□□ Software for and by communities - interview with Dark Crystal Social Key Recovery System / CoBox

Mu and Peg worked on the projects Dark Crystal Social Key Recovery System and CoBox. In these and other projects they are working with a unique collaborative approach and a strong focus on community topics. They are sharing their knowledge and experiences on the topic in this interview.

> *In your projects - CoBox and Dark Crystal Social Key Recovery System - one can observe two different approaches towards the topic of community, a) the way the software is functioning because both projects are distributed software, i. e. they rely on a group of people using them; b) the developers form a collective/cooperative. Therefore, it seems like community is very important to you. How did you come up with the community approach for your software? Do you feel like distributed backups and collaborative clouds are more secure and otherwise advantageous - or is there even more to it?*

There's a lot to it! Of course we are organised as a collective - which is a big part of it - but we will share more about that below. In terms of the software itself, the community approach isn't valuable to us along a single vector - there are multiple dimensions to it.

For Dark Crystal, the project has its roots in the needs of a very concrete community, around the 'Secure Scuttlebutt' social network. Many 'peers' (as they are called within the community, to reflect the p2p architecture of the protocol) had expressed frustration when losing access to their account. So we designed a social recovery system through open discussion on that platform. Even at the very early design stages there was a lot of input from the people who would later use it.

In terms of whether the decentralised architecture of the recovery system made it more secure, yes we would argue that distributing responsibility across a trusted group has security advantages over the traditional consumer-provider model. But we think the co-design / community-centered approach to software can be very useful regardless of what architecture you decide on.

For CoBox, the community aspects were multiple and resulted from a coincidence of threads coming together around an architectural idea. All members of the team involved in developing the project idea had long histories in the European cooperative and collective action movements and

had experienced first-hand the frustration of there being few values-aligned digital tools available, forcing groups to rely on infrastructure that was in contradiction with their lived values.

At the time of its conception there was also increasing public awareness around surveillance capitalism and the personal, political and environmental dangers of massive centralised digital infrastructure, and therefore more tech funding was being directed towards supporting the underlying protocols on top of which such an alternative digital infrastructure could be built. Having stumbled across DAT, which was one of these underlying protocols, we decided to build on top of it to create the infrastructure we felt was missing from the communities we participated in. To do so, we deliberately set out to engage this community in the process of collaborative design.

So as we have come to understand it, 'community' can manifest in a few different ways. It can manifest infrastructurally in the software, socially in the team, and also in the input that spans these two components, so that both Dark Crystal and CoBox were developed in collaboration with the people they were being built for.

A fourth manifestation of community that we haven't discussed above, but will do below, is the community of developers who are building complementary digital infrastructures, to address parallel needs and based upon similar values. This has been where Dark Crystal ultimately found the most traction, as while the UX for distributed key backup came directly from the participation of end-users, it is other tech projects, needing to integrate a key-backup feature into their applications, who have become the primary consumers of this tool.

> *For a lot of users your approach might be unusual, too. How do those get involved, how do you “target” them? Is there for instance some sort of critical mass of users that you need in order for the community to grow from within, i.e. without you having to actively acquire new people?*

We are acutely aware of the difference between the information available to big corporations for determining preferences and the tools they have to collect usage-data vs the capacities that small community projects have to do so. It is therefore significantly easier for a big corporation to create an 'ideal' product without requiring much conscious participation from any of their users (through, for example, mass A/B testing and the like). As a small team working on a project and attempting to use tools and practices that align with a particular set of values, it is much more important to speak with people directly, to find out what they want and need, and often these interviews and testing sessions reveal ideas and perspectives that had never occurred to us before.

In terms of finding users to help us design the product, in the case of CoBox (in which we more actively had to 'target' people) we used the channels that the cooperative movements we were involved with had already set-up. We posted on forums, spoke at a community event and contacted people we personally knew from within that milieu. However, we are aware that unlike the data that corporates are able to amass, our process requires a commitment of time from end-users, who are all busy with their own projects and work and community building activities. We have until now always managed to compensate users for their participation, and we make an effort to plan the sessions with an eye towards efficiency and respect.

