

# Learning Rapidly Changing Frameworks

Ryan Turner  
Georgia Institute of  
Technology  
rturner63@gatech.edu

**ABSTRACT**  
WIP

**Author Keywords**  
just in time

## INTRODUCTION

Some background on what it is you're talking about. Typically, this will include a cursory literature review just to define the problem area. It will culminate in the question the paper is answering or the problem the paper is solving, typically followed by a brief summary of the contribution itself.

For creators of technology, educating users is often a challenge. Some creators tend to introduce formal courses, while some others rely on users referencing documentation and trial-and-error. In some, third parties have created learning materials like weblog tutorials or even reference examples.

Learning about a rapidly-changing technology like augmented reality and artificial intelligence seems to depend on informal education. For these, often documentation and a sample are all the materials provided. Looking at more established topics like desktop computing and service management, the case is quite different. Corresponding certifications exist like CompTIA A+ and ITIL Foundation with accompanying curriculum and a network of professional teachers. The options vary greatly as the technology's maturity level changes.

There is a wide body of knowledge on informal learning, as well as learning by developers. However, there has not been significant study on how learning occurs in these contexts. How do developers learn rapidly-changing frameworks?

## RELATED WORK

In this research, non-institutional learning methods will be considered. Informal learning is one of these methods [2], which is defined as learning with little structure and often as the byproduct of some other activity [2]. Informal learning is "relevant to practice in many cultures and contexts" and "[takes] place wherever people have need, motivation, and opportunity for learning" [2]. While this method is often looked at from a business context [2,5], motivation could

come as uncertainty of creating a new solution or resolving a bug in code. A social component is also defining for informal learning. For example, while the trigger may be considered locally to an individual, it is often the result of an external change such as results of a tester's work or a request from a product owner. Informal learning takes place around all of us day to day.

In examining how developers learn rapidly-changing frameworks, we must start by reviewing their options. Developers have many solutions available for learning (see Research Problem). The next paragraphs describe the informal learning activities that take place within a subset.

In research of programmers use of tools, [3] show how peer interaction leads to more discovery. Discovery is more specifically called "the first stage of some kinds of learning." Murphy-Hill contrasts peer interaction from Marsick's definition of informal learning; however, when considered according to the definition above, the distinctions fade. In this work, Murphy-Hill demonstrates how situations like happenstance interaction, pair programming, and even change notification often result in peer observations and recommendations. These represent the "discoveries" of tools, but in some cases they also represent the teaching material itself: with pair programming, the peer interaction often creates an incidental situated learning experience. The same may apply to frameworks as well.

Hackathons are also proposed as "excellent informal learning platforms" [4]. A hackathon is a "fast-paced event where competitors work in teams to go from an idea to working software or hardware within a single day or a weekend." Like in [3]'s work, the authors cite peer-learning as common place in this setting. Consistent with the definition of informal learning, at hackathons the problems create the need, the gamification create the motivation, and the industry mentors plus online resources create the opportunity. While the research focuses on learnings within teams, they also recognize that the learning environment created at a hackathon excels at producing industry-relevant learnings and skills. It is considered a "great opportunity to learn" by participants [4]. Frameworks may be introduced or brushed up on at these events, though the researchers did not consider this.

MOOCs present as popular online digital learning tools today. Popular examples include Coursera, edX, and Udacity. When considering learning theories within MOOCs, there tend to be either connectivism-driven MOOCs (cMOOCs) or extension MOOCs (xMOOCs) [7]. These are significantly different in their application of learning models, and more needs to be done to make use of informal, personalized, or professional

learning on these platforms [7]. From anecdote, MOOCs appear to be a popular tool to learn frameworks.

The proposed research relates to a topic proposed by [5]: “What antecedents and conditions facilitate continuous learning, especially informal learning, and knowledge sharing?” In considering the form of learning, [5] mention that “informal learning may be equally important to or even more important than other forms of learning.” In fact, informal learning is so prevalent in organizations that it may account for up to 75% of learning [1,5]. Yet, the future issue is still proposed: “What are the antecedents and consequences of informal learning?” [noe2014learning].

In the problem itself, the term “frameworks” is used. For the sake of this research, a framework is defined as “a form of software reuse that primarily promotes the reuse of entire architectures within a narrowly defined application domain” [6].

