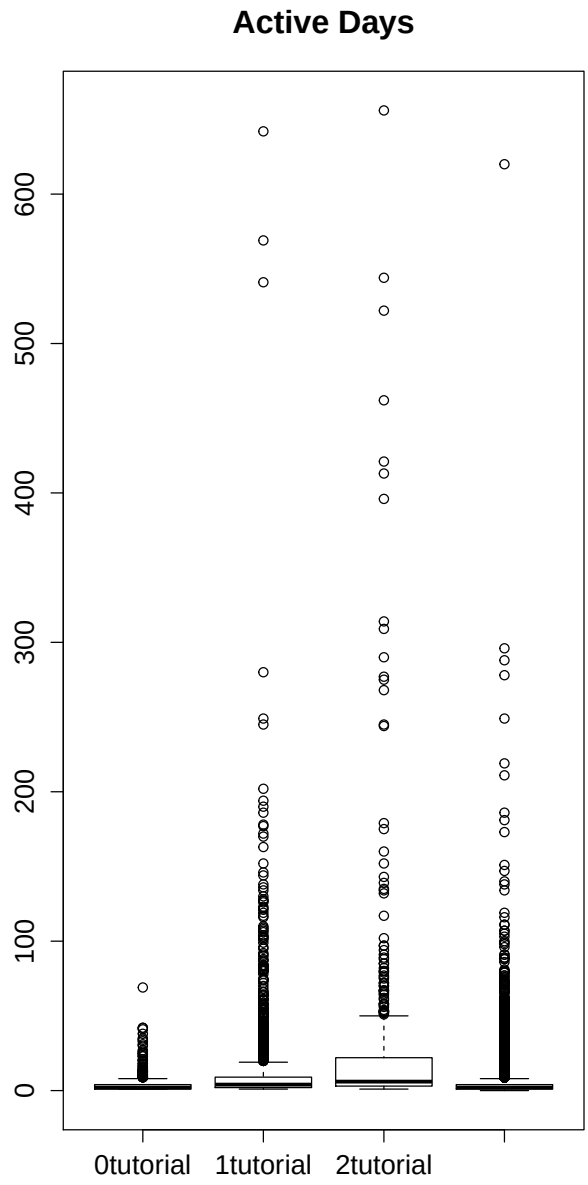
Authors who published one or more scripts	18015 (100%)
Authors who did not attempt a tutorial	15045 (83.5%)
Authors who attempted or completed a tutorial	2970 (16.5%)
Authors who attempted (but did not complete) a tutorial	468 (2.6%)
Authors who completed one tutorial	1977 (11.0%)
Authors who completed more than tutorial	525 (3.0%)

TABLE III. Proportion of TouchDevelop users who complete tutorials. Only 16.5% of users attempt tutorials and 14% complete at least one tutorial.

**The correlations are... The correlations appear in Table IV-A. The correlations are really strong for a lot of stuff and this needs to be interpreted. Many of these make sense, as one would expect that the more scripts someone has, the more features they'll use and the more subscribers/score they'll get.

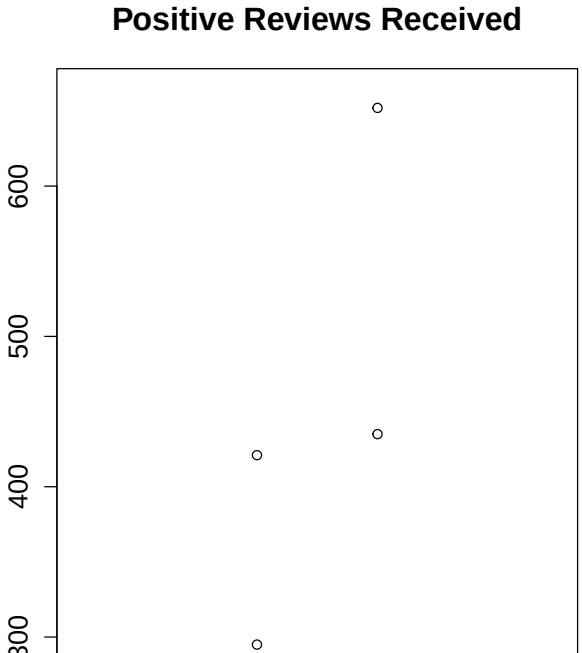
	Scripts	Tutorials completed	Features	Days active
number_of_scripts				
number_of_completed_tutorials	0.50***			
features	0.26***	0.15***		
activedays	0.31***	0.08*	0.74***	
receivedpositivereviews	0.09**	0.05	0.52***	0.60***
subscribers	0.07*	0.11***	0.34***	0.43***
score	0.28***	0.10**	0.91***	0.90***

