true

Contents

Student Developer Community: The Hidden Context for Building and Growing Your Skills as a Student

Disclaimer

This ebook has been published as a result of intensive efforts to provide students who would like to work in the software industry the guide towards developer community programmes. It is possible to find some mistakes in the content. Any mistakes found can be emailed to the author via `olivermensah96@gmail.com`.

About the Author

Oliver Mensah is a Software Developer who is interested in building developer communities and sharing knowledge. As a community-driven Software developer, he organizes, speaks and volunteers for developer community programs.

About the Reviewer

Who is this ebook for?

To all the aspiring software engineers, data scientist, UX/UI designers or any students pursuing career in computing fields, this ebook is for you. You will understand the community aspect of software industry and why that is important in your career development.

Why this ebook?

As students preparing your way up into computing field, you might be totally be absorbed in getting the academic excellence. Which is really essential and the thought of getting to know what you wish you should have known before getting into the software industry might not come to your mind. For me there is one hidden context that you must start exploring now as a computing student. This hidden context is now core of the software industry and that is community. While in college I spent some good times in developer community programmes and it has been helpful. As a fresh graduate, integrating well in the industry might sometimes be difficult because you might have been much exposed to certain aspects of the software industry and developer community is a great way to get into that. Meeting all kinds of people - recruiters,software engineers, architects, etc. and through meaningful and constant interaction with them, you will get to know much about how you can do well in the software world. And I have found it easier to integrate well with other developers on my first software development role after college.

Most colleagues keep asking what makes it easier for me to be doing most of the things I do - technical writing, volunteering for tech events,hosting and sometimes speaking at some meetups and conferences. And it usually follow up with how do I balance the developer community related works with my day-to-day work as a developer. In most of our conversations, I have come to understand that these guys are actually desiring to get into developer community like attending or speaking at a meetup or conferehce but how they can get started is the problem. I started guiding them with my little experience and make invite them to our events. And one of them is even about starting a developer community around his favourite tool. That is what we hope to see. There is huge power of coming together as community to learn, explore and build great products or solutions. And I hope my colleagues will never regret taking those steps.

I believe my days in school while attending meetups, volunteering and leading developer community programs broaden my perspective on about the whole software industry. And to help other students to identify this earlier and own thier career journey into software industry, I took upon myself as a challenge to put together my tips and experiences and that of some other developer community colleagues into this single ebook. This is to serve as a guide and informational reference for students on various campuses in order to have a deeper sense of developer community and how that would help them in thier career.

Chapter 1

Introduction

As a student there's no substitute for hands-on experience, hence many college students pursuing technical careers with strong interest in building a career in computing fields have resorted to accelerating their career through alternative pathways. Internships, apprenticeship, bootcamps, volunteering, graduate software engineering rotational programs, etc. are just a few of such alternatives. Student Developer Communities on the other hand look to be immediate means for students to accelerate their learnings and careers.

Almost every tertiary institution offering academic programmes on computing fields will have at least a student who is aspiring to work in the software industry as a software developer, architect, designers, etc. Exposing yourself and other students to immerse and surround themselves in a community of like-minded individuals is a quicker way to own their learnings and career advancements. Fortunately, the software industry has a culture of openness and sharing.

So far, I have not seen any material that has put together the experiences of the author and other experienced individuals who are doing massive groundworks on developer community programs that seek to guide and empower students to fully explore developer community - where other people have the same dreams and goals as you. I therefore took this a challenge to address what is missing; By putting together my experience with developer community programmes from the time I was in school and now, a professional within the software industry as well as the experiences of other students who are leading developer community groups or students who are working in the software industry into this single ebook.

And instead of bringing together theoretical reference, I rather had a deeper self introspection and engagement with student developer/student developer community lead. That makes this ebook special for students becauase we are sharing what we have done as students in terms of building communities around our various campuses. And the main reason why this ebook has been designed for students is to create the intention that will spark curiosity, motivate and empower those who have keen interest in software development to unlock their potentials and career through student developer communities. Let's unpack what I mean to "unlock" their career. By "unlock" I mean joining a network of other students with common interest, there is the creation of a meaningful network of people who have the passion and pleasure to fuel their growth in software development, personality and career. And that is where the opportunities come in; much is accomplish by a group than what you can do as an individual.

It is not just that you are going to be doing this alone with all your resources. No and not all. Recently, a lot of software companies have been investing a lot in students to build real world skills by setting up programmes to keep them learning new technologies and methodologies, explore and build interest in various computing fields - mobile, web, data structures and algorithms, etc. For instance, initiatives like Developer Student Club from Google, Facebook Developer Circles, ForLoop Africa, Github Campus Experts, Salesforce Student Groups, Microsoft Student Partner and many others provide students, developers or individuals a platform to connect with each other as both individuals and as groups. Developer Communities is becoming core of the software industry and that is time for such students to think in terms of community. It will be essential for your growth in exploring and building expertise in your interest areas of software development. Just imagine a community of students taking action towards their careers not just to only seek to inspire, but also grow practical technical skills.

