I1: Um, from your point of view what is sustainability in terms of software?

P: Um, I think it's keeping it well packaged and maintainable, documented. Um, obviously everything we do here is open-source so you got to, I mean there's a lot more to open-source than just making the source code available on-line, you've got to like make it easily installable and people can just check out the code easily and build it and run it. Um, I think we could use some, following some sort of coding conventions, um, following good practices. We developing code all helps towards sustainability, so keeping kind of it simple and clean. Um, yeah I that are the main things.

I1: Um, which attributes of, or features the software has that makes you believe that it is sustainable?

P: Um, so I'd say in project01 one of the, some of the values in project01 is it’s all very well tested. It's a lot of test automated and continuous integration testing, and I think that, that helps a lot with keeping it sustainable so every time we make changes all of it, it fires off all these tests that run all these tests and we get emails if they go wrong, and so it's, the whole like test and development is followed within Project01 and I think that, that's like a big value. It's not just the test itself it's the test code that sits next to it is, is um, it's worth a lot.

I1: Is there any other attributes or features?

P: Um, I think, I mean it's fairly modular, it's, it's going to keep it modular so it's easier to maintain, and continual churn of refactoring things so that it, you know it can cross bits of code anything you like even though it doesn't add a new feature. I think it’s worth the effort of going back and re-visiting it and to refactoring it, because otherwise it hits you in, you know a few years down the line.

I1: Um, regarding the software that you've developed, was sustainability consideration?

P: Yeah, I think so, yeah. Yeah always has been. Because of the, I mean I've worked here for quite a few years so I'm used to this whole like funding cycle and, you know you're paid for three years and then after two and a half years you start seeing a bit hairy and you're hoping for more funding, so you know with that in mind we try and keep the software sustainable in the hope that either it will get more funding to continue it, or if we don't get funding it would be a shame to see it all die and disappear so it would be nice if it was in a state that someone else could pick it up and use it. Yeah.

I1: Um, so was sustainability a consideration since the beginning, or it became a consideration after some point in time?

P: Um, I think since the beginning, it's, it's, yeah it's something that's always been-

I1: Discussed?

P: Yeah, but it's not necessarily open, well it is discussed from time to time when it's particular kind of stage is kind of in the funding, but I think um, there's a kind of natural background consideration going on to keep it sustainable. And I think I've, I mean I've worked with people in Country01 that work, it's a distribute development, and I think we, we think a lot more here than perhaps they do who've been our partners, so it is something that I'm quite often beating people up about, saying you know, "This is no good," and it causes a lot of arguments. Um, I think it's worth it in the long run.

I1: In the long run?

P: Yeah.

I1: Um, have you worked on any projects that were not sustainable, and were there any consequences of it not being sustainable?

P: Um, I think to some extent project03 went, I mean it's improved now but project03 went down a path where it became, it, there was a lot of inside information needed on how to build it and how the code structured, it became very complicated. Um, so yeah I think that would be one, one example of that, so there were like, there were like tricks of build that you needed to know that secret Python scripts that weren't documented. And only one person in the office knows about. So someone couldn't just come along and easily build it, it required this kind of inside information.

I1: So-

P: I think that's improved now.

I1: Can you, can you say that, that's a consequence of it not being sustainable?

P: Um, no I would say it's the other way around, I would say that was a cause of, you know a threading it's sustainability.

I1: I see.

P: Um, maybe I didn't quite answer the question right there. Sort of consequences of software not-

I1: Being sustainable?

P: Um-

I1: The ones that you've worked on that were not.

P: No, not I don't think so, no.

I1: No?

P: No, I mean it’s something I always try and either it starts heading towards having consequences of it not being sustainable I tend to try and you know get that in check before it gets to a point where it does become a problem.

I1: I see. Well that's all I have for today.

P: All right.

I1: It's supposed to be really quick.

P: All right, okay.

I1: All right.

P: All right, thanks.

I1: Thank you very much. You were-

P: No problem.