

On cultivating the installable base

Roel Roscam Abbing

roel.roskam-abbings@mau.se

School of Arts and Communication (K3), Malmö Universitet

Malmö, Sweden

ABSTRACT

This paper argues that technically-inclined PD practices that wish "to matter" are well predisposed to significantly contribute to public repositories of computational alternatives. It makes that argument by providing an account from an ongoing collaboration with a group of artist collectives that seek to build its own alternative social and publishing platform. That process explicitly relies on and contributes to existing Free-Libre and Open Source software (F/LOSS) applications in the process. Using this collaboration as an example, it then argues, based on Leigh-Starr's notion of "installed base", that contributions to the installable base of computational alternatives can be a key contribution of PD practices. With this notion, this paper adds further substance to discussions of how to rethink PD's relation with socio-technical work by highlighting concrete points of intervention where PD can meaningfully contribute to technology design.

CCS CONCEPTS

• Human-centered computing → Participatory design.

KEYWORDS

Installed base, Free/Libre and Open Source Software, Advanced prototypes

ACM Reference Format:

Roel Roscam Abbing. 2022. On cultivating the installable base. In *Participatory Design Conference 2022: Volume 2 (PDC 2022 Vol. 2)*, August 19-September 1, 2022, Newcastle upon Tyne, United Kingdom. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3537797.3537875>

1 INTRODUCTION

Over the past years there has been a debate within the Participatory Design community on the nature of the practice of PD and how and whether to recentre on both the strong tradition of values-based and political work, and the technical nature of that work, that is, how historically PD has made technical contributions.

Korsgaard et al. argue that Participatory Design as a broad practice has lost interest in technical research as a core practice of PD [10]. The implication of that loss of interest makes one "not only miss out of implications for systems design in detail", but also "make the theoretical contributions less relevant by not being able to show how research findings might have an impact on technology

design" (ibid, p.72). The risk is that current systems and computational ways of doing are taken for granted, which "may make researchers and users insensitive towards the ideological premises embedded within (commercial) platforms" (ibid, p.73). They hark back to early Scandinavian PD, which they see as balancing social and technical aspects of research, and call for a return to that focus. Specifically, they call for PD to commit to the creation of and research in to "computational alternatives", which are concrete technological artefacts and practices, "that make us see possible worlds alternative to the actual world" (ibid, p.74).

Similarly, Bødker & Kyng outline an agenda for a new PD [8]. They argue for a PD "based on a concern for influencing our common technological future" (ibid, p. 4:20) through high technological ambitions to create alternatives for (commercial) technologies otherwise taken for granted. This in turn requires working prototypes as a research outcome. Prototypes which can see actual use within and outside the research context and which "make appropriation possible, and can support the users in making more radical changes, and—if the use process is followed actively—support changes initiated by the researchers." Stable working prototypes make it "possible to begin to scale up and involve new groups in other organizations". Yet, that scaling up of technologies in turn should happen in distinct ways, not through traditional commercialization, which threatens democratic control of the very same technologies being scaled. To do so effectively requires alliances outside the direct and typical partners of such projects. They propose an agenda which is simultaneously more ambitious, but also more humble, as it envisions the partners or collaborators of PD projects to be the main drivers (ibid, p.4:15).

2 INITIAL OBSERVATIONS FROM CO-DESIGNING ALTERNATIVE SOCIAL MEDIA AND PUBLICATION INFRASTRUCTURE

In February 2021, I was invited to join an effort by a group of artist collectives to help design a new on-line platform for them and their network of international peers. The group of collectives had been gathering for about a year prior, working together on visual mock-ups, concepts and having discussions on what their new platform should and should not do. The impetus for this initiative was mostly around the values of the group, as the collectives have strong views around collectivity, sharing resources, conviviality, reciprocity and learning together. These values permeate through everything they do and determine how they organize themselves and what artistic work they make. Because of this, the group had become increasingly uncomfortable with their reliance on extractive platforms, which are necessary to manifest their practices on-line. The Covid-19 pandemic, which made the platforms even more



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

PDC 2022 Vol. 2, August 19-September 1, 2022, Newcastle upon Tyne, United Kingdom
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9681-3/22/08.
<https://doi.org/10.1145/3537797.3537875>

central to and defining for their practices, further compounded this sense of unease. The group shared a multitude of concerns around the platforms, sometimes more concrete and sometimes more nebulous. The purported lack of visibility they experienced due to sorting algorithms on popular social media platforms was, the fact that their work, organization, and contacts were scattered over multiple of such platforms and how tools designed for the boardroom were the ones manifesting their practice. However, they were motivated not only out of concerns of shortcomings, but also out of desires. The desire to have a place of one's own, a hangout, where collaborators could be invited to and an environment which could be the basis for a collective identity for the collectives and their collaborators.

