Transcript of Interview with P3

Speaker 1 00:00
Hi, I’m [redacted]. I live in Hong Kong. I'm a software engineer consultants, open source developer uh, and I’ve been a google summer of code mentor many times over the past several years, most recently this year.

Speaker 2 00:19
Yeah, I noticed that you have 7 years experiences.

Speaker 1 00:27
It's I took a break there, so I think I did the first 6 years or 5 years of summer of codes. And then I had a couple of years of breaks, and then I have one more mentee. And then I took a break again. And then this year I had mentee again, yes.

Speaker 2 00:49
it's a quite long history.

Speaker 1 00:55
Yes.

Speaker 2 00:58
First, I will introduce our semi-interview. In semi this interview. I will show your results as well as the overall results of the online survey. I hope you can tell us why you choose these options and your comments on the overall results uh. I hope you can feel free to talk anything you like, because all the data we collected in this interview will only be used in this study, and we will keep anonymous. So I hope you can tell your personal experience.
The first part is about the motivation, is about why did you participate in GSoC as a mentor? Could you explain your answer?

Speaker 1 01:57
Mostly because it's fun. I just enjoy teaching and I enjoy mentoring. And I like working with smart people on interesting projects. This summer, I had one students who had a very interesting projects and I said, count me in, I’m to help you out.
So it's purely for my enjoyments. And I like yeah, I like working with young, smart people. So the graphs, while that, what I’m looking at here. So my answers are the little circles, right?

Speaker 2 02:52
no, the left column is your answer.

Speaker 1 02:57
Yes.
Now I see, I was looking at the wrong side of the slides, okay?

Speaker 2 03:06
Yeah and, the rightmost column is the average result.

Speaker 1 03:11
right? I that's what I thought, okay? It's interesting to see what other people answer.

Speaker 2 03:27
I noticed that you strongly disagree with a career building.

Speaker 1 03:31
Is it because you already have a lot of experience?

Speaker 1 03:40
Yeah, I've been doing this for 20 + years. I think my career is a reasonable shape. I'm definitely not in it for the money as the site, and I strongly disagree.

Speaker 2 03:57
So what's your basic motivation?

Speaker 1 04:01
Fun? I like working with smart young people. And if I can give them a notch to set them on their way, it feels good.

Speaker 2 04:14
I think, is the intrinsic motivation. I noticed that you strongly disagree with the peer recognition, the reputation in the community um.

Speaker 1 04:30
My reputation does not matter either way uh. I think my peer group could not care less whether I mentor or not.

So it's not that I strongly disagree that this affects my peer recognition. It's just it's not important to me. It's not a reason to participate in this project.

Speaker 2 05:03
You strongly agree with the sustain of the project. Do you think just all they can help attract more newcomers to your projects?

Speaker 1 05:14
Yes, uh and, I think that's the key reason to have this project at all. It's important for open source projects to continuously attract fresh, young talents. There is that in my projects, in particular, the average age is drifting up. Most of our developers are in our 40s and late 30s and even older in 50s and 60s. We need to get people in there in their 20s involved or eventually we will die so.

Speaker 2 06:01
May I know do the newcomers still give some contributions in the project after GSoC.

Speaker 1 06:14
The good ones do, not all of them.
So when GSoC started, whenever it was in the stone age, one of my first mentees went on to contribute to participate in summer of code twice, I think. And after that, she just kept contributing, and she is still contributing today. So she's working for a company. And in her spare time, she's still contributing. I have had other mentees。

Speaker 1 06:53
disappear over time. So while they're still at university, or as they just get started, as young employees, they keep participating and contributing. But as time passes, they drift away. I don't have exact numbers on my mentees, but I think I have about 50 % success rates in keeping my mentees is engaged.
I don't think I can claim any credit for that is just a personal thing. Some students or some people like working on open source that they keep doing it for the rest of their lives uh and others don't. And the value of GSoC is that it gets people involved in the first place. Yeah. If they can get some money out of it, that's fine with me, anything to get them.

Speaker 2 07:43
Another question, I think the intrinsic motivation is very important for a mentor, but I heard that for some projects very difficult to have enough mentors. Do you have some suggestions? What do these projects they can do to attract more mentors?

