

Claudio Castro Chaponan

May, 2026

Dismantling computers through observation and play

Thesis submitted to: the Department of Experimental Publishing, Piet Zwart Institute, Willem de Kooning Academy, in partial fulfillment of the requirements for the final examination for the degree of:

Master of Arts in Fine Art & Design: Experimental Publishing

Adviser: Marloes de Valk

Second Reader: Michael Murtaugh

Word count: 7380

Introduction

During the past two years I got more and more interested in researching alternative usages of technology. The computing industry of today still chases an idea of constant exponential growth, even if mass production of computer devices and components implies an ever-growing usage of resources that are limited, both from an energy and mineral point of view.

When talking about technology, we mostly focus on ideas of progress, paying attention to the latest devices available on the market and their great feats. But what we do not pay as much attention to is how many of these objects are constantly discarded prematurely soon after their release, in favor of a perpetual cycle of "innovation". Innovation has made it possible for us to have thousands of smaller, faster, better devices, but what does it mean to have more of these resources constantly available? Do we really need this much computational power to begin with?

I started thinking of these questions during the development of my practice, which consists in investigating alternative usages of everyday technology by using old hardware that is considered obsolete as a way to reflect and explore a technological scale that is defined independently, following intuition rather than externally imposed standardized practices.

This exploration started from my individual experience and ended up taking form in "The 100th time", a research project through which I developed a tech pedagogy rooted in curiosity, embracing the position of the amateur as a positive role, trying to show how can curiosity and play help us learn alternative usages of everyday technology.

While practicing my explorations I discovered my own method to engage with the everyday technology that surrounds us, revisiting common habits and changing them by learning new tools and documenting my progress.

In the following text I will retrace my personal journey through learning about hardware, software and pedagogy, highlighting my findings, which were technical in parts, but for the most part completely the opposite of technical.

In this journey, I embraced my amateur position, standing up for the fact that experimentation does not require expertise.

Dismantling hardware

Everything starts from failure

My first effective introduction to a different technological scale did not come from articles or books, It all started from a failure.

In 2023 I bought an up-to-date laptop with the latest processor, graphics card and other components, I was not pressured into buying this device by someone specifically, but I felt that this was the right move to make as at the time I was doing some 3D procedural animation work together with some installations during my Bachelor studies. After a few months of use, the computer started malfunctioning on multiple levels, both hardware and

software. These issues never got solved to this day, not even by the official repair center covered by warranty, which kept my computer for almost 40 days after I sent it to them. During the same time, I found an old 2011 netbook in my basement that was not working anymore, which was used by my family when I was a child. Disappointed by the recent events, I got curious by this device and started to ask myself a couple of questions, was it possible to repair it? Could I do that?

While sharing my frustrations with my mother, she pointed me to a local repairman that she saw a couple of times in the local news. His name is Massimiliano De Cinque and thanks to him I got my hands dirty and together we repaired my netbook.

Massimiliano lives in a small town near my home in Italy called Olgiate Olona, he is both passionate about IT and about sharing his knowledge, which he does in many ways, like collaborating with the local and neighboring municipalities and schools by offering training to local communities and by running a repair laboratory in his garage and in his backyard. Massimiliano says he makes his skills available to help others find their way into the technological landscape that we inhabit, with an interest to use his service as a way to contrast planned obsolescence (De Cinque, 2026).

The idea of planned obsolescence was introduced by Bernard London during the Great Depression, stating that the fact that consumers could use the product they owned for a long time, caused the economical crisis to last longer. He proposed to set an expiry date on products and that the government should take action on those who used theirs longer than the established period by charging a tax. This proposal never passed, but his idea lived longer than him, as it got later reused and implemented in the design of consumer products, decreasing their lifespan on purpose and creating an artificial need to purchase more (Hertz and Parikka, 2012). Planned obsolescence today is a contributor to a big environmental issue, as things like computers and smartphones are discarded more than they need to, which creates a lot of e-waste.

The netbook that I repaired with Massimiliano was not working just because one component, called the CMOS battery, was out of charge. This battery is as big as a coin (see yellow component in Figure 1) and it powers the CMOS chip, an important component that stores system information, like date and time, which are tracked even if your computer is turned off, that is needed to boot a computer. By simply replacing that, the computer was working again.

I have been inspired by Massimiliano's actions and motivations. I started learning the basics of repair and maintenance of computer devices myself, to research in first person what would it mean to be more mindful of discarded devices, trying to impose some limits on myself that later on formed my current methodology. It all started by the idea of not buying a computer anymore, and looking for old devices to "occupy" as a way to reuse prematurely dismissed resources. I started doing this without knowing where this would take me, I just knew I wanted to research an idea of using less technology.

