I1 - From your point of view, what is sustainability in terms of software?

p - Sustainability is the thing that we pretend we pay a lot of attention to, alright. It's the stuff that is done at the end. That's it.

I1 - Which attributes or features the software has that makes you believe that it is sustainable?

p - Erm.. I am not sure any software is truly sustainable on it's own. Right, you have put up effort upfront to make sure it's sustainable. Things that are quite important is using technologies that people generally understand, reusing as much as you can, so don't write your own things, but whereas there's good solutions already. And obviously documentation, although that in itself is another issue because you can write good and bad documentation. Things are quite nice, tests, but people get a bit obsessed about tests.

I1 - Yeah.
p - And I think people think you can, if you have a fully featured test set then that's your job done. When in reality you never gonna cover every eventuality with tests. All you doing is ah.. I think tests should be used to sort of demonstrate what the software can do and cover a few amount of functionality cause you never gonna cover it all. So it's there just as a kind of nice hint about how things works I guess.

I1 - So documentation, readability and tests.

p - I think that's the main features, there is probably more, but that's probably the main things that as a developer help most right?

I1 - Regarding the software you've developed, was sustainability a consideration?

p - I guess I put a lot of effort, as much as I, more and more I put an effort into tests. But tests is quite hard right? Because you get different aspects of the software right. You got the front end, you got the UI aspects and you got kind of the back end, you'd call it the functions, the things that drive it right? So the business logic and this kind of things. And the business logic side of it is relatively straightforward to test, because its all quite well defined. But testing the UI is hard, because people use these things in ways that you'd never ever thought, right? So I think that as a problem increasingly is that everything is UI focused. More and more and more right? And testing these things is really hard. Very time consuming. Really quite difficult. So I think testing the back end stuff is quite easy, but the problem comes when you try and write the tests try and have people using your software, use your interface. That's the bit I try to do but I struggle every time. Mainly because I dont think the technologies are that mature, you know, I think we're still learning, still learning. I mean the web moves at what kind of pace, the web changes constantly it's so hard to keep up.

I1 - I agree. Hum, So was sustainability considered from the beginning or after some time in the project?

p - That's a great question. I think it's considered at the beginning. But I think it's only considered as far as write it in a piece of paper. We would do sustainability. Right? Right. I don't think there's a lot of effort goes into it at the start. For various reasons why, I guess, one is that we, in these times of lack of budgets, you know, we do things with as few people as we possible can. Which means, you concentrate on the things which are really important, and that's essentially delivering the software, right? So you deliver the core functionality as fast as you can, and if you're writing tests, or if you're trying to sustain these things, and thinking about sustainability at the same time you don't have as much time to put into writing the software. So it's a trade-off. And I think at the moment, I think the sustainability is the thing that looses out. And then off course at the end of the project you start to worry about it. And then, yeah, then you have to spend a lot of money, you know, into documentation that kind of thing right? Things that you sort of do, but don't do as much as you should at the start. So, see, I think we're trying to do it but we struggle with a lack, yeah, a lack of people really.

I1 - I see.

p - Makes it hard.

I1 - Have you worked on any projects that were not sustainable?

p - (laughter). That's a gre, that's hum, I don't know. I am glad this is anonymous right?

I1 - Yeah yeah.

p - Erm… I I, to be honest, I think almost every project I've worked on isn't sustainable.

I1 - If so, were there any consequences of it not being sustainable?
p - The consequences it that none uses the software, right? 'You write this software and in a years time no one uses it. Because they can't, they can't deploy. Because the world moves on. Anyway, the, the operating systems move on the languages themselves move on. I can, I can various examples of where we, basically, deploy these things then forget about them and think they'll just be ok. And then suddenly no. You know, a apache comes out, needs applied. And then you find that actually half of the supplied libraries don't work because it's patched. And you know, eventually these things just decay. They just decay, right? We have various systems which are still up and running which to be honest is a miracle. Right? Because these things are so old and so antiquate and half of the libraries are just terrible, now, they are so badly supported and dysfunctional. And, yeah. It costs money, sustainability costs money and we don't have it.

I1- Yeah, I see. That's all I have for today. It is supposed to be really quick.