Speaker 1 08:12
That's a very difficult question. I think. The problem is the most experienced developers are also very busy people making time to mentor a student is an investment in time. And particularly in my communities, there's a strong feeling that if you don't have time to do something properly, you're just better not to do it at all.
So I think I'm not sure if there's an easy solution, a solution to the problem is simply to have more developers grow into and become experienced. And they will then pay the favor forwards. And that's, ii think one of my mentees is from years ago was actually a mentor himself this year. So ii think part of the solution is to have more mentees, and they then go on to become mentors themselves.

Speaker 2 09:24
but it's also very difficult to attract many developers. Uh let's move to the next part is about the challenges for the organization application. I noticed you haven't participate in an organization application, why? Because you have participate more than seven editions of GSoC, why don't you didn't do it?

Speaker 1 10:09
I don't need to. So there, I already do a lot of administrative backgrounds, activities for the [redacted] project. I play systems administrator. I play postmaster, I do a lot of things. I don't really need the extra work filling in forms and chasing people to write reports.
I did the management aspects of summer of codes does not appeal to me. Having said that if at some points, GSoC happens, and there is nobody to do the org application for the freebies. The project. I would probably volunteer to do it. But so far, somebody else has always done it.

Speaker 2 11:11
Yeah. For the organization application, is it because most of the mentor they don't participate in this application. And there are some people who will do this. They are not mentors.

Speaker 1 11:33
I don't actually know if the people who did our application were also mentors this year. I think they were or at least they have been in the past.
So for the past, several years it's always been the same person who has taking the or application and chase the mentors. And as long as they volunteer and are happy. That's good.

Speaker 2 12:05
As far as I know, for most large organization, they have different projects. And there is one special person who will help apply for the organization, but the person is will not be a mentor.

Speaker 1 12:28
I think in our case, I'm just looking it up because. I have this data. In our case, the or application is done by a team, three people or at least three people get the email. And I think they have all been mentors in the past.

Speaker 2 12:56
If this organization they have the experience for the GSoC, do you think it is much easier for their success application?

Speaker 1 13:08
Probably, I think if you are familiar with the system, you are going to find it easier to submit an application.
Also, it is worth remembering that my specific project is a very large open source project, and we have participated since the beginning.

If imagine you are on the google side of things, you receive an application from the usual person at the usual projects, it's probably going to be accepted for us. For the freebies, the project, the org application is pretty much an exercise in bureaucracy and filling in the right form and sending it in rather than having to face an uphill struggle and fight for funding.

Speaker 2 14:02
May I know the name of your organization?

Speaker 1 14:05
We are the [redacted] project.

Speaker 2 14:07
It's very famous. Let's move to the next part. The next the part is about the challenges during the student selection and a community bonding period. Could you explain your answers?

Speaker 1 14:28
Yes. So for this year, I was very lucky. My mentee was in the same time zone as me, and always very nice. So we just chatted on signal and we have a call every week. So that works out really well.
Also, I was mentoring the students together with another mentor. Who was, I think, one of their professors or teaching assistants, so they already knew the students. So it was really easy to get started on mentoring this time.

Speaker 2 15:16
Yes, does your project have a formal procedure for introducing the community?

Speaker 1 15:28
Not really, we have lots and lots of mailing lists and IRC channels. And we encourage mentees to join those channels and communicate as much as they can. But there's no formal process. And also we realize that every mentee is going to be difference. Some of them really like sending status, reports, emails and do the whole bailing this discussion thing. Really, whereas others prefer a phone call or sending text messages. We just take whatever works best for both the mentors and the mentees. It does work best if the mentor and the mentee are in the same time zone, or at least not on the other side of the world.

Speaker 2 16:23
So I think why you disagree with this challenge, just because you think it is not necessary to have a formal procedure to produce the community?

Speaker 1 16:38
No, I don't think it's I in general, formality is not necessary. I think it is just do what works best.

Speaker 2 16:53
Yeah, I think especially for the [redacted] such a big community. So each project may be, they have their certain characteristics.