The process of discussions, ideation and sketching was already going on for a year by the time I was invited, but that process was a bit at a standstill. A difficulty was that the size and shape of the project remained amorphous and only growing because there was little in the way of material boundaries. That is, something tangible which could inform the process that could help more sharply define the realm of what is possible, but also what was actually desirable. Similarly, mock-ups as a method to ideate and concretize the project, although well utilized by the group, had lost their usefulness because it was hard to anchor them in anything concrete. This also made both the concerns and desires remain nebulous. Therefore, it was not clear how to continue. Then again, time pressure played a part, as budgets had to be drafted and earmarked and some form of the project had to be presented to the funders at some point. The contribution I wished to bring to this particular moment in the process, then, was to propose and follow a method of using fully functioning prototypes to test and try. These would be based on existing, lightly configured, software and allow for a playful trying of different functionalities and therefore help inform the urgencies of the group.

3 THE INSTALLED BASE

In the previous sections I've outlined both existing discussions within PD regarding the nature of technical work and also have given some context concerning the ongoing collaboration with the artist collective from which I draw my observations. At this point, it is helpful to characterize the notion of an installed base, since it lies at the heart of my argument.

In *Steps to an Ecology of Infrastructure* [13] and the *Ethnography of Infrastructure* [14] describe the properties through which to understand infrastructures. One of those properties is the notion of an installed base. It can be understood as the pre-existing contexts on which an infrastructure can be built. To paraphrase; infrastructures do not emerge out of nothing. Broadly speaking, the installed base consists of practices, conventions, tools, and systems that already exist prior to and are constitutive of the development of an infrastructure [7]. While the notion of installed base is generally useful to think about infrastructure, when it comes to software development the notion is especially tangible. One never starts a software project from scratch but rather builds upon existing concepts, software libraries, protocols, best practices and programming languages, or even the 'legacy' software already "in place". In addition, Aanestad et al. argue that the installed base is not only

a given "thing", but rather should also be treated as a conceptual lens. Through it, one can understand when and how something can become infrastructural, as it can help show when existing practices and systems become significant as triggers, resources, competitors, or alternatives when it comes to the creation of infrastructures (ibid, p .29).

As the eventual shape and use of an infrastructure is predicated upon that installed base, Aanestad et al. outline a notion of the "cultivation of the installed base" (ibid). Meaning, the infrastructures are not built, but rather grown over time out of and with the existing installed base. This cultivation entails that, "rather than design in the conventional sense, dealing with the evolution of infrastructures requires strategies to intervene and influence ongoing processes." (ibid) Concretely this means that "developing infrastructures entails engagement in processes of extension, recombination, substitution of parts and arrangements that already exist." Interestingly, this shifts the installed base from a set of givens, to preconditions that can be shaped. An installed base is simultaneously enabling and constraining the infrastructure that can emerge out of it. It therefore becomes interesting to ask how PD research outcomes could concretely contribute to the "cultivation" of artefacts, systems, and preconditions out of which computational alternatives can emerge. Can the installed base be considered a substrate of sorts, out of which new systems are developed (not necessarily by PD researchers) which can be extended or recombined by others to be part of their infrastructure of computational alternatives?

4 HOW ADVANCED PROTOTYPES RELY ON WHAT COULD BE CALLED AN INSTALLABLE BASE

While sketches, mock-ups, and mind maps all make up the repertoire of what constitutes a design's prototype, Bødker and Kyng argue that a PD that matters requires the development of working prototypes: "a mock-up is not, and will never become, an important part of the daily life of partners since it does not support them in their daily activities" (4:21). As previously described, due to the lack of actionability of mock-ups, the process of designing an alternative platform with the artistic collectives had reached an impasse. From the phase of mock-ups, at best, one could draft a list of requirements to be sent to a technical party that could implement them. However, given the fact that both desires for a new platform and concerns with existing ones remained so nebulous in the process, that seemed inappropriate. Instead, what would benefit the process were actionable prototypes that could provide situational backtalk [12](p. 79) and thus function as a way to interrogate the material. With actionable prototypes, we could learn together what was desirable and possible, by concretely trying and evaluating different options.

