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

Person2: I, I would suggest [inaudible 03:04] work at the [inaudible 03:05] maintained, updated [inaudible 03:11] from system to system. It is a word I have not heard in terms of software before, but that was my understanding [inaudible 03:23] how they maintained it when it was ...

Person1: And --

Person2: How changeable.

Person1: Uh-huh. What are the attributes or features the software itself, has that led, led you believe that it is sustainable?

Person2: In order to be [inaudible 03:40], it needs to, the code needs to be fairly clear. It needs to be in sizable chunks, very well structured. You don't want to spend weeks, I did in my first one, first job six years ago, [inaudible 03:53] software from when, from when the [inaudible 03:58] computers to [inaudible 03:59]. It was Fortran, it was all totally dependent on the computer it was written for, it was [inaudible 04:05], and I used to spend weeks putting, [inaudible 04:10] that, wrote multiple [inaudible 04:11] on the lab bench, and deconstructing the software. You do not want to have to do that.

Person1: Mm-hmm (affirmative)

Person2: You want something that [inaudible 04:18] writes, clearly defined classes and objects, and then you can either make another child's class or at least you don't, you don't want to change it. You know what CHERRY is?

Person1: Mm-hmm (affirmative) Regarding the software that you've developed, what sustainability consideration?

Person2: We tend to work very much, as a [inaudible 04:44], or to [inaudible 04:47] sustainability does everything wrong. The, uh, just I think we've been, having written, I mean, written the Team1 software, I've been turning it into a project called Project1, most of the code, most of the functions, most of the units can be carried over, with very limited modification. We've got, we've got a, the database, we've got files which can be used to create the database, but when we're changing our question set we just need a different question file out there, that's about a conflict file, adjacent file, so we just moved in a different set of questions and you've got a different application.

Person1: So ... uh, what sustainability or consideration from the beginning or not?

Person2: Well, I intend to write everything [inaudible 05:44] anyways, so yes, there was a lot of consideration from the beginning, also having it at the experience of the [inaudible 05:50] which is not easy to understand.

Person1: Mm-hmm (affirmative)

Person2: It's not portable, so if, as it happened, we [inaudible 05:58] the team, that we, it's just the software [inaudible 06:01] added up, anyway, into an individual's, work in individual sections.

We've got a full log, a Jira track of what we've done, what each thing does.

Person1: Mm-hmm (affirmative)

Person2: That's not [inaudible 06:14] considered as sustainability from the start.

Person1: And have you worked --

Person2: Because this is all new, very new work.

Person1: Mm-hmm (affirmative)

Person2: This would not be the case if you went back a few years, let's say to [inaudible 06:24].

Person1: And have you worked on any projects that were not sustainable?

Person2: Um, Project2.

Person1: Mm-hmm (affirmative)

Person2: Also, the pro --, the work that I did in my fir --, my early days, the programming way back in the late 80s, early 90s, the work in the 90s was classic, was classic case of sustainability because the first line was go to 100 and the last line was go to one, and it was ... because somebody had decided to tack on a whole load of new functionality right at the start of the program, and that's how they did it. Totally unstructured, and that ... it was, I think, that was actually given to a family, a gra, a family of students as an undergraduate projects, to break us up into modules. [inaudible 07:16] is just seen through [inaudible 07:17] in a very, the database it's leading to, is directly [inaudible 07:25] keyed into the type of the Java code, you haven't got, you didn't start with a database, and then write the code to go with it. It seemed to go the other way around.

Person1: Mm-hmm (affirmative)

Person2: You can't pull up the database to any other system, because there's no structure that is usable to pull it. When you try to figure out what does anything, you find yourself wandering through masses of almost incomprehensible code.

Person1: Yeah, so --

Person2: So that was, that never considered any attempt to be, at portability or sustainability. It was in Java, which means it can be taken from computer to computer, but to change it is a nightmare.

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

Person2: Various, nice to have switch people who use a software, and [inaudible 08:11] we've just had to say, "No, this cannot be done," and a big consequence, now what people are trying to pull up the software off, they're trying to pull up the databases it created, off to a different databases. We're having to work backwards. We're having to run reports that print out the data, and then we are giving them the output reports, for which they have to get somebody else to create a database from scratch.

Person1: Mm-hmm (affirmative)

Person2: There's no way this database can be just moved, so it is having a big impact on the work [inaudible 08:50] just brought up that ...

Person1: Well, um, that's, that's all I have --