Chapter 2

Becoming and Surviving as a Student Developer in a Fast Paced Developer World

In my personal opinion which I stand to be corrected, it is not easy for a student to become a student developer. Academic workload coupled with working as software developer, here comes a lot of responsibilties. You have the responsibility to ace you academic work as well as software projects with strict deadlines. Usually these are the students who see software development beyond the walls of the classroom, delving directly into working on software projects. This is true for those who have the experience but if you are beginner you can tell what you are bringing to yourself. And student developers usually have that solid understanding of what software development is about and they are eager to explore software development as well as understand concepts one level deeper. To them, it is not enough to play around topics, they love to be learn and build applications that will be used in the markets or real world. They believe buillding such projects exposes them to gain the practical software development experience that they won't be taught in classrooms as well as develop other essential soft skills needed to excel in their software development careers. And that is true.

Simply put, students who have identified themselves as having the skills to work in the software industry and then deciding work while in school are considered as the student developers. Having the skill and you are schooling? This is a great decision you can ever make if you have demonstrated competencies in software engineering or development. Apart from gaining the skill, you gain extra money to cover some expenses. However this can be quite challenging in new world of technology where there is everything is getting fast paced. Thus new frameworks, libraries, operating systems, software methodologie, software architecturing principles are being released as never before. In this chapter, we will explore together on how students with demonstrated software development skills can become and survive as a student developer in this fast paced developer world. Technology is fast changing and mostly companies or individuals are building sofware projects with the latest technologies or advocating for using latest technologies in their software projects. And this means, you as a student will have to use such tool even if you are not used to it. And you dont have the potential to learn new technologies you might find yourself stranded. Let's explore together some of the few important points you can keep in mind which will help you avoid pitfalls and excel as student developer.

Creating balance between your software development career and academic work.

Well,your main goal of being in school is to gain your academic qualification, however, combining your academic journey with working as a software engineer, developer, etc. increases your workload as a student. Creating a balance between your academic workload and work is essential in an ever fast paced developer world. You might be thinking how would someone create such a balance? Well, since your decision to build software comes from the passion to grow your skills then balancing your tech career with school must be your priority. Depending on the type of work you will be much engaged with - freelancing, part-time role,etc., maintaining balance with school will be quite difficult. Technology is exciting and there is always an increased level of excitement when you see your product being used out there. So getting to create those products while studying might make you give much attention to your tech career than the academic workload. You might be thinking the academic work is of no need because you have the skills as a developer. Though you are building your skills through various projects but the end goal is to deliver a project or software product from a given task(s). Here are some of the possible ways you can adopt to balance tech career and school work;

1) Let your school time-table be your friend: *One of the best way to really balance your academic work with your software development role is to treat your time table as a friend. Creating such tighly coupled friendship with your school schedule means you are always checking for the time lapses and taking advantage of such. There is more you can do with these times lapses. Personally those were the times I had to research on certain a topic I will be writing about as a Guest Writer. It work tracking your time table intensively.*

2) Talk to your boss or team lead: *As a student developer, you have a shared purpose of a well established software engineer or developer that you will either worked with or supervise your works. Understanding that shared purpose will drive the need, not to forget about your development team in times of issues which can academic, personal or work. They are people who believe in your capabilities and know what it means when you are stuck at an*

issue. This is probably a core truism for software developers, you don't know it all. This means you should be ready to discuss your project's strengths and weaknesses with either the development team, company or the client. Hence instead of wasting all the time to figure out issues with your program, you can get in touch with the team for help or a colleague to help you out. Some companies in certain cases pairs your wih much experienced engineers who can mentor and oversee your work. They are people who cares about your development experiences and academic excellence. Obviously, getting help from those sources is a great measure to save much time for academic related task.

3) Working with your school: *Many schools have come to the realization that some of their students have the will to work on software projects to get the practical software development experience that is not taught in the classroom. Schools like Yale University, has started an initiative that provide the students the opportunity to work as a student developers. So instead of you working for a client or company which might with a lot of working hours, working with your school is a better option. You will get to build up your skills as you had wanted and also get ample time for academic related work. According to the Yale Student Developers website; "we also work to support the community of students interested in learning and practicing software development". To me the schools understand your academic schedule and hence the amount of time that you will be needed to work on a project will be less. For instance the Yale Student Developers are supposed to work for a minimum of 10 hours per week[^1] and that of University of Washington is 12 hours per week[^2]. For you to stay current or work with modern development tools, techniques and methodologies, they tend to partner with other agencies that would help you in achieving that. "In addition to our course offerings, we work with groups on-campus such as HackYale, Yale BootUp, and the YCC to host events such as the YCC App Challenge. We also hold open office hours (where students can drop in to get advice or just chat about building apps), and focus groups with ITS for students to provide feedback on ITS projects". Some schools even go the extend of building a coherent community for the student developers where they can come together as a community to "learn applicable, real-world skills, and marketable modern-day technologies, Work with teams of cool, motivated students like you, learn skills for your next job or startup, mentorship from professional developers and collaborate with a community of developers" .However, some schools might not have initiative like this. In cases where your school has not designed such initiative, you can adopt the first two approaches to balance school workload and your tech job*

For me making my school schedule my friend as well as constant communication with my supervisor helped me a lot on my technical writing role. I usually used the time-table lapses to research deeply about the topic I will be writing about or the changes I have to make to a code project. Gather some thoughts and outline them on GitHubGist. Through that platform, my supervisor always have the updated content and the last time I worked on the content. I don't hesitate to ask for assistance any time, I have issues. All these happened during the second semester of my final year in school- where I was working on my final year project with a lot of tasks but I was able to manage those hard times with work. I got learn more about certain technologies. Balancing academic and tech work is one of the ways to survive as a student developer. Furhtermore, collaborating with other student developers on campus will also help you to attain that survival in this fast paced developer world.