Speaker 1 17:05
It's also it presumably also depends on the number of new participants you have with the [redacted] project, because we are an operating system uh, which is automatically limiting the set of people who are interested. We only get a handful of mentees.
So it's easier to workouts a personal personalized plan for them to get them in. If you're a very large open source project with hundreds or dozens of new contributors, every time you need a process, if you're a community like the freebies the community, where we have maybe six or seven or ten. I don't remember how many students we have. Students, it's easier to just work something else, 12345 students this year. So no need for a process.

Speaker 2 18:10
The I noticed that you agreed with too many or too few applicants to select。

Speaker 1 18:20
there's always too few

Speaker 2 18:23
what I think your project is very famous.

Speaker 1 18:28
Yes, but nobody wants to work on operating systems. It's and I don't know why I think they are fun, but I。

Speaker 2 18:38
Is it because the project is difficult?

Speaker 1 18:42
It's not difficult. There is a lot of codes and a lot of history. You have to want to work on operating systems to get involved. And that limits the number of people. So I think in our biggest summer of code year, we might have had 15 students, but only about half of them succeeded.
So in in in, recent years, we've been We've been limiting the number of students to what we can well to the ones we think will succeed. And to a level, we can mentor effectively.

Speaker 2 19:28
Do you have some good suggestions to for the last challenge?

Speaker 1 19:40
I don't know in a perfect world universities with teach operating systems, a as part of the computer science degree, but that's gone out of fashion. Most university computer science curricula don't really do operating systems for undergraduates, other than on a purely practical level, many universities still

teach operating systems in their masters programs, or obviously in their research directions, but undergraduates generally are not really taught operating systems, did this. There is no, but to be honest, there is no needs to learn operating systems.

Speaker 2 20:27
Yeah. In my university, I also think the subject of the of operating system is not good. The students can, I think it's a little difficult. So maybe the teachers should design a good subject it's.

Speaker 1 20:54
It's difficult to teach because operating systems become bigger and more complicated. In the 19 eighties and, 1990s, you could read through the entire source code of an operating system in a weekends or maybe a couple of days. If you want to go and read through an operating system in 2021, it's a semester of your time just reading the high level architecture. So it's difficult to teach and a there there's, a lot of areas you can concentrate on.

Speaker 2 21:35
I want to know why do you contribute to [redacted] ? I know you are a volunteer.

Speaker 1 21:49
I do it because I've always done it at this stage. I got involved with the 3 years the projects about 20 years ago. And I have always just enjoyed working on previous the many of my customers, my clients use previous them if I fix things for them, I can contribute them upstream.

I don't get paid specifically to work on open source 3 years that I do often get paid to submits work. I have done professionally uh two to three [redacted] projects. I just enjoy it. It's fun.

Speaker 2 22:40
Okay? Uh let's move to the next part is about the challenges during the official execution. Could you explain your answers?

Speaker 1 22:57
So it's fine. I could read it. Uh uh. There we go. Challenges, doing coaching. I agree. That's just saying that's fine. Yeah, I as I said, this year, I was very lucky with our mentee. He was very, very self-directed and self-managing in the same time zone as me.

So what we did this here is we just Thursdays had a signal group and we just kept in touch as he was working. Every week we have a quick phone call on Thursdays. We just got along and. There were really no challenges. It worked very well. In the past as well. I've never really had difficult mentees. The ones that failed, it was obvious that they would fail after about a week. They just disappeared and never came back, and the ones that the ones that stayed on were a pleasure to work with and easy to communicate with.
So, yeah, some of these are yeah I strongly disagree with the harsh project. At this year. My like to think the previous project is very friendly. Where else was I strong here? Cognitive differences? Yeah, you learn to, you learn to learn work with people.

Speaker 2 24:44
I think uh because you have a lot of experience, the students they have not such experience. So maybe there is an important cognitive difference between the experts and new commerce.

Speaker 1 24:59
Yeah, I think it's probably importance to reassure the students that meant is that the that they should not be afraid of you or that that you don't bite. So I apparently have no problem with this. I think listening to the students and listening to what they're struggling with is importance and just setting them on their way that just works.
So I think part of it is experience. I think if you have 20 years of experience teaching people and doing consultancy, it's very easy to get along with students. Whereas if you, I don't know other people might struggle with this more.