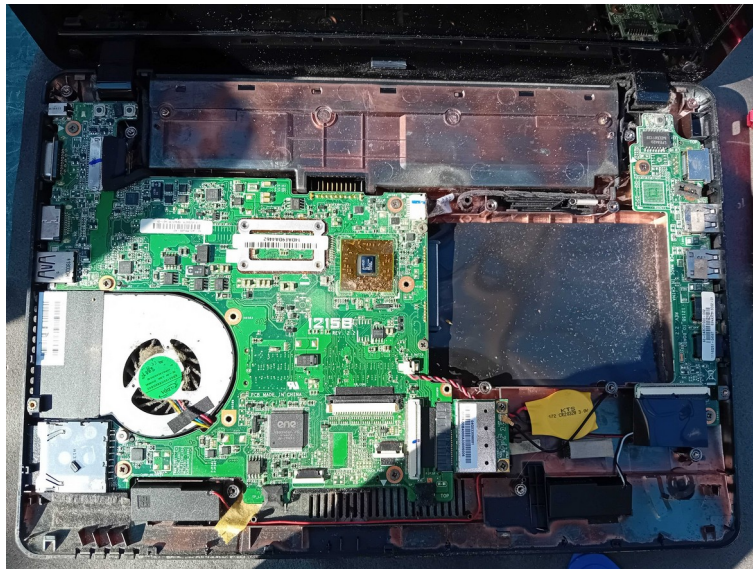


Figure 1: Top view of the netbook I have found, opened

Meeting Massimiliano introduced me to a new form of learning that involved a prevalence of doing things rather than reading extensively about them, like I was used to form my studies during my bachelor. I developed an interest to continue the practice of repair and in the months that followed I did some exercise on my own, like taking apart my recently fixed netbook entirely and rebuild it, hoping it would still work once I tried to turn it back on, I even got my hands on another older computer from my aunt, which I also used to practice by disassembling it entirely and putting it back together, but this time, I broke it.

Upon my arrival in The Netherlands I was surprised to see that there was also a shared interest in these topics from other people that are connected to the course like Rosa Schuurmans, who organizes weekly (un)repair cafes (010 kalender, 2026) which I attended when I arrived in Rotterdam as the atmosphere was similar to the one I felt when visiting Massimiliano's backyard laboratory.

I managed to meet more people interested in repair by sharing my experiences through the pictures I took of the devices I worked on, emphasizing that I was learning this new skill and that I was excited to do more, even though it was all new to me. Sharing my ongoing progress with others, created some momentum for this practice to continue as I ran into people that were also interested into getting their hands dirty and trying to do things themselves.

In most cases people needed some support to deal with maintenance issues, together with my flatmate and other friends I started taking care of things like cleaning a computer's insides and its ventilation system (see Figure 2) from dust, and changing the thermal paste of its processors graphics cards. These are not really malfunctions, they can be perceived as such if, for example, the computer would make a strange worrying noise from the clogged up vents or if it would reach high temperatures. Taking care of a buildup of dust or dried thermal paste can most times help to postpone or treat these sort of bigger issues.

While being a newcomer I made time to share my fresh knowledge with others, while still learning together with them, by researching solutions and tackling problems together, sharing both frustrations, when we repeatedly tried to open a computer that wouldn't budge, and satisfactions, when we would put everything back together without messing up.