Collaborate with other student developers

Creating balance between your academic work and tech job is at the core of you being able to own up your work and school workload. And that is very essential. However, there is more to building technical skills while in college. And that boil down to community. Community in the sense that the tech world as we see today is evolving around communities of people in tech or tech products. And you cannot miss this as you journey into the software industry. The community has been seen as one of the awesome channel that developers can funnel their survival in the tech world. And that is why there are so many tech conferences, meetups happening in the tech world. To survive in this world of tech as a developer, you definitely need to be in the community. It is everything now. This translates equally to you as a student developer, you will need to be in a community where you feel welcomed and people are willing to invest much of their learnings in your career growth. campus is a great place to start as a student - easier to find like-minded people who are interesting in building real-world software projects.

Building the connection with these individuals to create a collaborative and supportive atmosphere is a great way to construct yourself to cope with workload and projects. These are people who can motivate and help you whenever you are stuck, it could be academic or tech work. Having a strong sense of community of like-minded students simply makes you feel belonging somewhere and to something. And, what does it really mean to be a part of a community? Basically, the tech world is more of collaboration and that ranges in many areas of software development. If your school has student developer communities, join them. If there is none, that does not mean you can't be part of any developer community. There are many solutions to this; You could be collaborating with others via online platforms like StackOverflow, Dev.to, etc. However, offline collaboration with other students will help foster the tech scene in your school and that is very beneficial in using tech to solve issues on campus or local communities.

Building and growing community is an interesting aspect of this ebook, which would be explored in later chapter. It starts by advocating for students to join you to build strong technical or developer community in your campus that is geared towards bringing students together for more collaborative or supportive work. Collaboration with other students will come in many ways. You can start with a fun way of hacking around algorithmic problems or hacking around a task you want to accomplish for a software project. And it is usually fun doing those kind of hacks with your colleagues. As this becomes repetitive action among you and your colleagues, it is easier to open it up for other students to join. Obviously there might be other students who have the skills and would love to hack problems with you. Transforming this into student developer community that brings people together will not only help you collaborate on projects but as a medium to develop essential skills that are very necessary when working in software development industry. Always remember that the people in various developer communities are some of the kindest, most excitable folks you can ever meet. They're the kind of people who look forward to helping you and through the community, you find other students who will challenge you, identify others who can also identify you with what you do as student and you can easily seek help from whenever possible. Two things achieved in here. Firstly, attaining the the mindset that If you come across a situation where you don't have what you are looking for, you will be that person who creates that change you want to see. Secondly, you have attained the power of influence since community is about inspiration, motivation and growth of its members.

Chapter 3

Student Developer Communities and Their Importance

Community is essential and at the very core of software industry today. Have you ever thought of it or realized the impact the individuals you meet can have on your journey and that of others? For me, these individuals are the ones get me motivated, inspired, embrace obstacles and uniqueness. How does that happen? In the software world, it takes a community to truly thrive. In this chapter, we are going to delve into some student developer community programs, why software companies are investing in such programs and then provide student centered, and holistic experiences that focus on why you need to start or join a developer community in school?

The rising rate of student developer relations at organizations.

I have talked pretty much about how student developers can survive by strengthening community and collaboration. Talking about student developer communities, when you take inside glimpse into the local tech and developer ecosystem, you will find out that most developer outreach programmes targeting students are backed by reputable software companies. And they have even created job roles filled with individuals to help in growing and sustaining tech communities across campuses. Truly, it is a big win for companies to make their technology available to students and reach them at their level.

What does this mean for a student or student developer like you? Companies adopting communities to build audiences around their technologies and software by providing students with free access also has significant impact on young students around campuses who are eager to join the tech sector. Such students are presented the opportunity to build up their skills. Truly, this is a two-way relationship for you. Below are some are some of the amazing developer programmes for students;

Name of Student Developer Community	Owned By	Location
Developer Student Clubs	Google	Global
Mozilla Campus Clubs	Mozilla	Global
Microsoft Student Partner	Microsoft	Global
ForLoop Africa	Individuals	Africa

I honestly do not know all or most of the student developer programmes in this world. I decided to research on that and fortunately, I found a good resource with compilation of resources for college students or working professionals on various hackathons happening in different part of the world or online, tech communities, student tech ambassadorial programs, and many others. This link really useful and thanks to Deepak Kumar for this awesome work. You(as a student developer) must into that. **NB:** Most of these student developer community programmes backed by companies require you to apply, you can find out from their websites the application that so you can apply to lead those communities that you are more interested in. In case in you need help any special help, feel free to reach out to me via my `email(olivermensah96@gmail.com) or twitter(@Oliver_Mensah).`

Why start or join a developer community in school?