METHODOLOGY

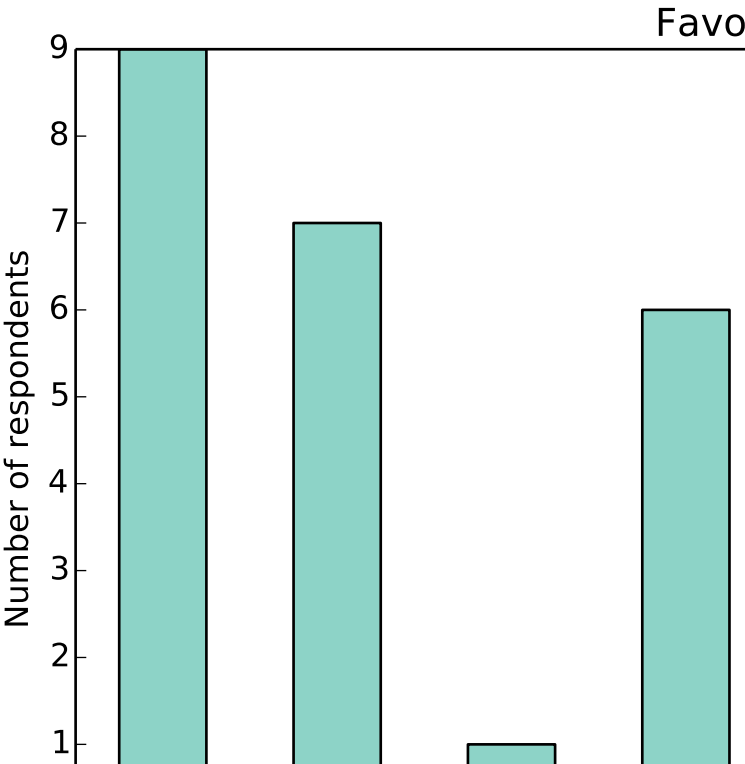
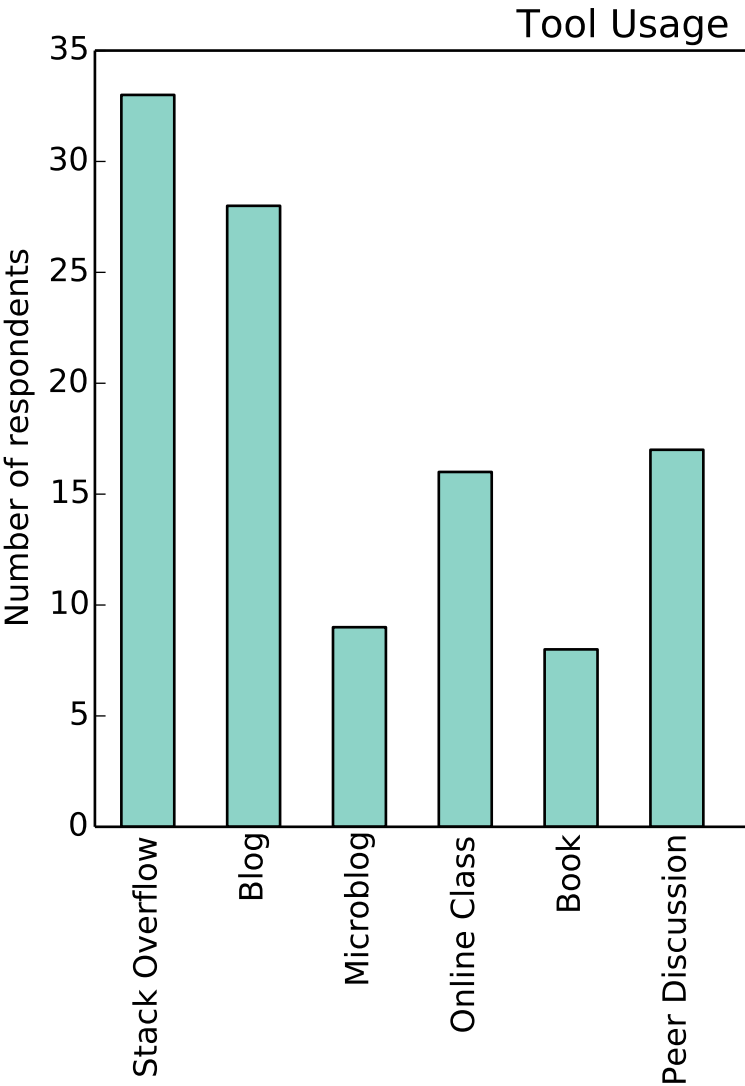
To better understand learning in this area, developers use of learning tools was reviewed. Ideally, developers would have been observed in their learning. Instead, a survey approach was taken. This is partly due to the feasibility to observing these interactions as well as the importance of capturing responses from an international audience. Finally, this approach permitted developers to report on their perceptions of the learning: why did it occur, how successful was it, and how soon was it put to use.

