Marshall Copeland

CEN 4020

Chris Mills, Ph.D.

October 22, 2021

The Future of Software Engineering in 5, 10, and 50 years

Introduction.

Software engineering is a relatively new field that has come up with the need to efficiently develop software for businesses. As we continue into this age of technology and era of computers being a necessary aspect of human lives, this need will only increase. That is why there will always be a demand for the people who make the software that has become such an integral part of our society. However, in the coming years, it is up to speculation as to how the process of developing software might look like, especially as we look forward to 5, 10, or even 50 years away from now.

Software engineering in 5 years.

Software Process:

In 5 years, there will be more than enough time for there to be change in the software engineering world. There might not be many significant changes, and the way people actually produce the software will likely look the same, but there will be many places where the process has been refined to where the production is seamless, and teams will have more effective communication tools. I believe software engineers will likely rely heavily on Agile-styled development as it is the preferable method today, but a more streamlined version. I think that requirements will still be gathered by clients through the usual methodologies that are used today, as it is likely going to be hard to get the top people in businesses caught up in the latest and greatest advancements in technologies, but they will likely have a better understanding of the development and the way the industry works. As far as code, I think there will begin to be automation where there is software that is able to write the code for other projects. There is already a program called GitHub Copilot, which on their website states “With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor” (GitHub Copilot). This already makes writing software easier, so in the not-so-distant future it isn’t unrealistic to assume there will be some sort of automation in software development.

Software team composition and organization:

I still believe that software will be developed in teams because that is such an important aspect of the process now, and for that to change it would mean there will be many other drastic changes that just aren’t likely in such a short time span. However, I do have ideas about how teams might increase in size with the release of more effective means of communication, but then I also think the teams might shrink because people will be able to produce software a lot more efficiently and won’t need as many people as today. Either way, teams will remain in the next 5 years, but the size is up for debate.

Software bugs:

As far as bugs in software, I believe that there will be less bugs than there are now because people will be caught up with the past known bugs and be able to fix them. As far as new bugs I believe that they will still exist, but not to the extent that they do now. There will be better software out there that will be able to detect bugs and maybe even prevent them, however it will still mostly fall on the software engineers to fix the bugs themselves.

Software’s impact on society and the planet:

In five years, the impact that software will have on society and the planet will be greater compared to how it is now. There are many ways that software helps today, but it will be to a bigger extent in the near future because there will be even more advanced technology that will be able to facilitate it. For example, I can expect that many stores will likely not need any physical workers stocking the shelves or checking people’s items out, it will all be automated. This is already present in places like Walmart with things like self-checkout, but this idea will be pushed even farther forward with the increase of Amazon grocery stores, where there are no workers present in the store at all. “The cashierless-store concept popularized by the Amazon Go store — which Jeff Bezos’ company has major plans to expand across the U.S. — allows customers to pick up all of the items they need and then simply walk out the door. Skipping the checkout entirely, shoppers are sent a receipt of their purchases through the Amazon Go app (a necessity to enter) after leaving the store” (CNBC). These ideas will only continue to thrive in the next 5 years, and I imagine that it would eventually become commonplace everywhere.

Software trends:

One of the next software trends that I see continuing in the next 5 years is the experimentation with artificial intelligence and machine learning and deploying them into the world. There are already many different products out there, but I can envision many more being developed as they begin to solve new problems that arise. As far as what is going to become prominent, I think many different coding languages and protocols will begin to merge with each other, and you will begin to see a way to produce software without having so many different moving parts like we do today. To elaborate, I think that things will be comparatively simpler in todays time than 5 years from now, just because there will be a lot more moving parts that have to work together, like the more advanced machine learning programs and the implementation of artificial intelligence in internet of things devices. This will lead towards a trend of the removal of human workers in favor of machines, but I believe that it will only begin to really take hold 5 years from now, and then expand exponentially after that.

Software engineering in 10 years.

Software Process:

In 10 years, the way that people produce software is going to be drastically different than it is now. After 5 years, it is understandable that some ways of doing things would stay relatively the same, but after 10 years it is not crazy to expect the entire system of software development to dramatically shift to a more effective system. I think that clients will have a very specific idea of what they want, as they will have been working with the software engineers for a much longer time. I also think that the clients will likely be familiar with how to actually produce software too, which will contribute to knowing what they want the engineers to make for them. As far as actual code being written, I am expecting it to be all written by AI and then pieced together by the actual software engineering team. There will likely be no need for an actual person to have to type anything out logically because a computer will be much better at doing that.