If you had been reading from the start to this point, you might be already gotten the whole gist about developer community and why student must get into such programmes. Here, we will discover that community is way more than just collaboration and supportive work but a great medium that student for student to build a great career in technology sector. As there is more to just collaboration, meeting new people,learn best practices for building world-class apps from your fellow developers,etc. It tells it worth it sacrficing your personal time for developer community related stuff. You will also get to know that developer Community is becoming and now at the core part of the software industy and the new the frontier; the software industry has taken a huge step towards community-driven approach in publishing technologies or products. Earlier on, I shared with you the stories of my colleagues on why they wished they should have done that earlier why in school. Getting involved with developer community related programmes while in school you can elevate your profile in the developer communities around you. This makes it easier to transition when you are out of college. In all, getting involved in developer communities as a student gives you the right support for professional development, allows you build your network growth as well as serves as an avenue for personal development;

1) *Get support for professional development*

Student Developer Community programmes are giving college students the chance to develop their skills, craft their careers, network with potential employers and among themselves. As a student this come with the opportunity to keep advancing your career goals, learningd and completing new learning goals. With developer community programs, you have access to the right support and guidance to help the members achieve great things. And when you are entrusted to lead such groups, you are believed by the sponsors or initiators of the program that you are capable to help in developing, growing and highlighting the tech ecosystem in your school. You are now in the right support for your professional development. When I say professional development, what does it look like to developers? Thanks to the Stack Overflow 2017 developer survey, developers told their view on what it means to have professional growth and development in the software industry[ˆ3]. Base on that data, I have developed some insights on what "professional development" looks like to software developers. According to the survey continues learning in different forms, ability to work on personal or side projects as well as non-technical skill development embodies the professionalism in software industry. Let's take a deeper look into these and how getting involved in developer communities as student is true path for you develop professional software development skills;

`i. Continuous Learning in Different Forms`

There is always something new in the software industry- new framework, methodology,architecture, library, programming language, ect. Keeping up to date with these modern technologies and applying them to solve problems is very necessary. Even if you are not using them for development, you should be able to know about them, why they were developed and what kind of problems that they are solving. These makes you better at choosing tools for development. Staying relevant within the software industry, you have to keep learning and that is one of the key insights from the survey - the ability to continue learning is their preferred way to grow in their profession. This can include learning a new programming language or framework, develop a new skill, or even change their career path. With the importance of career development in mind, here are just a few things developers should look into in order to boost their continous learning:

- Coding workshops
- Programming seminars
- Online courses/certifications
- Tech conferences

- Meetups
- Books/magazines
- Necessary software or hardware
- Bringing in guest speakers

With the vast array of technology, language and platform choices available today, getting involved in programs like the above will build you up and survive you as a lifelong Learner. Most of the above programs will get you into lifelong learning are key events and programs happening every now and then within various developer communities. So being part of community developer in school has already giving you the exposure on what it takes to stay professional as a developer when you graduate from college. And this become a plus for you when you get into a company or startup that does not have fair idea on how they can build continous learning culture for their developers. Though you might be graduating fresh from college but having been exposed to these programs, you can give solid recommendations and back them with valid reasons on what they can adopt for their developers to stay as lifelong learners. For instance, I joined a startup to work with for my nationa service. For those who don't know about national service, it is a program for Ghanaian students who graduate from accredited tertiary institutions to do a one-year national service to the country and it is required under law. They have good software developers building their software solutions but due to my exposure to the developer community work, I was given a position to lead to software development team till I complete with my service. This was as a result of the community leadership skills and some recommendations, I made to ensure that the developers continue to enjoy learning new stuff while making the startup maintain their talents. There is so much you can bring on board to any institution you find yourself since it is not just about development but also the people part of the software development.

ii. The Ability to Work on Personal or Side Projects

Another way to ensure that a developer become professional is to get himself/herself involved in side projects. With this developers are getting the ability to work on new or exciting projects. If they are not working on something new, they are likely to get bored. Side project is one way to combat boredom and allow developers to learn new skills and grow in their career. This can include developers finding time to; contribute to open source projects, work on personal app, or do something outside of their typical job duties. Some companies allocate time to their tech teams to achieve these. However, if you find yourself in a company that do not have these to create a culture of learning, you can still work on side projects. Because as a student developer, you are working on academic path together with software engineering roles. That means you are already acquainted with what it means to strike balance between differnt work schedules.

iii. Non-Technical Skill Development

Developers who are looking to take on more managerial or leadership responsibilities such as growing into an Engineering Manager roles obviously needed to be trained to get into those areas. This can include opportunities such as bringing in a leadership coach, offering training for first-time managers, or mentorship programs. For those who aren't looking to grow into a Manager or Executive role, there are still many non-technical skills they could learn. This could include programs that help improve their soft skills, communication skills, project management skills, and more. Whether getting into managerial roles or not, developers need to invest into building their non-technical skills. Being a student developer and also getting involved in student developer community related tasks, you are exposed to vast array of opportunities to build those skills. These skills will come in handy while organizing, attending or volunteering for these tech events. Though you might not be expert to train much people in engineering manager roles but you have the fundamentals which is people management, communication and managing task through your contribution to student developer communities. It would be easier to be trained to get on those jobs than someone who has never tried that.

