Person1: From your point of view what is sustainability in terms of software?

Person2: Right. So, I mean I think from my point of view um it is uh practices um sorry, processes as well as tools that are used in these processes that allow um for uh software to be um, basically to keep, to keep good track of, of, of what you're developing, of who is doing what development. Um so, the things to, just I would day that you're doing things that allow you in future to um reuse, efficiently and correctly reuse and uh extend uh software that you briefly produced.

Person1: Mm-hmm (affirmative).

Person2: And also to be able to refer back to it um, uh accurately um when you're for example discussing results so especially within the uh scientific context. I would say sustainability is partly about you know this use of reuse-ability of research, and uh being able to say, well um it's kind of you know it's important to, to keep track of uh, um your developments because you want to see exactly which version of the code produced what results that went into some application. Um, that's simply becoming very important for, for research.

Person1: Mm-hmm (affirmative).

Person2: ...In the scientific world. So, yeah I mean I guess there are kind of two arenas for sustainability that I see; that's that one the one hand you have these I guess big applications, scientific applications where there's maybe one original author but over time there's a team of developers that have grown up around it. I'm thinking of like big aerodynamics code, I mean computational chemistry codes for example. And they're um, these have often adopted uh practices that work for them. Specifically practices that work for the original author with regards to the process of integrating changes, uh doing releases of the software, and taking care of how the software can, how bugs are fixed, how the software can be deployed or extended to be deployed on different platforms. So they have, there's a method that works basically somehow it works.

Um, now what sustainability means there I guess is that it continues to work, and that can be an issue when it comes to um the people involved because I think a big part of sustainability is not just about the the tools that you um but using them intelligently. And it's not just about the tools themselves but the process in which you use them.

Person1: Yes.

Person2: Some people say, "Oh just use version control." That'd be great but you need to kind of have a good process where that fits into, and that involves people. So, with these big projects, these big applications that have been existed for a bit, some of them have existed for quite a long time they might have an original author or core groups of authors that um have their set way of doing things and if they want to uh adapt, uh they want to get new users, sorry they want to get eight new developers involved there might be issues trying, trying to mix. If their practices are not I guess um sustainable, uh I don't know how you define sustainable and that could be very not sustainable because it makes it hard for new developers to um that uh practice.

So, I guess that, that speaks to the notion of having some um not necessarily generic, but some kind of methodologies that are good practice, that are not, not massively quirky, and that are, that's there's some common understand of developing, you know that's not, that the developers don't interrupt their workflow which is so uh quirky, and for example naming can come down to things like file names. You know things that seem silly, but it's, it matters and it can add up in complexity where as a new developer if you join a project it can be so difficult.

Person1: Yeah.

Person2: It can take an extremely long time to understand the conventions, and the sort of embodied kind of work uh flow and the way that people, the way the original authors have um, uh just the way they see everything. And for them it's easy because they know, I mean like I know this from experience in, in this project that I just mentioned I've been working on where the original author is um has been maintaining, developing maintaining this code largely by himself over the past I guess twenty-five years or so. And um, he knows every nook, and cranny but for people who are new it's, it's uh, it's not easy. And it's made less easy not just by perhaps lack of good documentation, though there is some documentation in this case, but it's made difficult by programming practices which you could definitely define as not being sustainable.

Um, namely sort of global, global variables um things which mean that you need to look in every place at once. You need to have a godlike vision, and see it. You need to basically know everything in order to understand everything instead of being able to start somewhere, and have things be more encapsulated in terms of programming practice. Encapsulation, is, it's really (laugh) a lot of people don't appreciate how encapsulation is really a good sustainable practice because it means things are in a way less uh, more understandable you know for somebody who's new to the software. So that's one, that's the first arena that I wanted to mention which is the sort of software development uh arena for say large projects, big, big software projects where you really, you're really pursuing a big application.

I guess the other arena I thought was relevant um for sustainability, especially sustainability in the context of scientific research is they day-to-day, everyday workflow of people who's primary goal is not to produce software; uh but who are just using little bits of strips, little strips, little bits of code uh to um maybe to run small simulations but maybe just, just to, just to process some data ready for further analysis, or to actually plot stuff.

Person1: Mm-hmm (affirmative).

Person2: These can be very simple, and extremely wide spread of course. I mean everybody does this. All scientists to this to some extent. Um (tongue click) and, there I think there are a lot of um um you know, things that people that again um perhaps are not so sustainable which means that there are lots of wheels being reinvented.

