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

Person2: Well, oh ... To me, it means someone should be able to build, run, and understand your software, say one year, five year, or ten years after you're finished with it, and without having to come and ask you how to build it, or why you did things in a certain way.

Person1: Hmm.

Person2: So, the software becomes a sort of like- A standalone from the individual that wrote it.

Person1: Uh, what are the attributes or features of the software itself the let you believe that it's sustainable?

Person2: Uh, it's well commented and documented. It is uh, designed in a modular, maintainable fashion. It provides information as to how- What you need to do to build, compile, and run the software.

Person1: Hmm.

Person2: Provides tests to validate that- Run correctly. Provides uh, some form of documentation about the high level design- The rationale. Why it's as it is. So that, to me, would be the, the main ones, I think.

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

Person2: Yeah. So I'd was hoping someone else could understand my software, or software I'd worked on, without me being there. That's especially important when we're actually helping research groups come sustainable. If we develop components for them, we do one thing today and come back six months later, and say, "Well, how do we use this?", because then we've failed.

Rather ...

Person1: Hmm.

Person2: ... we want to do some work with them, and then you can take it on as if we were never there.

Person1: Hmm.

Person2: And continue with that.

Person1: Uh, was- So, was sustainability a consideration from the beginning, or no?

Person2: Yes.

Person1: Yeah?

Person2: But it's because this is institute1, so ...

Person1: Mm-hmm (affirmative), I see. Yeah.

Person2: But, from some of the projects we have worked with, you can tell that, for them, it was necessarily-

Person1: From the beginning.

Person2: - Something that we have come to realise is important. But I- When we started, they were more concerned about just getting something running-

Person1: Hmm.

Person2: - To get some data to do a paper, for example.

Person1: Um, have you done any projects that were not sustainable?

Person2: Ooh ... Ow (laughs). Well, that's the tricky one. I can say they're all sustainable at time we left. But I guess the truth is, if I were to go back and understand them now, I'd sort of say- I mean to say there was none that were not sustainable, but I'd say there was a sort of, uh, a spectrum from ones I could just go back to and understand why it is as it is, and other ones that I'd really be better off to start again from scratch, so ...

Person1: Mm-hmm (affirmative)

Person2: I see the whole spectrum, really.

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

Person2: Um ... Ah yes, I- I have an example. Uh, okay, so one of the projects I work with as part of Institute1 involved porting a MPLAB code in- vi- Uh, C++. But then, once we'd done that and left, the people we did it with, did not develop the visual- Uh, the C++ one further.

Person1: Hmm.

Person2: So, they did not sustain it in any way. So, that just died when we all left. And that was, I think, because their understanding of C++ was not as great as they'd led us to believe, coupled with a reluctance to invest the time to actually migrate to using the C++ codebase, rather than the MPLAB codebase.

Person1: Hmm.

Person2: Which would take a little bit of time. So yes, I think that one wasn't sustainable- Or was not sustained.

Person1: I see.

Person2: I guess if there is a reluctance amongst the people who are made to be sustaining it, to sustain it, then it was- Yeah, it's not sustainable.

Person1: Hmm. Well, uh, that's all I have for you to ...