In all, getting involved in student developer community programmes prepare you into a professional journey into software engineering roles.

2) *Network growth*

Becoming a developer community-driven student means immersing yourself in student developer community. Moreover, a complete immersion into the student developer community is a great investment in broadening your network. Participating and engaging yourself in community programs expose you to new people. This presents a great opportunity to expand your network, exposes you to new way of thinking that might lift you higher. For instance, learning to code is hard and this is usually true for someone who is fresh to the programming world. One way to get prepared to learn to program is to get friends who can share your oath with. Student developer communities are great places to get them. So when you join any university and you have the passion to get into computing career then feel free to join any of the student developer communities. Through this, you're able to meet all these people who have done same things as you about doing. And a lot of them have certain challenges that they have already overcome so you get the shortcut to that as well. That,to me, is so valuable that you can leverage on to build your expertise.

3) Avenue for Personal Development

It is absolutely true that getting involved in the tech community allows you to develop, grow and hone skills that you would not otherwise find time to do. Even though you are building your professional skills and network, I think Personal Development is indirectly gained through these programs. For instance when you even start by doing something as simple as presenting a lightning talk at a meetup event, there are so much you can attain from this. Through this, you are learning how to communicate and present to an audience. You will also learn to respect the views, needs and rights of others, including people of different genders, ages and cultures to themselves. Although, the community helps to indirectly develop your personality, taking conscious effort to develop yourself is really import. Personal development can take different forms, but teaching is one of ways that you leveraging on the power of community as up and coming computer programmer, etc.

- Paying it forward: Leveraging on the power of community to learn by sharing knowledge.

The joy of getting into developer community started at a campus program where clubs on campus host an open day program to tell more about what they do. Upon hearing about Google Developer Group, their mission and what they have accomplished since its inception on campus, I signed up to be a member. I was always attending the meetups because our seniors were inspiring us not to give up even if we are not finding it easy to program. After a while, another batch came to take over the leadership role and it was not as effective as before. Program will be fixed for certain date and when you attend the meetup, you will find just few people. Later, I started to connect with more developers outside my school, and ended up attending meetups in the Accra township organized by DevCongress. I loved that meetup, learned a lot from the speakers. The developer community on campus was dissolved and we later tried to get it reinstated. We got it back with a lot efforts.

I had a lot of interests with the developer communities so I later decided to get into leading any of the groups for learning experience. I got accepted into the Facebook Developer Cirlce program and it has been great working with other developers to build the tech scene in Accra. That's where I decided to share with the world what I have been learning. I started with giving lighting talks. Initially, I was not good enough presenting in front of people, but things got better along the way. Along the way, I started technical writing on Dev.to platform - an online platform where programmers share ideas and help each other grow. Starting was not easy but I managed to write some articles.
The feedback got me to write more because I got to learn more from the feeback and I got better in writing about technical topics. It is through this, I got to work with Auth0 as a Guest Author. You might not have heard about Auth0 but to simply put it, it is an Identity as a Service(IDaaS) startup based in the US that provides authentication and authorization as a service. Through their services, you as a developer have been giving the tools to secure your applications without having to become security experts. And I was paid per article. It has been an interesting journey so far.

You might already know that the best way to learn something is to teach it to someone else. And Student Developer Community is designed to create value for you, both leaders and members. However, it starts with you taking control of your career or personal life by taking advantage of the opportunities and resources. You might not know whether you will be the next engineer documenting APIs for others to use, or something else. You can get started through blogging, technical talks, etc. and the community is so amazing that you are surrounded by thousands of lovely people who can review your works and improve along the way. Honestly, sharing knowledge through either writing or public speaking worth it. For instance, technical writing or speaking at tech events build your communication skill and being able to communicate well is one of the most important skills a software practitioner can have.

Personally, It was very difficult for me, I'm a very shy person and speaking in front of people seemed something that I could never do and was not even good enough to write on technical stuff. But heading a step out of my comfort zone through community programs has made me better and gained a lot of benefits, and this is one of the things that I am most proud of. So as a student, you should take advantage of these student programs to help shape our personal lives and careers. Let's explore how you can get into sharing your knowledge as a means of learning and reinforcing your knowledge;

i).Just Getting started.