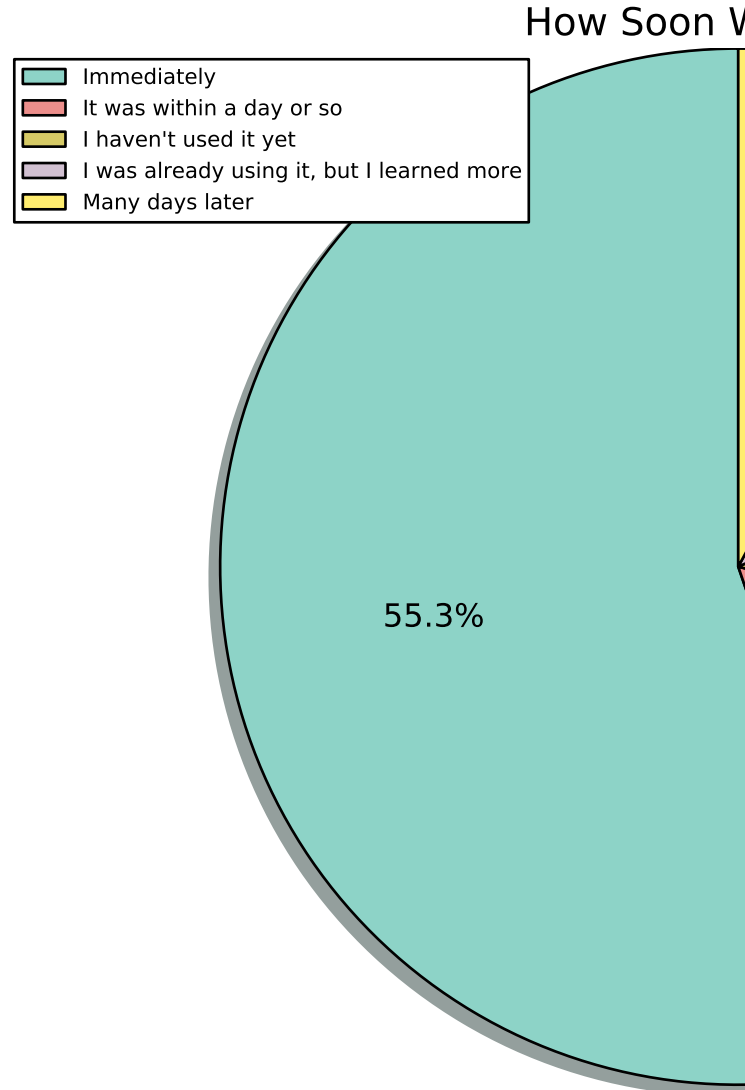
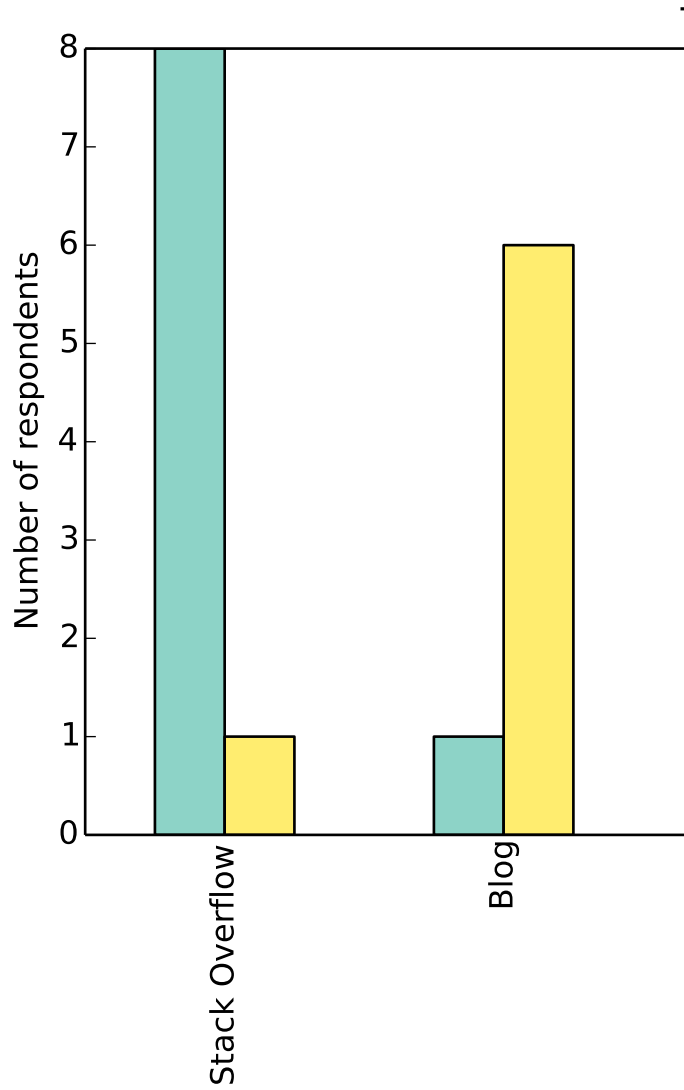
The