Speaker 2 25:55
You are neutral about ensuring that mentees finish their work. Do you think it's difficult?

Speaker 1 26:09
Yes, I don't think it's difficult because it's not really important that they finish their works. I think it's more important that they are adjusts there or they manage the projects expectations. If I have never met a mentor who sets the project plan and sticks to it down to the last minutes or milestone. But the best mentors are the ones that keep an eye on what their work is, and can adjust their work in function of what they're actually getting done. Especially in operating systems, a project is not easy to clearly define. So the goals will change as the projects progresses. It's not too important that they finish their work, but it's important that they manage the expectations of what work is getting done. If that makes sense.

Speaker 2 27:17
Do you think the mentees in your project they are proactive?

Speaker 1 27:25
The ones I have worked with definitely have been. We part of that is probably because of the way we select our mentees. We are automatically drawn to people who are self-managing and self-managing people tend to be more proactive. In earlier years, when we had 15, 20 students, some of them were less proactive, and those were the ones who failed. In recent years, we've gotten better at picking students who will not fail. That's automatically selects for more productive people.

Speaker 2 28:09
What signals do you use? If a student they are proactive?

Speaker 1 28:17
If they ask questions, I like people who ask questions and someone who asks a lot of questions, especially difficult technical questions and open ended questions, is much more likely to be proactive than someone who goes into the corner quietly. And only send us status reports.

Asking questions is a key indicator of a productivity.

Speaker 2 28:54
Could you give some comments on the cultural difference and the difficulty in creating an inclusive community?

Speaker 1 29:07
It's difficult to comment on uh because I get along with everyone and I like this. I live here. So I’m accustomed to cultural differences and I do it automatically. I know my mentee this year was what was Chinese? I know how Chinese culture works because I’ve done. I live here, but in the past, I have had mentors from other cultures and I just get along with them. I personally have no problem with this. Okay.

Speaker 2 29:54
Could you explain the difficulty in setting milestone for the maintenance?

Speaker 1 30:03
Yes, that is always difficult, especially with operating system projects, because there is no clear or not always a, clear definition of done. It's an operating system, a it's always a work in progress.

So setting a milestone is always going to be a bit of guesswork uh. How far how far do you think you will get with this project? Is like asking how long is a piece of string? So the challenge is to come up with small sub projects which can be tested or which can be which at least compile, or sometimes the milestone is some documentation or a plan. What's the best way to do this?

So this is this is actually a difficult project. Setting milestones is one that's ii find it difficult to set milestones for myself as well. It's something everyone struggles with. I think if it's not a huge problem, but it's one that's always needs some time.

Speaker 2 31:22
Yeah, I also think setting milestones is very important, especially in such a very short time for GSoC also. Let's move to the last part. Last part is about the strategies. The first one is about the strategies for the organization application. Okay, because you haven't participated in this application, but you can. Do you have some comments on the overall results?

Speaker 1 32:03
Yes, it's ii think the guiding principle is make it easy for people to work with you, be as clear as you could possibly be.
And we've learned this with the previously projects over the years is make a list of ideas and keep it updated throughout the year. Don't wait until just before GSoC to start writing ideas down. It's a living document of projects that we think are useful for GSoC students. So we keep that on our web page somewhere, and it's easy to find. And that also makes it easy for the GSoC admins. To just put a link to that, ship it off to google, and it gets approved. So it's important, but it's always important to be clear.
But particularly when you're seeking funding, or when you're uh trying to find students to work on things, having clear descriptions of what it is, you would want them to work on is very helpful.

Speaker 2 33:21
Do you think it is difficult to write idealist? I coming up these ideas?

Speaker 1 33:27
It is very difficult. It's both very easy and very difficult and. It's very easy to write down ideas. It is very difficult to write down useful ideas. It is very easy to run into a problem and just write it down. There is a problem here.
But then going back and actually describing what the idea is in a way that is actionable for students is work. It's not difficult work as such, but it is work, and it is time consuming. Because unfortunately, students cannot reads my mind. So you can't, you need to condense a lot of information, into a, small paragraph, and be clear about expectations.