When it comes to learning and development, knowledge sharing must be a developer's best friend. Our industry has always had knowledge sharing as part of its culture. The community want to make everyone else's work better since it is better for the entire ecosystem. This makes it full of the nicest and most amazing people you can ever meet and work with. The culture of sharing, educating, and giving away our work is incredible within the software industry. The only to get started in knowledge sharing culture is just get started. Though it might not be easy for the first time but even if you have opinionated point of view on the topic you are sharing, you will have people who will be nice to correct you even though there might be a lot of "shit" responses. No matter how, you will need to prepare to make your audience understand what you have for them. In order to know who you are users are, you have to gather as much information. It is important to know if your audience holds expertise in the field, if the topic is totally new to them, or if they fall somewhere in between. The audience will also have their own expectations and needs. Understanding your users is very key to delivering good materials.

ii). Mentorship

As you prepare to follow the "share and gain" principle which is basically about teaching what you learn or know as a means to reinforce your knowledge and help others who are learning around your knowledge domain. To me there is no better means to guide you into this area of sharing your knowledge to wide range of audience by publishing articles,speaking at conferences, etc. than being mentored by someone. Whatever you want to do, someone has done before or is doing that. They are experienced and trusted individuals who can build you up in such area. You could be bothering how do you find yourself a mentor? For me. the best advice I can give you is to just ask. You can start looking for people with expertise in these areas from you community and if you could not get anyone, then you do some online searches. You will obviously get someone who you think can mentor you. You can directly reached out to them to mentor you or read from how they write their works or speaks at events. When I started, I learned from different people who were speaking and writing technical blogs through their blogs and videos. Later gained a mentor from Auth0 who reviews my work and how best I can write technical articles as I started writing for them. Sometimes, some of the people you wanted them to mentor you might be busy with work and other stuff, so their response to you might be no. Do not let that stop you from asking. You keep on with it and you will surely get someone who will be ready to help you out.

4) Empowering and growing a local community of developers.

There is nothing more fulfiempoling than investing in someone's growth. Community is a great way to achieve that. The most empowering thing for developers is being able to create something. With community, starting programmes like hackathon, codign challenge to address community based problems is a way where developers can help build something useful for local communities.

The need for education-friendly policies to support growing student developer communities.

Student who are helping in building technology communities in various campuses should thinking beyond just buiding a community.They must strife for established well-sustained programmes that will help the community to have strong ties. Building a strong empirical partnership for the effectiveness of such programs is the key to its sustainability. As companies continues to support students to champion their technologies and software, this where schools must be precise on creating a long lasting bonds with such companies to ensure the continuity and sustainability of student developer programmes. Strong collaborations between the hosting schools and companies can help bridge the gap between students, and professionals and make the student industry-ready.

Through increased investments and partnerships, student tech communities can transform the employment rate among students upon graduation. With student developer communities, companies make sure that the students learn best practices of their technologies and tools, helping students build projects through hackathons and code competition. While students continue to go through activities designed by the student developer managers, they would eventually build their capacities to get themselves ready for the market. However, implementation of tech-friendly policies like free or less-expensive broadband access and active support from the school authorities will be critical in making in growing student developer communities in schools. I believe this will help very much in boosting the number of students who get jobs upon graduation. We know schools boast of these numbers. Partnering with institutions that invest more into student developer groups can help increase the rate of student employability upon graduation. According to Yale Student Developer website for instance, they have partnerships with such institutions and through that some of these student developers have secured jobs in at least one of the biggest software companies.

Starting another student developer community amidst existing communities.

In some instances, you might have joined other developer communities on campus. However, if the existing ones are not serving the kind of experience you want to gain, you can provide them with measures to solve the gap you have seen. And you still think they are not serving the purpose of gaining technical experience, there is nothing wrong in setting up a new student developer community. But before you do that, here are some of the tips you can follow in order to successfully create another tech community that can co-exist peacefully with others;

1) Describe the existing technical student communities on your campus. Who is served? Where are there gaps?
2) Choose one student developer community and how could that support your on-campus student developer communities?
3) What would you need to do in order to support the growth of student tech communities on campus?

Student Development: Some student developers' and Community Leads' Personal Reflections.

To make this ebook relevant as well as find it useful in starting student developer community in your school, I interacted with five student developers and community leads to share their experiences- why they got into student developer community programs, challenges, success stories and why would they recommend others to be part of such programs; as well as tips on how to students can start and run successful developer communities while they are in school.

1) Wayne Jotham Gakuo

Wayne is a senior at Ashesi University, Ghana and the 2018-2019 lead for the Developer Student Club - Ashesi University. According to him, he got into community developer program because of his passionate about sharing the tech-knowledge with other students

- What is your personal motivation for starting student developer community?

I am passionate about sharing the tech-knowledge I have with other students. With an ever-changing society, communities, especially the marginalized ones, tend to be left behind in terms of development. I believe that teaching communities these skills empower them to be critical thinkers and problem solvers whose main aim are not to get paid for it but rather solve real-world challenges that affect *most members of a community. Technology is a huge driving factor in community impact; it can either make or break a country in terms of development*

- Has there been impacts on the members of the community?

I believe it's important for students, especially those in the university, gain mobile and web development skills which will give them an edge of their counterparts when it comes to employability. Even if one does not plan on getting employed, he/she will be able to work at their comfort of their houses while undertaking clients' projects. These skills will also enable them to build tech-based solutions for communities in a society where the marginalized are looked down upon when it comes to development.

- Being a student and community lead, what kind of challenges do you face?