Person1: Mm-hmm (affirmative).

Person2: And you may also have some um, um ... Well I was going to say actually in a research group for example you might have that things get handed down from one

generation of PhD students to the next. Uh, and to some extent that's good because as long as it's well documented you know that's good, that's not reinventing the wheel. That means that you are uh, it's uh that the original thing was apparently useful, that it's, it's sustainable to keep on using it.

Person1: Uh, regarding the software that you've developed what sustainability consideration?

Person2: Yeah, I mean basically very much so because one of the reasons why when we got involved in a project was actually to um try and uh part of was to improve the performance, um simply. So (tongue click) the parallel performance, but another aspect that we wanted to provide for the project and that we were planned to provide for the project was to improve uh sustainability. To basically uh try and promote a way of working which included uh changing maybe virtual control systems, moving to Github as a proposal, um and really trying to promote um dialogue between developers um so yeah definitely.

Person1: So it was a consideration from the beginning?

Person2: Oh absolutely, yes. It was planned in, yeah.

Person1: Yeah. Uh have you worked on any projects that were not sustainable?

Person2: Um, well you could argue (laughs) um. I mean yes. I mean don't think it's, I don't think that you can say they either were or weren't. I don't think it's a zero, or one situation. In some sense time will tell uh because the uh, you know we have some ideas of what could be considered sustainable because we think, oh this is a good working practice and everybody's doing this, or a lot of people are doing this therefore if you go through it and practice, it'll be sustainable. But um, time will tell because you know lot's of uh sort of for example programming practices that come from more mainstream software development world as opposed to scientific software development, like Agile development, Agile programming, Extreme programming all these things. They become very popular, and then there's a backlash against it. So um, I think we have to be very careful. They get to be sort of conservative and say, "Okay well um we may think that what we're doing is a good idea, but you know it might not." Now people say Github is great for version control and for pull requests, and to allow um good working models but it might be that um, it might be that you know that's just a fad or a tool that um will become less important.

Person1: I see.

Person2: I think, I think when it comes to sustainability it's more about like go back to what I said earlier, it's more about the people involved and the attention they give to in the processes and the way they work to um, sustainability. But to answer your question, yes I have been involved in projects uh that I guess may not turn out in the end to be sustainable.

Person1: And were there any consequences of it not being sustainable?

Person2: Um (tongue click), well again that, that still has to be proven, but I think the consequences, the consequences as far as you can see is that um ... The consequences are that it's basically wasting time because it makes it more difficult. So what happens for example is that um because there's not enough communication for example, there's a lack of communication that could be one thing that leads to lack of sustainability and therefore um (tongue click) there is um ... For example a bug appears in uh one version of the, of a code and that doesn't get communicated well, and therefore people waste you know weeks or months uh not using the latest because perhaps they didn't check um what the updates were to the repository code, or to look at a way this bug was fixed.

Um, other issues I mean ... I was also involved with a project um a year, over a year and a half ago which was to setup basically a prototype website to um gather together um software projects um that were produced, funded by Institution1. Do you know Institution1?

Person1: No.

Person2: I forget what the acronym stands for, but if you go to Institution1.ac.uk you'll see it's a, basically they provide software development for um for higher and further education in the UK.

Person1: Mm-hmm (affirmative).

Person2: So, uh what we noticed, it was an interesting challenge in a way because we had a list, a big database of lists that were projects that had supposedly produced software. These projects were funded by Institution1 and they produced software, and we wanted to, to find out lots of information about these and basically put them into a website. So you basically spent a lot of time, um chasing up these, these projects and some of them are a few years old so between, maybe between fifteen years-old and still just having started. So it's, you gotta really ... You really see that things got out of date. A lot of projects that fizzle out, and don't exist anymore.

Person1: Yeah.

Person2: They just, they ... You lose all traces, and and documentation goes out of date and everything goes out of date. And just basically very little to count on. Um, now why that happened, so clearly these were not in some way sustainable. Now why that happened is difficult to say. Maybe they weren't a good, maybe the software was only ever meant to be a prototype so it was never meant to lead to anything further, or maybe the, it just didn't find it's right user group.

Person1: Yeah. I've, I've come across this before.

Person2: Yeah.

Person1: So Person2 that's all I have for you today.

Person2: Okay.