If the tools being built accurately identify a need and solve the problem in an effective way then what we have seen is that a kind of network effect develops off the back of that. Because of the open-source nature of development, projects are able to tap into each others' work, rather than a particular feature being siloed into a single application as a 'USP' over the offer of a competing app. This is what happened with Dark Crystal, where we are still being approached by projects that have heard about the key backup tool through word of mouth or shared communication channels and would like to integrate it into their application for the benefit of their users.

So yes, there comes a point when the community around a project exists without needing much active publicity. But rather than a critical number of users, it is probably a critical level of interest - you can have a small number of people who are very enthusiastic or actively involved, and it is enough to build a project around.

> *How do you organize your work as a cooperative/collective? Is it very different from a traditional firm? How did you come up with it and why did you decide to organize your group in that way?*

Our decision to work this way is rooted in our belief that work can and should be fulfilling, rather than alienating and exploitative. And in our understanding (which is ever adapting and developing) this derives from a feeling of shared ownership, agency, belonging and mutual respect between individuals who are standing on level ground.

Workplace dynamics can be really debilitating when there is no possibility to feel heard and engaged in your work, or when that work conflicts too sharply with your personal values and with what you would like to see exist in the world. At least one aspect of which is the case, as far as we know, in many workplaces, technical or not. Of course we recognise that especially among tech workers there can be some appeal to forgoing the above for work that is extremely well compensated. But such work can often also conflict with other personal and social values, making the compromise even less appealing in sum.

Internally, we try to organise in a way that there are not big power imbalances in terms of making decisions or accessing resources like servers or the bank account. There are, of course, many complexities to working this way. One challenge for a group that works like this is the process by which someone joins or leaves the group. When someone who is helping on a particular project becomes more involved, there needs to be some formal process which makes it clear at what point they become a member.

> *Maybe you could also tell us a bit about your approach to design and what your decision process looks like in that area. E. g., on the CoBox website you write about the peer-design process - what is that?*

Big corporations can do massive and continuous user-testing, unbeknownst to the user. Sometimes this is ugly and manipulative (for example, Facebook's psychological impact testing), but it doesn't have to be. When you do not have the capacity to do this kind of research, you have to rely on smaller and more personal processes. This is emphasised in the start-up community as well, where the available resources are still grand in comparison to what we have access to, but much smaller

than what established companies have access to. The co-design process has a lot of benefits. It not only allows you to design and create a product that the people you are building for actually want - taking into account their more subtle and difficult to identify needs, but it also allows you to very consciously choose to prioritise particular communities who may be underserved by other products available.

There are of course also challenges with co-design and user-testing. For example, when you are developing a new architecture, and developing a UX for it, the fact that it is quite far removed from what is already familiar, comfortable to and expected by people can be a challenge. The UX-design coaching that Dark Crystal received from Simply Secure as part of our funding was really helpful towards addressing this challenge, because distributed key backup or recovery involves some concepts which are quite unconventional. It was therefore difficult for us to find terminology and UI patterns that were not too confusing and the design coaching helped a lot with that.

Another challenge that we faced is that sometimes what users want doesn't align at all with what you want, or what you thought they would want. This happened to us during the second development phase of CoBox. After our first release, we had a clear roadmap that we thought prioritised what users would need, in order to begin using the tool. Our co-design interviews revealed that this was not the main or most important barrier to usage at all, and an issue that we had completely overlooked was far more prominent in their decision-making process about when or whether to begin using the app. Fortunately we were able to pivot in this instance, to prioritise addressing the issue that they raised.

Revision #6
Created 30 August 2022 07:29:58 by Claudia
Updated 26 September 2022 08:41:21 by patricia