As a students, I am figuring out on how to balance my school work and community involvement, trying to make sure that I give them all equal attention.

- What achievement(s) are you most proud of?

I have been able to introduce over 70 Ashesi University students into the tech field; they came in knowing nothing about cloud computing or web development. As a developer community leader I have also been exposed to new technologies which I am always eager to share with the rest of the community which helps them keep abreast with an ever-changing technological world.

- Would you continue to be part of Developer Community after you graduate from college? Tell us more why you are opting for either yes or no?

Yes. I would love to continue being part of the Developer Community because I believe there are more out there who need to be taught these skills which enable them to be world problem solvers, with the focus of their communities. It is my belief that local communities are the back-bone of a country and empowering community members goes a long way into uplifting members from their current economic statuses.

- Would you recommend Developer Community to any student and Why would you?

Yes. A Developer Community will teach you more than what you may learn in class. Having skills can go a long way into becoming a visionary who always seeks to empower others to do more than just sit and watch!

I personally know Wayne from school, Ashesi University. He has done incredibly well to help build the tech ecosystem around the campus. During our time where we did our best to get the developer group reinstated after its cancellation by the Developer Community lead at Google. Thanks to Anthony Abeo and Vladmir Fomene. We later got to know about the Developer Student Club programme. When we talked to Wayne to apply for that role so he can take over from there, he was not fully decided and he applied about few minutes for the application to be closed. And it amazing to see his good works for developer community.

2. Sai Yerni Akhil Madabattula

Sai is a student of Sathyabama Institute Of Science and Technology. He is one of the organizers of the Sathyabama Coding Club, a student developer club that aims to bring together the core talent of the Institute under one umbrella.They envision to improve the art of competitive programming skills in the students and to make them ready for renowned competitions like ACM-ICPC, Google code jam, various Hackathons conducted at national and International levels etc. According too Sai, being part of this communit is to empower learners and learn others to get ready for these competitive programming contests. Getting mentorships from industry people is the biggest thing we receive being a part of Student Developer Community, he added. For him, designing programs to attract large strength of students is their challenge while running this student developer community program. However, he sees great things to happen with their community as more students get access to internships and getting appreciation from other parts of the world, they can leverage on that to build a better student developer community.

Chapter 4

Building Long Lasting Developer Communities with High Engagement

A community is about learning, sharing, teaching or mentoring. But it is more than that, but a way to revamp the energy and inspiration to keep it active and growing. Hence active communities are essential to building and growing student developer programs. You may not be fond of community work so you might be thinking of how do you go from an individual student developer to a full-fledged developer community as well as make it active? When you are running any program that is meant to engage people, knowing your audience is key. Understanding the people you are targeting with your program and the individual activities within it can help you design events that appeal specifically to that audience, generating higher levels of engagement. This is where positioning comes in. By positioning, I mean how are you identified as a student by other students or faculties during interactions? If you like to interact with other students and faculties, it would be easier for them to understand where you are coming from with this program and be convinced to join you in creating the community you would love to see on campus. Maybe you have gained that recognition on campus to drive a lot people into the community. Don't be afraid to start. As students love to build up their knowledge, leveraging in your skills or that of your colleagues who might be able to at least break things down into easily digestible bits that others can understand will help achieve that goal. In order to build a highly engaging, experienced and mature community, I have put together the following tips;

1) Promoting diversity and inclusion in the community

Many has been said about the tech world when it comes to diversity and inclusion. Mostly described as male dominated area and that has made it uncomfortable with women. I have not experience that yet. To prevent the community you expect to build, you have to make it more inclusive and diversified in terms of gender, content, age and other factors. Here are some of the steps you can take to show support for community diversity;

`i) Open the Doors to More Female Participation`

To ensure more female participation, provide them the access and opportunity to advance their careers in tech, pursue their dreams. This will get them the drive to inspire other females to join your community. Opportunities like allowing them to hold leadership positions, speaking or hosting tech events and many others. Design programs that will bring them together, mentorship for their needs and growth. Doing so will help reach the gender parity in technology fields.

`ii) Creating more gender allies`

Though you might create opportunities to ensure female participation. However, if the community is not welcoming, then their involvement in the community will be less. This is where having male who can take action and drive impact in making them involved and feel welcome is essential. And this is why you should encourage the males to get involved and be true allies to females in tech.

2) Reward System

Students or Developers joins the community of the passion, innovation and the love for diverse group of people with the ultimate growth hack for products and services. No matter their passion, by rewarding members, you can begin to construct involvement and engagement funnel. Hence, accelerate engaging a community of student developers by making them feel welcome and most importantly, thank contributors when they help you. Recognizing each contributor's input in a variety of ways. Easy things to do that go a long way are shout-outs, a kind note, or sending some swag, like a T-shirt, stickers, well-branded attires, etc. Or you can do better at swags and incentives by looking for sponsors who can provide students at discounted or free online courses, workshops, etc.

3) Working with your college/university Authorities

Leadership around your community is very essential and that is what glues the community together and focused. Getting school authorities that have passion for technology and community is the best way you can build great leadership around the community. Every effort should be made to let them keep the community in check, it terms of growth, collaboration, engagement, diversity and inclusion.