Software team composition and organization:

I don’t think there will be any large teams greater than 5 people actually contributing to software development unless it is an especially ambitious project because there will be no need for that amount of manpower. Most, if not all, of the project will be completely automated, leaving the typical software engineers out to dry. I do think though, that the people who are employed to produce software will have to be more qualified than ever before, because it will only be the best of the best at that point, and they will have to have ample experience in machine learning and artificial intelligence so they can work together with the machines. According to Forbes, “Currently, approximately 30% of all tasks are done by machines—and people do the rest” (Forbes). This figure shows just how important and prevalent automation is, and it will only increase in prevalence 10 years from now.

Software bugs:

In 10 years, it will be very unlikely that there are any bugs in software, due to the fact that computers are going to be writing most, if not all the code. In fact, the likelihood of any bugs being introduced will increase dramatically the more involved a human was with the process, and companies will begin to see this and shift towards software engineers that have the least amount of human interference. Humans will likely be the ones tasked in fixing the bugs, as they will be able to have a more natural approach to knowing what exactly is wrong and how to fix the bug, but the detection will be up to the machines, as they will be able to flag when something goes wrong much easier and faster than a human would.

Software’s impact on society and the planet:

As far as the impact on society and the planet, I believe that software will completely change the current physical landscape, just as it has over the past 10 years. After 10 years from now, the world will be irreversibly changed as we become increasingly reliant on technology in our lives. Everyone in the world will likely be more connected than we are today, and there will begin to be more global issues arising like climate change and overpopulation. Whether or not technology will be solving those issues, I don’t know, but I do think that technology is definitely contributing to it. I also think that there will be some semi-sentient beings that we will interact with on a daily basis, and then we will probably have many different technological modifications that we outfit our bodies with. I am expecting that the world will be a lot more reliant on technology, even more so than we are today. With the increase of these body modifications that enhance our physical, cognitive, and social aspects, society will become crutched on the idea that the human body is not good enough. From a societal standpoint, different classes will emerge, depending on how many body modifications they have and what they can afford to alter, and that is something we have never seen before in today’s world.

Software trends:

The only trend I can think of that I haven’t mentioned is the trend towards impossibly secure systems being put in place. Since everything in the next 10 years is going to be digitized, it makes sense that important things like government function, communication, and commerce will be online. With everything online, the systems in place will have to be extremely secure to withstand impending cyber-attacks that seek to bring down the entire web of interconnectedness that is the world. With globalization, many systems are going to either become centralized or decentralized, but both options need to be able to stand the test of time and the test of billions of potential users each day. With RSA today, we are beginning to see many systems adapt to the new landscape of security threats and protect themselves with strong encryption, but in 10 years I believe that RSA will be the minimum standard encryption for everything, assuming it hasn’t already been cracked with quantum computing or an algorithm that hasn’t been publicly announced, for fear of the collapse of society as we know it. Regardless of what the threat is, there will be lots of security protocols in place to make sure that everything runs smoothly, otherwise you can say goodbye to a future that involves any sort of technology that seeks to improve itself on the current versions that we use today.

Software engineering in 50 years.

Software Process:

In 50 years, the world is going to be unthinkably different than it is right now. The year that many people attribute to as far away is the year 2050, but in 50 years it will be 2071. The difference between even those years is more years than I am old, so it is fair to assume nothing may be the same since it is so distant in future. But for software, it will be even more different. Any aspect of software engineering that exists today, such as any sort of human involvement, will simply not exist. There will be completely new ways to produce code that we may not even be able to understand yet, and in ways that haven’t been invented yet. At that point, clients would be anyone who needs a piece of software, not just businesses, as it will be more common for everyone to be familiar with how software is used. As far as payments, I think that everything will become way cheaper since it will be much more common, and because of that, I think that everyone will be able to have access to ways that can build software.