While the general plan remained vague in the mock-up process, a few requirements for the platform did get articulated. The collective was looking for a low-key hangout, to be together, meet one another, but also "run in to" one another. In addition, it had to be a way to collect, store, organize and search materials such as working documents, plans, and documentation of events. This also had to include a method to initiate video streams (such as talks) and the possibility to archive both those streams and other

videos. While most of the tools were to be internal, the group also desired possibilities to publish to a wider audience from within the platform, or to have a single point of information and updates for an outside audience.

To help move the prototyping process further, we approached a technology worker-cooperative to help with the development and the hosting of the web applications. Together with them, over several on-line meetings, the group discussed particular aspects coming out of the mock-up sessions, upon which we proposed different existing software applications that offered a rough approximation of the desired functionality. These meetings happened on a weekly basis and involved discussions, demonstrations, walkthroughs and comparisons with representatives of the artist collective. Based on the meetings, applications were then brought on-line and, after introductions, opened to the group to playfully and interactively try out, without having to commit to any particular application. This way, within a matter of weeks, we had a system that allowed us to start addressing some of the questions that naturally followed from the mock-up sessions by practically investigating them using existing applications which served as advanced prototypes. If the desire is to be able to “hang out”, for example, should that hanging out take the form supported by this or rather that application? Trying a particular application could help confirm the direction of inquiry, sharpen the requirements for the platform to be, and elucidate what still had to be developed. Importantly, the goal was not to establish which single application could satisfy all requirements at once, but rather to consider how existing applications could become subcomponents of an ecosystem of tools that together could be combined and integrated to constitute the platform.

Over time, one of the outcomes of these sessions was that members of the artist collective themselves started looking for existing software and introducing them to the meetings to be discussed and tried. This connected the mock-ups to actionable next steps. Using these existing software as advanced prototypes allowed for informed and concrete discussions about the platform’s functionality, language, governance and future possibilities. In addition, it made the process more concrete, allowing for more informed discussions on constraints in terms of budgets and timelines as well as prioritization. In effect, the existing body of different software applications that could potentially be installed and tried in this manner, thus came to represent the space of possibilities and directly helped imagine what could be.

Centrally to the prototyping method described above, in which existing applications could be interactively tried and evaluated together, was an architecture that allowed to quickly and easily stand up and evaluate the different web-applications. At the core of the prototype lay a Single-Sign On system which held the account information of the group. The SSO system in turn connected to the various applications we wished to try, so that people could log in to them with one and the same account and try out the applications for themselves. Not having to sign up dramatically lowered the stakes of trying a new application for participants. At the same time, the software was installed using so-called containers, which in turn lowered the barriers to installing or uninstalling software.

The SSO system and the applications were all exclusively Free/Libre and Open Source (F/LOSS) software packages. There were several motivations for this. First, to create the platform and

make it potentially function for multiple years, given the time frame and budget, the prototype had to rely primarily on pre-existing software suites. Ensuring the group’s values “worked through” the various elements of the platform, including the selection of software and providers, was another reason to look at F/LOSS specifically. However, practically, F/LOSS also helped ensure the longer-term sustainability of the platform. For example, in terms of maintainability, as one could work along existing communities of practice, using and maintaining the software. In addition, the ability to make meaningful modifications to the software was important and working along existing communities of software users and developers provided the ability to enrol expertise from multiple parties should we need it.

The project eventually took the shape of different pre-existing software applications running next to one another under different subdomains of the site. These applications all added particular functionality and together became considered the platform. The only custom components of this prototype were scripts which integrated with these applications and which allowed to users to publish materials to the platform’s public front page. These materials published would normally only be visible internally, to those part of the platform, such as videos in the video archive. The public front-page resided under the main domain name and was discoverable by a general audience.

4.1 Departing from what is already there

The software applications used are larger F/LOSS projects such as Nextcloud [6], Peertube [3], Keycloak [4], Element [2] and Mastodon [5]. Most of these projects are intended and developed as an alternative to existing commercial applications, and each also have meaningful (non-extractive) revenue streams and user bases. The software packages are supported by different mobile and desktop applications, localized in to different languages and shipped with both technical and user documentation which means it was possible to quickly evaluate them.