4) Constant iteration through feedbacks

Getting feedback from your members helps you to interact faster. The more you engage them for feedbacks, the stronger sense of community is created.

5) Engage audience with consistent events.

I think one of things you can do to grow and build lasting tech community on campus is reaching wider audience with consistents events. You can design and organize workshops tailored for your audience, thus knowing your users can help in whether to split between intro courses designed for non-programming majors and more advanced classes for special interests and people with some prior training. Though events are created to bring people together as means of building stronger sense of community. Also programs like hackathons, community challenges and other programs that can help them earn some amount of money might be appreciate. Do you best to include those programs as well.

Chapter 5

Common challenges while running developer community programs and how to overcome them

Building community comes with plenty advantages as members continue to support each other, learn from one another and from a group of great speakers. Generally making this an open, vibrant, and thriving community will help you achieve fantastic experiences over time. However, the benefits, there are some issues that can be considered common challenges and no immediate measures are taken, it can affect the growth and sustainability of the program. In that case, I will discuss the potential challenges that I have seen with most tech events and how they can be overcome.

1) Lack of Awareness: *If your community is not well know on your campus, it will create additional challenges for you as the organizer. In order to create more awareness about your program, you can get your school write about your developer community, its existence and how it hopes to build the tech scene on campus and help supplement students growth with what they learn from various classes. This can get others inspired and make curious effort to attend and be part of your community*

2) Lack of resource personnel: *The type of resource person invited to your programs has influence on how many people will react towards your event. This is something that I have realised with the events we have organised. When you get resource persons who are well-known in the tech community usually has high conversion rates for your events. Lot of people attend programs like that. In this case, do your very best to invite lecturers, developer advocates, software engineers whom you have seen their works with greater influence to your programs*

3) skepticism of your works: *Of course, there will be naysayers. Of all the energy and enthusiasm you might have, there would be a lot of skeptics from people who has no interest in developer community program and that will permeate you attention on the work. They see that as waste of time and energy. Let me tell you one thing, most students who get into developer community in their early days in school might have experience it. I experienced it. But what kept me going was the purpose I had when it comes to developer community programs. I did not want to end that just in school.I saw campus was the right place to start, familiarize myself with technical speaking and by the time I graduate I have the skills to be able to deliver at any tech conference I wish to attend. Truly, it will take your time and energy but with your purpose in mind, those skeptics cannot permeate your work*

There could be other challenges, but I will restrict to only these. One important tip I can also share with you is that you are not alone in this software development community. There are beautiful souls out there that are willing to help you in case any challenge emerges. Share these challenges with your point of contact, colleagues or co-leads or co-organizers and you will be fine.

Conclusion

The software development community takes a bit of nerdiness and studiousness as you have to constantly be a student and learning to keep up with the industry. And that is why developer communities have been so keen to my heart. It has proven to be useful avenue for developers to learn and share together and I do appreciate how tech communities help developers like me to grow and evolve. There is much support among each other, rather than by competing against each other. For me, becoming a software professional is much more than simply knowing how to code these days. There is so much you have to know because you will deal with people, understand how software is best built from a high-level, best practices and trade-offs between them,etc. There is so much. One means to get you, students, prepared for these software careers is to think in terms of developer community .

You will always remember the feeling of getting your first code to run smoothly after hours of confusion. But you did not just get it run, there were menagerie of mistakes, pain points, might have giving up for hours and later come back to continue. So it does when you start running developer communities for students to come together and create impacts with through technology. There are so much that student developer community can teach you. It will take a lot of time and sacrifice as a student. You might give up if you just see developer communities as just an ordinary community. Then rethinking about developer community as an “apprenticeship” phase of your career journey to build your capacity and grow your skillsets is must. This will require deliberate, earnest and open-minded exploration that most people cannot enjoy.

And now that I've provided my context for the "student developer community" discussion, let me start by saying to those of you who would like to get into these programs: my advice to you is to ignore naysayers - peopel who will be saying all sorts of negative statements about you. You could be describe as incompetent to even start or lead a team . Bear in mind that it takes time for you to be quite good in whatever you do in life. Ignore them and go for it and you will later get know the actual values you will be adding your personality, skillsets and many domains of life.

It there is one thing that you will take away from this ebook, then there is no such time as time wasting for student developer community programs. Have the opinion that you're either apprenticing towards a lifestyle you want or you're apprenticing to position yourself for a future opportunity. Neither choice is wrong or right, just decide for yourself which one allows you to grow in your Technical and Team skills, be good at goals setting and squashing them within the set period. and finally become a software professional.

I hope you find this helpful. Keep being amazing students and continue to leverage on the power of community as your journey you way into software industry. Let's keep in touch, you can follow me social media; Oliver Mensah on Facebook(olivermensah.me), Instagram(@olyvamensah) and Twitter(@Oliver_Mensah).

References

[ˆ1]IT at Yale

[ˆ2] Student Developer | University of Washington Information Technology [ˆ3] What Does "Professional Development" Look Like to Developers?