The methodology tells the reader how you set about answering it. The important thing about the methodology is that exists separate and before the results: it tells the reader why they should trust the results.

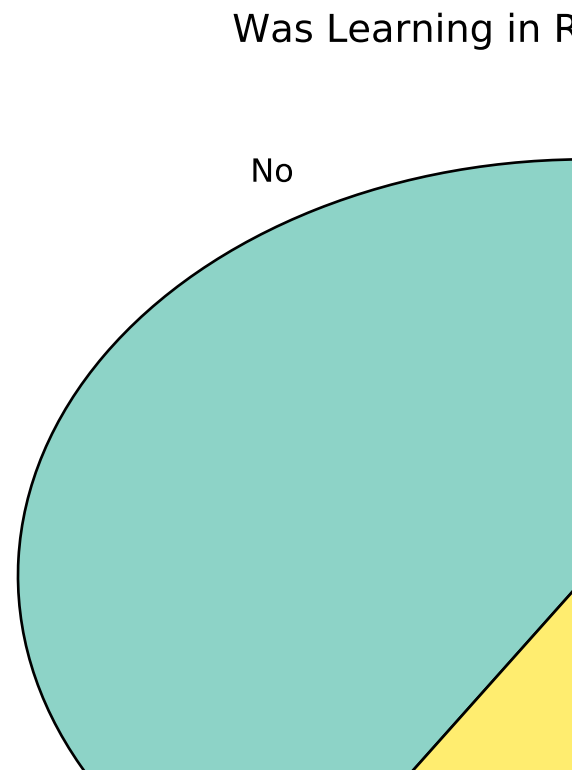
RESULTS

What was the result of the investigation or evaluation? This is typically what the entire paper is building toward: some assertion that the solution worked or some answer to the question that was raised.





Learning in response to change



## LIMITATIONS

Another underappreciated section of any strong paper: the limitations section clearly articulates exactly how generalizable the conclusions are. For example, if work was done in the context of a middle school, then the results may only be generalizable to middle schools. If there were clear potential lurking variables in the methodology, then those would be disclosed here. The importance of the limitations section is that it clearly and honestly articulates how far the contribution goes. In my experience, any limitation you identify will not be held against your paper, but if the reviewers have limitations you don't acknowledge, they'll be far more reluctant to accept the contribution. This is what separates research from advertising.

## CONCLUSION

A summary, basically. Reiterate the context, the problem, the solution, the results, and the limitations.

## References

1. DJ Bear, HB Thompson, CL Morrison, et al. 2008. Tapping the potential of informal learning: An astd research study. Alexandria, VA: American Society for Training and Development.
2. Victoria J Marsick and Karen E Watkins. 2001. Informal and incidental learning. *New directions for adult and continuing education* 2001, 89: 25–34.
3. Emerson Murphy-Hill and Gail C Murphy. 2011. Peer interaction effectively, yet infrequently, enables programmers to discover new tools. *Proceedings of the acm 2011 conference on computer supported cooperative work*, ACM, 405–414.
4. Arnab Nandi and Meris Mandernach. 2016. Hackathons as an informal learning platform. *Proceedings of the 47th acm technical symposium on computing science education*, ACM, 346–351.
5. Raymond A Noe, Alena DM Clarke, and Howard J Klein. 2014. Learning in the twenty-first-century workplace. *Annu. Rev. Organ. Psychol. Organ. Behav.* 1, 1: 245–275.
6. Alessandro Pasetti. 2002. *Software frameworks and embedded control systems*. Springer Science & Business Media.
7. Ahmed Mohamed Fahmy Yousef, Mohamed Amine Chatti, Ulrik Schroeder, M Wosnitza, and H Jakobs. 2014. A review of the state-of-the-art. *Proceedings of CSEDU*: 9–20.