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

Person2: Uh, so, a couple of different things. In terms of the scientific software, then the ... one of the important things that we are very concerned about is that software last beyond the lifetime of the PhD student that wrote it, or the post-doc, or whoever in the group that wrote it.

Because, often it happens ... in what we call, quotation mark, "PhD ware" that someone writes something which is good enough to produce a paper. But then afterwards, no one basically knows what it is. So, in that way we are very concerned about trying to write software which has a longer life time which has ... can be read, understood, rewritten, and maintained by other people.

So that's one important thing for us when it comes to sustainability of uh ... of scientific software ... That, that the software is a product that can be used beyond the lifetime, and not just a tool for grading that specific paper or dissertation.

Person1: Hmm.

Person2: Um ... So there are various techniques that we would encourage people to use which asked of like ... unit testing, and continuous integration, and deployment tools, um, to make deployment.

Person1: Hmm.

Person2: Depending on which kind of project. But, deployment tools make it easier to deploy on different platforms.

Person1: Um, what, what are the attributes or features of the software that lead you to believe that it is sustainable?

Person2: Well, we certainly know why some of the software is not sustainable as it is, because we have seen this happen over and over again, that ... that software falls into disuse.

Um, so we try to redeem that. But whether or not this stuff that we make is successful and will be sustainable ... I guess I would say that only time can really tell.

Person1: Hmm.

Person2: Uh, but ... we believe that the experience is that if you do these kind of things, and if you do unit testing ... if you enable yourself to write a piece of software that can be re-factored, and that you have documented to unit test how it's intended to work, then you stand a much better chance of ... being able to maintain it, and sustain it.

But, another thing with scientific software, is of course, that it is sensible to funding. Because ... we are paid, or we are working on short term projects.

Person1: Hmm.

Person2: So, it may be that, once our involvement leaves, that is very crucial that, that someone else ... that knows what this is and can take over the development. Um, so we try to do that by working as close as we can together with the scientist, and, to insure we don't sit down and write something ... um, which they don't, then don't understand.

So, that's the main idea that we try to ... meet, and talk, and discuss, and interact both ways, and ... and hopefully that we will succeed in uh, creating some somewhat more ... sustainable ...

Person1: Hmm. Um, regarding the software that you have developed, uh, was sustainability a consideration?

Person2: Yes. So, most of the software that we develop is not, like, greenfield, we start from scratch, but we work on a project where scientist typically have some existing software. We might do a full re-implementation, but typically we work on the basis that they have, and typically that's not ... from our perspective very sustainable. Because it's ... often lacks proper testing and it's often very much written by a single person. Um, and it can be a challenge to ... try to put these things into the software retrospectively. But, we try our best to ... to add to the new features that we do ... um, stuff, and that makes it more sustainable.

Person1: Um, if yes, at what point in time did it become a consideration? Sustainability in the [inaudible 00:05:54] ...

Person2: Um, so, so in, in our ... um, in our work here, it's always been the team ... it's always been the ... key goal of the team to work with scientist on making ...

Person1: From the beginning.

Person2: Yeah. On making more sustainable software.

Person1: From the beginning.

Person2: Yeah, so, for ... from ... since the team started before I joined, um, that's always been one of the sort of key goals, of, of making ... and that ... of making the software more sustainable.

Person1: Hmm.

Person2: In software that I did before, I gradually, at some point, started following open source projects realising that there is a better way than this, and I tried adding test to some of the software that I had taken over from older PHD students, and, when I did re-factoring, and there was some test that so on so, that ... sort of realising what goes on in the wider community of software development ... that unit testing and stuff like that is really a good idea to make it ... operation control, for that matter, which is typically not even used in scientific software.

Person1: Hmm. Um, have you worked on any projects that were not sustainable, and were there any consequences of it not being sustainable?

Person2: Uh ... well ... it's hard to say whether or not they would show up to be sustainable. But, we certainly worked on projects where we think that ... if this has been done in a different way from the beginning, then we would have been way more productive ...

Person1: Mm-hmm (affirmative).

Person2: Because, the software that we took over was not really sustainable, if could say it like that. So, we tried to make it more sustainable. Given ... but given that it had been more sustainable, then ... you would easily have been productive because you could re-factor. If it had unit test you can re-factor the code, if you had operation control you can go back and figure out ... why is stuff going on, and stuff like that, and, well more documentation, more testing, and stuff like that, um, would ... would have enabled us to develop software faster. But, on the other hand, that is also what we are here for.

Person1: Hmm. I understand. Well um, that is all I have for today.

Person2: Okay.