Marshall Copeland

CEN 4020

Chris Mills, Ph.D.

December 8th, 2021

The Advantages and Issues with Reusing Code from Online Sources

Introduction.

In the world of software engineering, people are paid to produce code that works. Now a common tactic in business is to use the cheapest way that takes the least amount of time and money to achieve the same results as another more costly method. And when deadlines are approaching, it may seem reasonable to cut a few corners to get the job done. If you’re responsible for producing code, the idea of reusing code so you don’t have to reinvent the wheel seems appealing. This is especially true since there are many different sources where someone can look for code online and be able to find exactly what they are looking for. But should this be as common a practice in the workplace and in the world as it is today? It is easier than ever to copy and paste code from online sources and incorporate it into your own project, however there are some detrimental side effects that many programmers don’t even consider, including ethical violations and security concerns. Like anything, there are many advantages and issues to reusing code, and it is important to look at both sides to come to a conclusion of whether or not this initially harmless practice should continue.

Advantages.

Writing code can be a tedious and exhausting process. You need to analyze the problem that needs to be solved, and then you need to try and figure out the most efficient way to solve that problem, with thoughts of cost, complexity, and the actual size of the finished product, all before you even begin touching the keyboard. There is so much thought that needs to go into a piece of code, especially when you are writing something that is going to be used by thousands of people millions of times all over the world. When that kind of pressure rests on a couple hundred lines of code, it can seem overwhelming. So why shouldn’t you be able to take a few shortcuts that rapidly speed up the production time, without having to sacrifice too much in return.

We live in a time where everything is online, especially in a field like software engineering. Whether it’s distant Zoom calls that act as meetings, tools like GitHub that serve as a home for the team’s code, or even the paycheck that you receive from doing all that work. So, it makes sense that there would be areas online where these people can go to discuss their issues, and even seek help from more experienced workers in their field. Places like StackOverflow, where anyone with a background in topics like computer science, programming, and other things can initiate dialogue and ask questions about problems they have. And the best part about this is that everyone is able to see the conversations that happen on the website, so the information is spread from the source to thousands of people who might have the same issue. Then they can use that solution in their own program instead of having to spend unnecessary resources solving it by themselves. Code coming from these sources can be reliable at times. Coming from that study there can be some instances where the code is there to correctly serve a purpose and is able to be used by other people in their projects if they choose to do so. However, it is important to note that they don’t recommend relying on these places to get your code, it should only be supplementary (Zhang et al). If places like StackOverflow are able to provide code to people to use in their own projects, why shouldn’t that be possible? It is much faster to get code from an online source instead of having to start from scratch, which can save lots of money in production environments with tight deadlines that must be followed.

Another reason someone might seek to use another person’s code is them wanting to use something that has already been thought through so there will be less time spent on debugging compared to something that is built from scratch. It is a common practice to use templates in computer science to limit the amount of redundancy that occurs in code writing, and in this case these pieces of code can be viewed as templates that just have to be slightly tweaked to fit in with the rest of the project. There are many reasons to use another person’s code, especially since it is coming from an online webpage that encourages its users to post solutions to problems that somebody might have and use that solution in their own project. In fact, it is not only small project coders that take advantage of this open-source platform. In a study done in 2017, researchers found that “…15.5% of the studied apps have reused code from Stack Overflow” when looking at uploaded Android applications (An et al). If there were large ethical or financial issues with this practice, it wouldn’t be as common as it is today, as it seems that places like StackOverflow are one of the many tools used by software engineers. If every great idea had to begin from scratch, and couldn’t be built upon other people’s accomplishments, humanity would not have gotten to where it is today, and to limit this practice seeks to limit the human spirit of creation.

Issues.

There are many issues that can arise when somebody takes code from an online source and uses it in their own program, acting like it is all their work, and it has never been easier to do. A common trope about computer science is that programmers are lazy. This can be a good thing, as they can think of the fastest way to get something done using the least amount of effort, which is helpful in a business where results are prioritized. But issues begin to arise when this becomes commonplace in the coding culture, which it is fair to say that it has. When someone is producing code in a business environment, that person is being paid to make that piece of code. The key phrase is “that person”. Not some random person on the internet, no, the person being paid is the one who is obligated to produce that code. If the company wanted the code to be written by someone else, they would have decided to spend their money elsewhere. When someone comes in to do a project, and they show up with somebody else’s code acting like they’re finished, that seems very disrespectful to both the company and the team that they are working with to get the project done. Not only is the company being cheated out of money, and not only is the team being cheated out of talent, but the person who is just grabbing code off the internet is cheating themselves.