Mastodon, for example, is an alternative social media [9] project with affordances very close to those of Twitter. The software is recognizable and usable to people familiar with Twitter. It also modifies the format of the microblogging application in meaningful ways, however. For example, by being able to network separate installations of Mastodon together into a larger network that allows the exchange of messages, content and interactions between different providers. Similarly, Peertube closely mirrors the affordances of YouTube while Nextcloud and Element those of online office environments and team chats respectively.

Software like this closely mirrors existing applications, but at the same time changes fundamental parameters. For one, it allows groups to set them up themselves in trusted environments. It also allows for deep modification and integration into larger ecosystems of tools. Because of this, these applications are valuable as a way to kick-start discussions on computational alternatives, to “known” genres of software, as they give a concrete ways to try them, modify, invert and otherwise prototype with them.

The fact that for this project we were able to eventually move rather fast and make an advanced prototype was due to the availability of these software packages in the first place. This prototype

could then facilitate several dozens of participants, who could evaluate the possibilities of the platform being built on their own, crucially, not through descriptions and simulations but rather through practical and tangible engagement with the actual material. The way the group could iterate ideas quickly was fully based on the fact that we were able to draw from public repositories of ideas and concrete technological implementations of these ideas. These software packages were solid enough that they could be used as-is, but also open enough that they could be extended and modified in ways relevant to the project and possibly in ways relevant to others.

This body of user-facing applications can be considered an “installable base”. In the very literal sense that it consists of applications that can be installed, on servers, phones or otherwise and thus tried, modified and evaluated as part of a PD process. This installable base in essence is a public repository of computational alternatives, which to a large influence how groups can practically interrogate and change the affordances of known genres of software for themselves. The contribution to this installable base of computational alternatives should be one of the goals of PD.

5 CULTIVATING AN INSTALLABLE BASE

So, what does it mean to cultivate such an installable base? While the longer answer to that deserves a paper of its own, this section will describe how in our process of working on an alternative social and publication platform we attempted to do so in several ways.

Some of these contributions are fairly basic and common when it comes to working with F/LOSS software, such as reporting bugs or clarifying parts of the project’s documentation. Other contributions were more involved, for example when we found we had to refine and test or even create the integration with the Single Sign On provider in some software packages. More involved, still, was the articulation and implementation of novel features to the existing code bases. For that, we either worked with the technology co-operative who then implemented the features and maintained a version of the software with our customizations, upon which were offered these changes back to the project (but were not always accepted). Alternatively, we directly commissioned the developers of the projects we were relying on. When considering what a Participatory Design practice which cultivates the installable base could look like, this could be central. Working directly with project developers not only address the free labour issue rampant within F/LOSS software, it also ensures that expert maintainers of the main codebase are the ones that add novel features. This not only ensured the quality of the changes, it also meant that these changes became part of the main software to the benefit of other users.

Furthermore, establishing a relationship with developers can be a way to establish alliances with actors with roughly similar interests to the benefit of current and future collaborations. In a similar vein, establishing a relationship and collaboration with a technology worker cooperative, should be seen in the light of cultivating the installable base as well. Especially considering they are a party which is actively involved in attempts to help bring alternatives to current trajectories of platformization in to being. The outcomes of that collaboration resulted not only in increased knowledge and capacity for the different parties, but it also leads

to the concretion of concrete technical artefacts. In the process of getting the software to run for our purposes, the technology cooperative created container images, which make the software of the platform easier to distribute and install. These container images have since become part of the “Co-op Cloud” [1] initiative, which creates standardized and easy ways of installing such software available to other hosters and technology cooperatives.

6 DISCUSSION & CONCLUSION

A call such as the one made in this paper obviously raises questions. One of the big questions that follows out of the above, is who sets the agenda for the cultivation of such an installable base? In [8], Bødker and Kyng argue engaged partners should be driving the collaboration, and I argue along roughly similar lines: that it can be meaningful to contribute to existing projects to expand the body of installable applications and their quality. Especially as communities working on these existing applications are engaged and driven by values roughly similar to those of PD, such as the desire for computational alternatives. The question is, to what extent do the values actually overlap? For example, F/LOSS projects might provide computational alternatives, but might not necessarily offer democratic control over technology. Either because projects have “benevolent dictators for life” as the only governance structure or because the projects implicitly reinforce hierarchies commonly found in software designs [11].