Speaker 2 34:21
I think it's difficult because you need to combine the development status of your project as well as the difficulty of these tasks. You cannot put forward too difficult tasks or students。

Speaker 1 34:42
and tasks evolve as well. It is. It's always possible. That's what used to be a clearly defined task. Has been overtaken by events, because something else in a project changed. This is particularly true in operating systems where you have many moving parts that all need to interact. If your idea was about one small piece, it's going to be affected by other pieces. So you need to keep the list up to dates.

Speaker 2 35:13
Okay? Uh let's move to the next part. That's the part is about the strategies using during the student selection and community boarding period. Could you explain your answers?

Speaker 1 35:31
I strongly believe in a lot of communication. Communicating more is better and uh work with a mechanism that works for the students. So if a student does not like writing emails, but enjoys chatting on google or on signal or wechat or whatever, pick whatever works for the students and just work with them. That is just very important. As for technical skills, those usually become immediately apparent. I think, if you manage to submit a proposal to the three [redacted] projects, you automatically have the necessary technical skills. Because if you can even interpret the idea in a way that gets you to a proposal, you have the technical skills. As I say, we are an operating system, so we're pretty self-limiting in the sort of people we attract, and they tend to have the technical skills they needs.

I don't think I've ever gone and look through get our profiles for history. I why? Because it's never been necessary. It just becomes a parents that someone is competence and technically able to work from the way they interact with me, either from the way they type or the way they structure their questions. It's you can tell an engineer from an engineer, right? That if you hang out with engineers, you automatically get a sort of feeling for people who know what they're doing.

Speaker 2 37:32
How about if a student they have solved some medium priority task? Do you think the student is a good student?

Speaker 1 37:48
Yes, that always helps, because it shows that they're willing to learn. So and that they're willing to just work on things for fun. So even if it's, they this this happened in the past, they were trying to set up their development environment and ran into a bug.

If they then go and fix this bug, then that obviously speaks very highly for the and that really works. Well. I I think my student this year fixed a bug while he was applying.

Speaker 2 38:28
It's very important for student to come complete some task before the application.

Speaker 1 38:39
I wouldn't say it's important, but it certainly helps. It's, I say, if you apply as a students, and you already have a long track records of interacting with the projects, if we are not going to worry too much about accepting your proposal if you show up out of nowhere with a poorly written proposal that shows that you can barely read what the idea is, then we're probably not going to accept you.

And most of our applications are somewhere in the middle of that spectrum, clear, clearly incompetent on the one side. And immediately except on the other sides, most of our applications are closer to this side.

Speaker 2 39:31
Okay, you strongly agree with assess whether the student can talk like an online friend? Is it not necessary?

Speaker 1 39:43
Can I strongly disagree that a student can talk like an online friend? I don't really care how they talk uh. I think it's more important that they can communicate what's on their minds. The form really doesn't matter.

Speaker 2 40:03
Did you check whether the student has applied where else?

Speaker 1 40:12
I just do a quick look if the name is familiar. I'm obviously very keen to work with them. I do take a quick look at the mailing list archives to see if has this person contributed before. And usually what happens is the students will write it in their proposal uh. I have worked on this before. And then. I don't even have to go. And so we also find that most students apply only to us, because they really want to work with us. It's been, ii can't remember any student recently who has applied to multiple organizations.

Speaker 2 40:58
Okay, maybe if they apply to, multiple organizations, they need to spend a lot of time.

Speaker 1 41:08
Yes, I think especially with operating systems, if you want to write a coherent application to work on an operating system, that's a helps grown-up are serious investment in time. If you have time to apply for 40 other projects, then you're probably not committed enough to work on our project.

Speaker 2 41:32
Yes you, do you think the summer plans of the students is very important?

Speaker 1 41:39
They're important to some extent. It's iii don't mind if the students wants to go and sit on the beach for most of the summer if they still get work done. That's great. If the student tells me that I am going to lock myself up at my desk and work 20 hours a day, all summer, then I am much more likely to reject the students, because that's not the realistic expectation.