Many times, the code that is being taken isn’t even correct. According to an article, “The most copied StackOverflow Java code snippet contains a bug” (Catalin Cimpanu for Zero Day). This is a dangerous example of what can happen when people are careless and a little bit too trusting. The article then goes on to say that the piece of code has made its way into over 6000 projects on GitHub over the 9 years that the code has been out, which goes to show the reach that places like StackOverflow have. Now there might have been some people who took this code and then tested it to make sure that everything was working properly before incorporating it into their own project, but for 6000 people that wasn’t the case, as the code found in the GitHub was the same incorrect code originally posted by the author. That is just another example where the programmers are lazy and seeking to cut corners, but it backfires on them. Think about the countless other snippets of code out there that have bugs in them. And even worse, there are bugs that can present some very serious security concerns that can leave an application extremely vulnerable to hackers or other adversaries that exploit the error in the code. According to a study published in 2020, snippets of C++ code were analyzed to see if they contained any security vulnerabilities that could compromise the integrity of an application that it is incorporated with. Of “…the 72,483 reviewed code snippets used in at least one project hosted on GitHub, we found a total of 99 vulnerable code snippets” (Verdi et al). This is an extreme problem, because all it takes is one of these issues to potentially compromise a company and cause thousands, if not millions of dollars in damages. If this was something that was just up to chance, the odds are not the worst, but the fact that this can be totally avoided really gives you an idea of the software engineering community, and what they will do to get results. And that is not the only study done on the security of code coming from websites like StackOverflow. When looking at the impact of copy and pasting code on Android application’s security, it was found that “…196,403 (15%) of the 1.3 million Android applications contain vulnerable code snippets that were very likely copied from Stack Overflow” (Fischer et al). Not only is this a security issue, but it is a looming financial issue, and it seems like someone is begging to lose their job or even get sued if something bad comes out of it.

Copying and pasting code from online sources is just ignorant and extremely dangerous, and there needs to be a serious reevaluation of the entire community for supporting such a terrible practice like this. There are real world issues that are not only possible but have already happened and impacted thousands of livelihoods. The fact that this is not just used by programmers, but it is often encouraged to use online sources to solve coding issues, shows that things need to change. It’s not too late to drop this bad habit, and hopefully people begin to look at this in a different light. Otherwise, a massive problem will arise, and it could come from your very own “secure and tested” code.

Reflection.

I think that there are some situations where using code from an online source may be a good thing. If you are writing a program, and you are in need of help, it is perfectly acceptable to go to a place like StackOverflow or GitHub and see how somebody else solved a similar problem. It is the same as asking a friend, teacher, or peer for help in any other situation and they give you advice coming from their own personal experiences. One of the most important things in the field of software engineering is being able to grow and learn about new things, and it should never limit someone’s ability to understand new topics or solidify present ones. It is very hard to think about where we would be today in a society that did not have helpful information regarding coding practices shared to the public, as it seems so commonplace today. It has allowed for significant growth in many different programmers as they are able to research almost any aspect of the field and immediately be able to learn how to do it. The beauty of the computer science community is that so much information is out there because the authors want people to learn from their mistakes and achievements, so a better generation of programmers can replace them when they retire.

If learning how to code was limited to a classroom setting where the content moves at a snail’s pace through the semester, and is forgotten mere months after the final, the progress that we have made in software engineering would grind to a halt as less and less coders applied for computer science related jobs. When content is made available online, people can learn what they want to and are interested in, which will lead to the retention of the topics, and the more topics you are interested in, the more you learn. And in the job market, if you only know how to do certain things, you will never be as desirable as someone who knows how to do more, especially in a competitive field like this. The only thing that is different is that the answers are online instead of in person, which should have no impact on your learning ability.

Only one issue can arise from “borrowing” code, and that is copying straight from the source and pasting it in your own program. Not only is that dangerous, as you are introducing foreign, possibly untested code into your application, but you are also directly taking from somebody else and passing it off as your own work. Plagiarism is a serious offense in the academic world, and it should have the same consequences in the computer science field. It is never ok to do that because you are not getting the permission from the original author, so they don’t receive the credit that they deserve, and you are cutting corners in a team setting because everybody else is putting in effort to get the project done. When you show up with code that you didn’t even write, you are disrespecting your collaborators who did spend time writing their own code. Imagine if everyone showed up with recycled code for a project, there would be no new ideas or innovation in the space like there is today, just the same old code reused over and over for every new project. That is luckily not the case today, and hopefully it won’t become the future’s either.

Conclusions.

Whether you are the one writing the code on StackOverflow or you’re the one using it as a reference, there is nothing wrong with having a space where people can go to have coding questions answered and where people can talk to each other about code in general. Problems begin to arise, however, when code that is posted has security issues or has bugs that the author did not realize and passes off as perfect code for other people who will happily copy and paste it. There lies another problem of people ripping code and passing it off as their own in projects, thinking that there are no problems associated with the practice. Being lazy and slacking off are not good professional qualities to have, so when you take code and drop it right into your project you are cheating the company, the author of the code, your team members, and most importantly yourself. It is perfectly fine to look at code online as a reference or guide, as we are all learning new things each time we touch the keyboard. But as soon as plagiarism and dishonesty is involved, all of the merit of learning dissolves due to the true nature of your actions. When you have permission to use the code, and the company is authorizing the practice, a lot of time and money can be saved, which is very important in the software engineering space, but when you cross that line of stealing, there is no turning back. There are many advantages and issues of using code from online sources, but as long as you are learning how to be a better coder and are learning from your mistakes, you will be able to make your own decisions as to how you write your code.

References.

1. <https://ieeexplore-ieee-org.proxy.lib.fsu.edu/stamp/stamp.jsp?tp=&arnumber=8453166&tag=1>
2. <https://ieeexplore-ieee-org.proxy.lib.fsu.edu/stamp/stamp.jsp?tp=&arnumber=7884629>
3. <https://www.zdnet.com/article/the-most-copied-stackoverflow-java-code-snippet-contains-a-bug/>
4. <https://ieeexplore-ieee-org.proxy.lib.fsu.edu/stamp/stamp.jsp?tp=&arnumber=9195034>
5. https://ieeexplore-ieee-org.proxy.lib.fsu.edu/stamp/stamp.jsp?tp=&arnumber=7958574