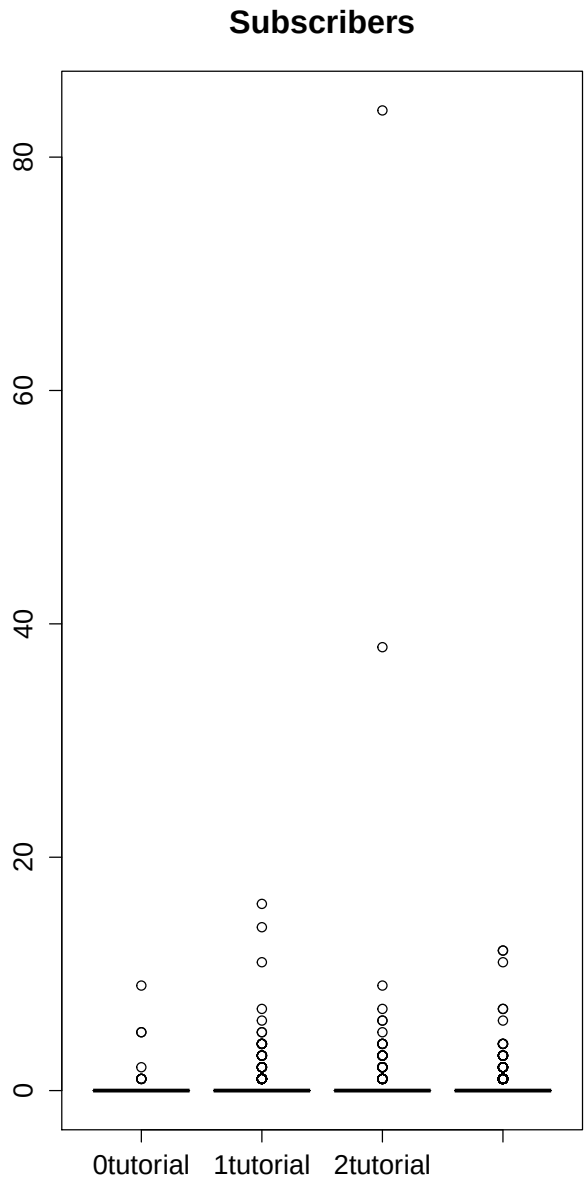
TABLE IV. CORRELATIONS BETWEEN COUNTS OF ITEMS PER AUTHOR
**THIS NEEDS A BETTER CAPTION

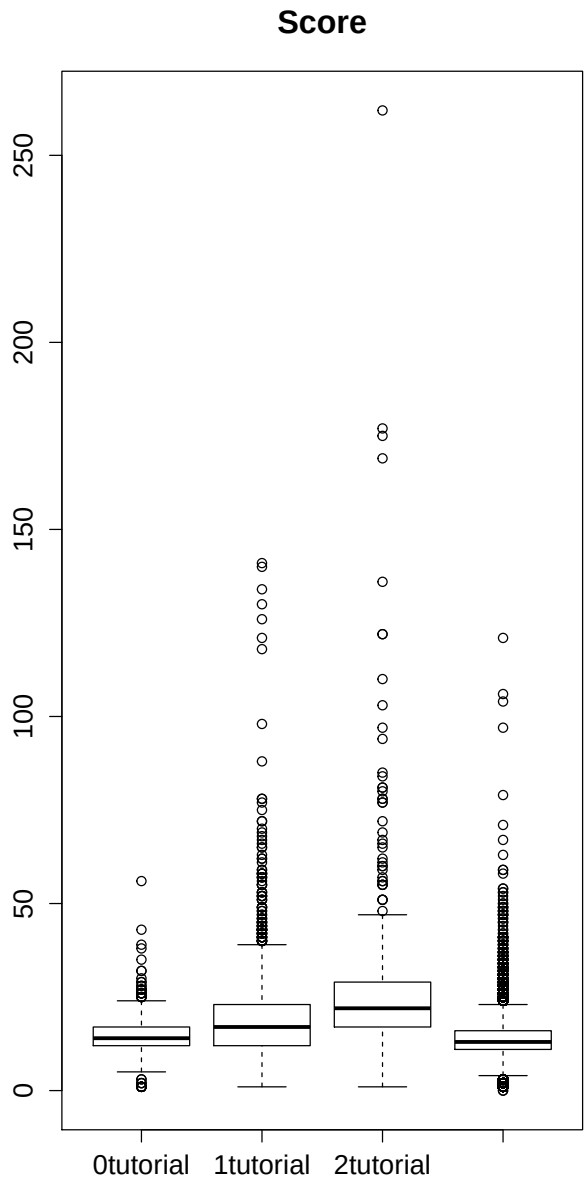


B. RQ2: What effect does completing tutorials have on end user programmers' engagement?

We compared the average usage statistics of authors who completed zero, one, and more than one tutorial in TouchDevelop.







C. RQ3: What effect does completing tutorials have on end-user programmers' learning?

We built a linear regression model to identify factors that influenced tutorial completion.

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

REFERENCES

[1] B. Athreya, F. Bahmani, A. Diede, and C. Scaffidi, "End-user programmers on the loose: A study of programming on the phone for the phone," in *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*, 2012, pp. 75–82.

[2] C. Bogart, M. Burnett, A. Cypher, and C. Scaffidi, "End-user programming in the wild: A field study of coscripter scripts," in *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on*, 2008, pp. 39–46.

[3] S. Li, T. Xie, and N. Tillmann, "A comprehensive field study of end-user programming on mobile devices," in *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, 2013, pp. 43–50.

[4] K. T. Stolee, S. Elbaum, and A. Sarma, "Discovering how end-user programmers and their communities use public repositories: A study on yahoo! pipes," *Information and Software Technology*, vol. 55, no. 7, pp. 1289 – 1303, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095058491200211X

[5] B. Dagenais and M. P. Robillard, "Creating and evolving developer documentation: Understanding the decisions of open source contributors," in *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE '10. New York, NY, USA: ACM, 2010, pp. 127–136. [Online]. Available: http://doi.acm.org/10.1145/1882291.1882312

[6] J. M. Carroll and M. B. Rosson, "Paradox of the active user," in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J. M. Carroll, Ed. Cambridge, MA, USA: MIT Press, 1987, pp. 80–111. [Online]. Available: http://dl.acm.org/citation.cfm?id=28446.28451

[7] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer, "Two studies of opportunistic programming: Interleaving web foraging, learning, and writing code," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1589–1598. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518944