Speaker 2 42:07
Yeah. Could you explain the last one?

Speaker 1 42:16
I agree if a student disappears for the reason or just utterly fails to communicate, just fail them. There is really no point in bothering with them. It's GSoC students are meant to be adults. They are meant to be in tertiary education. They are meant to be a university. If if they are not doing the works, then it is a lot better for them to fail them than it is to try to enable them, get something out of them. We haven't.

In the early years, we've had students, we failed in the first or second week, and they applied again the next year or 2 years later, and they were much more mature, much more grown up. So we like to think that failing them early, help them on that journey. it sounds harsh. I have get no mitigating words for us. If you if you're not going to do the work, you promise to do, then I should not spend my time trying to get it out of here because then ultimate. Ultimately I end up doing the work, right?

Speaker 2 43:54
Yes, I agree with you because google they give the money to the student, so they should finish their work.

Speaker 1 44:02
I don't really care about the money too much. I care more about the time and about the about the benefit to the students. If the students gets free money for doing nothing, that's not great. But more importantly, if I have to spend a lot of time trying to get something out of the students, ultimately, I am doing the work that the student should be doing, and that's not okay.

Speaker 2 44:32
Yeah. Okay, the next part is the strategies. Uh during, the official execution. Could you explain your answers?

Speaker 1 44:47
I think this page just comes down to how to be a nice human being. I think all of these attitudes apply, in general, to interacting with people, whether they are students or whether they are people on the bus sitting next to you. It's it seems clear that if you make a mistake, you should apologize for it and the students are working on a project for you.
So clearly, their opinion matters for in making decisions. And I always feel inclusive this is importance. Mono cultures are terrible. And students are young, silly questions are great. I've said it before. More questions are better. So all of these are just be a nice person.

Speaker 1 45:54
The one I would like to talk about is the request regular reports. I think that has worked really well for us. Over the years is a few sets, set a weekly call, or you ask them to just every Friday, please write me a little email to paragraphs. What if you done? Where are you stuck? And it doesn't need to be difficult. It doesn't need to be long, just a simple regular report. That works really well.

Speaker 2 46:27
May I know how many communication applications do you use?

Speaker 1 46:33
Email, IRC and signal. We did video calls as well. So for。

Speaker 2 46:48
so do you think the communication is very efficient?

Speaker 1 46:56
I think so especially if you're clear with the students about which one they prefer to use. So if the students really like to use signal on their phone or wechat or whatever, then just use that one. If the student prefers to email, use the one that works best, it's always good to have email as well.

Speaker 2 47:27
Do you think it's difficult to set the completion dates with the task?

Speaker 1 47:38
Yes, that's always uh. That's always difficult because the tasks will change. So I'm much more interested in the progress reports. Where are you? Rather than on when are you? manage the end of the day, what matters is that there has been visible progress. It is perfectly possible that's about halfway into a project. It turns out that the actual idea is wrong and if. The students manages to come up with a better idea as part of their work. That's great. And you can't put a completion date on that. It's generally much more open ended.

Speaker 2 48:26
Could you explain the strategies of the make sure that students are familiar with the workflow of the community as early as a possible.

Speaker 1 48:44
especially in the [redacted] project we have, but we're an operating system. We have a complicated build system. It's important if you are going to get any work done on previously that you can actually build the operating system. And it's important that you have a way of sharing your contributions with us.
I don't really want to get into the bureaucratic details of how to submit a patch and how to get it reviewed, or how to make sure it passes all the automated tests. But I think it's important that the students are familiar with the basic requirements. How do I get a patch out of here in a way that someone else can review it? I don't pay too much attention to the entire process. The broad lines, the vague lines of the workload should be familiar to them. And the rest they can pick up as they go along.

Speaker 2 50:02
Why don't you pay attention to the entire process? Especially the workflow.

Speaker 1 50:16
You pick it up as you go along. So you learn it by doing. There is, I see no needs to spend 3 days with the students showing them all of the different options of our backtracking system, or all of the ways our code review project works.