Software team composition and organization:

After 50 years, official software production for companies will be performed by the best of the best, which I think will be forms of artificial intelligence or humans with implants that allow code generation and decision making to be much faster. Instead of a team, it will be up to a single person or machine to actually generate the code and interact with the client. From the design stage to the final product, it seems likely that there will be an expedited process that will allow only one person to work on projects, unless it is something that is very large and needs a lot of decision making for. Since everything will be automated and made much faster, there will be a large surplus of software or a constant stream of updates to software. Otherwise, the software developers or machines will have lots of time on their hands.

Software bugs:

The unfortunate occurrences of bugs in software will never escape us, in my opinion. The very idea of a bug is just a mistake in the software that causes unintended output, and the problem is that the idea of mistakes is what makes us human. And even if machines did 100% of the software development, there would still be errors sometimes because they were programmed and designed by humans, who are imperfect. And then, if there were robots that designed other robots, there would still be mistakes because let’s face it, everyone makes mistakes no matter how many precautions there are. The point is, we are not going to outgrow software bugs. According to Medium, “…the complete eradication of software bugs is impractical). But the ability to detect and fix bugs will continue to get better over the years, so even if there are bugs, it is unlikely that it will be run and cause harm or be exploited by people with malicious intent.

Software’s impact on society and the planet:

By this time, it should be fairly obvious to see that software would have irreversibly altered the course of history. The speeds at which new software will be produced will surpass any “Law” that is used to limit it, both theoretically and judicially. The ability to limit the human spirit of creation has never been achieved, and in 50 years, I don’t believe it will be able to succeed either. But as far as the planet is concerned, I think that it will become a cyberpunk wasteland full of overcrowded cities, pollution everywhere, just a real doom and gloom future. The issues of pollution are already being felt today with the loss of biodiversity, but after 50 years the physical effects like increased temperatures and rising sea levels will also be observed. Technology may be able to alleviate some of the problems, but there is no chance that it can fully prevent the drastic changes that are coming, and it will have a more adverse effect due to things like computer parts being thrown away instead of properly recycled. Another possibility that I envision happening is to use tech to only temporarily solve pollution problems instead of addressing the real problem, like today in China, where giving air filtering masks to breathe in dense air pollution instead of trying to get rid of the air pollution entirely is advised. As far as plastic pollution, “…this impact is increasingly compounding upon itself, meaning the effects of the next 50 years will far outweigh the impact of the last 50” (GoPure). Software may be able to fix these issues, however it will all depend on what’s prioritized, and I just don’t think that the planet will come first.

Software trends:

Software trends come and go, so I don’t know what will be popular in the years leading up to 2071, but I really do think that there will be a large presence of artificial intelligence and cyborgs and other far-fetched science fiction ideas that will begin to be realized. I also think that they will all be fully integrated with society and be highly functioning members who pay taxes, raise their own family, and live their own lives. Whether that will be a good thing or not for humans will eventually come into question as we become a minority on our own planet, but we will cross that bridge when we get there. Other trends that might be prominent in software engineering might be the fixation on solving problems, since that is the reason why software exists, as well as the process that goes about solving those problems, whatever shape it takes.

Conclusion.

Software engineering has completely changed the world in the past 50, 10, and even 5 years. So it comes to no surprise that it will change the world in the next 5, 10, and 50 years. The ways that software will be produced will change, the ways teams function, the bugs and how they are eradicated, the impact of software, and the trends. Not only will software affect people who are directly related to its production, but to the entire world as time passes. Even at the 5-year mark, software will have a profound impact on the world that we can’t even imagine yet, with technologies that haven’t even been developed. The beauty of software is that it is just a tool used by humans to use their creativity and problem-solving skills to create new and exciting things, and I don’t see that ever stopping in the future.

References.

1. https://copilot.github.com/

2. https://www.cnbc.com/2019/11/12/amazon-go-cashierless-store-of-the-future-has-some-new-competition.html

3. https://www.forbes.com/sites/jackkelly/2020/10/27/us-lost-over-60-million-jobs-now-robots-tech-and-artificial-intelligence-will-take-millions-more/?sh=3127dc5f1a52

4. https://medium.com/acmvit/the-future-of-software-bugs-4f8891184bdf

5. https://blog.gopurepod.com/sustainability/our-earth-in-50-years-how-plastic-pollution-will-impact-our-future/