Another question arises out of the fact that many, but not all, the contributions made to the installable base as part of the project described above have been obvious, necessary or otherwise low-hanging fruit contributions. These were contributions that had to be done to make the projects work in our architecture, and which for that reason found quick acceptance in the upstream projects. What contributions and relations are possible when the majority of obvious contributions have been made? More generally, how does one move past the transactional nature of quick code contributions towards longer-term relationships and alliances?

Finally, calls for computational alternatives such as in Korsgaard et al. fundamentally go deeper than what F/LOSS, and especially existing projects, might be able to offer. If one wishes to address the fundamental issues of Twitter, is an open-source Twitter-clone the right direction of inquiry? This also highlights the shortcoming of the employed methodology. To come out of the impasse after the mock-up process it was useful, and considering resources also necessary, to work from essentially a catalogue of pre-existing options to inform the group work. However, these options also necessarily strongly colour what is possible and how that is done. This in turn risks favouring “solutions” over genuine inquiry in to needs and possibilities, and reinforce pre-existing genres over novel ones.

To a certain degree, the questions above can only be answered by engaging practically with the matter. In any event, things like alternative social media suites or advanced publishing tools that can lie at the basis of alternative community platforms, platforms which are anchored in public values and democratic control over technology, are extremely complex enterprises that no single group or PD research project can create and sustain. Especially given the

often limited timespans we realistically have to work with. Therefore, in practice, one relies on what is already there. At the same time, what is already there informs us what is possible. Therefore, I argue that contributing to the installable base, that repository of existing user-facing software packages which help reconfigure what is computationally taken for granted, can be one of the core contributions of a socio-technically inclined PD practice that wishes to matter. Contributing to the installable base expands the existing possibilities for PD researchers and their collaborators to prototype with complex systems, upon which different sociotechnological configurations can be imaged and tried. However, this is also done by people outside PD-led collaborations as well. In effect, contributing to the material artefacts that make up the installable base becomes a way to scale the urgencies and questions that PD carries by making it possible for others to practically engage with those questions.

REFERENCES

- [1] 2022. The Co-op Cloud. <https://coopcloud.tech/>.
- [2] 2022. Element | Secure Collaboration and Messaging. <https://element.io/>.
- [3] 2022. JoinPeerTube. <https://joinpeertube.org>.
- [4] 2022. Keycloak. <https://www.keycloak.org/>.
- [5] 2022. Mastodon. <https://joinmastodon.org/>.
- [6] 2022. Nextcloud. <https://nextcloud.com>.
- [7] Margunn Aanestad, Miria Grisot, Ole Hanseth, and Polyxeni Vassilakopoulou. 2017. Information Infrastructures and the Challenge of the Installed Base. In *Information Infrastructures within European Health Care*, Margunn Aanestad, Miria Grisot, Ole Hanseth, and Polyxeni Vassilakopoulou (Eds.). Springer, 25–33.
- [8] Susanne Bødker and Morten Kyng. 2018. Participatory Design That Matters — Facing the Big Issues. *ACM Transactions on Computer-Human Interaction* 25, 1 (Feb. 2018), 4:1–4:31. <https://doi.org/10.1145/3152421>
- [9] Robert W. Gehl. 2015. The Case for Alternative Social Media. *Social Media + Society* 1, 2 (Sept. 2015), 205630511560433. <https://doi.org/10.1177/2056305115604338>
- [10] Henrik Korsgaard, Clemens Nylandsted Klokmoose, and Susanne Bødker. 2016. Computational Alternatives in Participatory Design: Putting the t Back in Socio-Technical Research. In *Proceedings of the 14th Participatory Design Conference: Full Papers - Volume 1 (PDC '16)*. Association for Computing Machinery, New York, NY, USA, 71–79. <https://doi.org/10.1145/2940299.2940314>
- [11] Nathan Schneider. 2021. Admins, Mods, and Benevolent Dictators for Life: The Implicit Feudalism of Online Communities. *New Media & Society* (Jan. 2021), 1461444820986553. <https://doi.org/10.1177/1461444820986553>
- [12] Donald A. Schön. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Temple Smith, London.
- [13] Susan Star and Karen Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research* 7 (March 1996), 111–134. <https://doi.org/10.1287/isre.7.1.111>
- [14] Susan Leigh Star. 1999. The Ethnography of Infrastructure. *American Behavioral Scientist* 43, 3 (Nov. 1999), 377–391. <https://doi.org/10.1177/00027649921955326>