So at, the beginning, what I care about is, can they send me a patch either by email or just send push something to get repository? I don't care how if I can see the codes, I am happy. And then as the code progresses, and the patch gets more completes, will go through the process of it's now starting to look goods. Why don't you submit it to the review system and then spend time on going through that? As they run into bugs, we can explain the workflow for submitting bugs. But at the beginning of, the earliest possible doesn't really matter what matters at the beginning is that the patches come out, what they look like doesn't matter where they put them, doesn't matter as long as they exist.

Speaker 2 51:32
I think so. The next part is about what kind of tasks do you think a suitable for newcomers? Do you think it is difficult to identify a suitable task for newcomers?

Speaker 1 51:50
It depends on the newcomer.

I think in this listing here, I think if your tasks are only low hanging fruits, you're going to turn off submitters or applicants, because they will not feel any sort of satisfaction if all the work is trivial they are, not going to be interested in doing it on the other hands, if all the work is risky and exploratory, it's really difficult to keep students engaged and get something useful advocates. Because it's very possible that you spend the summer doing something that just gets thrown away.

So the first two categories, I would be very careful with fun and peripheral projects are really great for students, because students are young and they have new ideas, do something fun and do something peripheral to the operating system. I also do something that we all operating system people don't usually do is useful. We have a few years ago, we have a summer of code students work on our web page, our web pages and embarrassments. We don't do web pages. We write operating systems. So that was a peripheral project that was very important. And we're using this web page now, and it helps us get new people. So there has to be a balance, also infrastructure and automation are in that box as well. We write operating systems, we build systems, someone has to test them, someone has to automate them. It's great. If we have motivated students who can work on this, it's a little tricky to mentor them, because it's not our core competence, but it's great to have students doing this.

Speaker 2 53:54
Some mentors, they mentioned that finding fun, peripheral tasks, it's very difficult. Do you think so?

Speaker 1 54:06
I don't think so. I think actually it's one of the easier ones, because it's things that we don't have strong opinions on. So things we have strong opinions on are more difficult to mentor, because we are less likely to let the students follow their own path. And do it the way they want to do. Whereas a fun and, peripheral project will have opinions on them and will be able to tell whether they are doing a good job, but we are less we are less likely to micro manage, or we are less likely to have difficult interactions with the students if they're working on something peripheral, and it's useful to have as well, projects need web pages, and nobody wants to make it.

If to put it bluntly, if google wants to pay a student for 2 months to build a website, that's great.

Speaker 2 55:14
The next part, the last part is about what did you gain from being a mentor in GSoC could you explain your answers it is?

Speaker 1 55:32
Always very satisfying when a student goes on to do great things in this year? Student was great. He had a project completed his project in no time at all and did a much better job of it than anyone could have expected.

He went on to fix more problems and bugs for the remaining time. That's very satisfying for a mentor when amenity goes and does something that just makes me happy. Learning. I don't really learn much from mantis. I think it's polishing existing skills. It's part for the course. Peer recognition we talked about before. I don't care the nations or rewards. I don't care. If they don't matter to be. Okay.

Speaker 2 56:31
I think it did especially the aspect of learning. It depends on the expertise of the mentors.

Speaker 1 56:41
It sometimes you always learn something, but that's my life. Changes from 2 months of mentoring a student, so you always learn something. You get, you shed new lights on the codes. It's always you learn from new people, from talking to them, but it's not a huge amount.

Speaker 2 57:06
That's all the questions. Do you have other comments or suggestions for building a good GSoC program.

Speaker 1 57:22
I think you should keep doing this, and you should interview more and more mentors from different organizations, because I'm very interested in the results. I I like your numbers. I'm going to look through them in a bit more detail. I think he said to me that presentation. I think it's good to analyze open source projects from the outsides. And I hope this is useful to Google as well that they see the impacts of the money they're spending. Yeah. Yes.

Speaker 2 57:59
I'm very happy to have this talk with you, and you give me very useful ideas.

Speaker 1 58:07
I hope you don't have to try too much when you listen to the recording, okay?

Speaker 2 58:19
Um um. In order to show our appreciate, we will send you $50 after this interview. Thank you very much. That's great.

Speaker 1 58:29
I will send it on to charity. Thanks for talking to me. Have a good afternoon. Bye, bye.