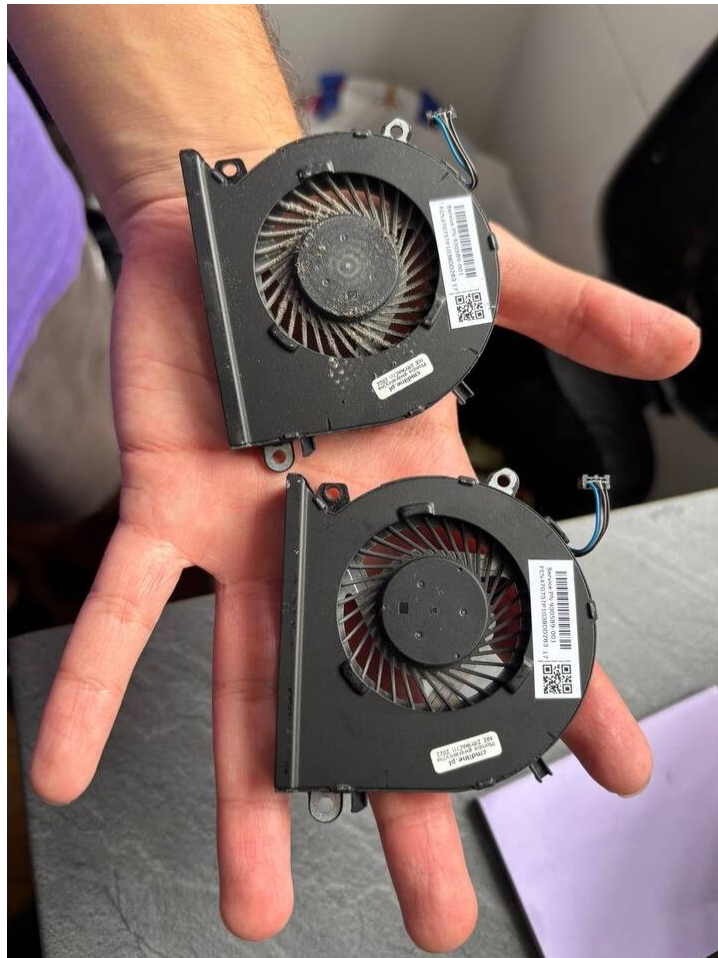


Figure 2: Computer fans before and after cleaning

Learning in public

As I worked on more devices, I realized I also wanted to expand my form of documentation, to try to find a way to share something more about the feelings I felt while working, and also gather general observations that came up from this practice. This is something that cannot usually be found inside of a guide or a manual, which is stripped from this type of information in order to condense a general solution to solve a given problem.

I got inspired by Anne-Laure Le Cunff, a PhD researched at the Institute of Psychiatry, Psychology & Neuroscience at King's College in London. Le Cunff keeps a personal “public notebook” online, where she shares unfiltered early thoughts on networked thinking, creativity, metacognition, and collective intelligence, which can then be expanded into longer forms elsewhere. (Le Cunff, 2022)

She wrote an entry about the power of learning in public, which, in contrast to learning and thinking privately out of fear of being judged by others, can be thought as a way to invite them to participate into a learning process, enhancing exchanges between peers. As she writes: “seeking validation should not be the goal of learning in public. Instead, the goal should be to tap into your network's collective intelligence to create constructive feedback loops.” (Le Cunff, 2020)

Being public about what I was doing helped me to meet other people and brought also an interesting narrative layer to the work we did together, like sharing stories behind the devices we wanted to repair. If I were sitting alone in my bedroom, this would have never happened.

So I chose to document this process in a way that would highlight more of the contextual information from the work done with others. By the end of the first year of my studies at XPUB I got more comfortable with writing inside of the space of the course's wiki¹ so I decided to start making a page there to collect my repairs and the notes connected to them. The information that I write inside is organized to share the model of the device repaired, the approximate date, to give an idea of when the repair happened, the status of the repair (if it is done or if it is on hold) and the category of the work done (like repair or maintenance).

I use the bottom of the page to collect some reflections that came up from my experience, which consists in simple observations.

Some example are the first time that I opening my own computer, which felt almost wrong as I was afraid of breaking something, or the feeling of opening someone else's device for the first time, which felt even more scary than working on my own as I felt more responsible of dealing with other people's computers.

When working with smartphones I noticed that opening up a one feels many times more delicate and scary than opening up a computer, because it is more complicated than twisting a few screws and removing them. Contemporary phones have their screen attached to the rest of the device with glue which needs to be heated in order to access the rest of the phone. If the work on the device (which could be replacing the battery) takes too long the glue might dry out and it might not stick anymore once you would try to put everything back in place. Another observation related to smartphones shares a frustration that comes from hardly finding the replacement parts of a component of older phones online (like a screen) and only finding replacements for prices that might be higher than buying that same device second hand (usually also with its share of defects)(Castro Chaponan, 2025).

I noted down how I felt about doing something in between the information I generally share of how I solved the problems I ran into (if I actually managed to solve them) trying to focus also on things other than technicalities, talking openly about my failures and the personal thoughts felt by me or the people I worked with

This was the first time I thought of a form of documentation that would speak more to me rather than a generic person who is looking simply for a solution, that highlighted the things I was giving importance to when practicing. By practicing over and over, I started to concretely imagine a different scale of computer technology. Making old devices work again years later, for example, is something that made me think about their resources, both material and computational. Technology is often described as something that constantly evolves and gets better, but since my journey into repair I started asking myself what does it mean to have all of this computational power available, and how much do I really need personally.

1 At XPUB, we share various forms of collective infrastructure, both physical and digital and the wiki is one of these. It is a self-hosted instance of the Mediawiki, the free software that is used by Wikipedia. Inside of the wiki students and tutors have user accounts, that allows us to write personal and public wiki pages that contain information about classes, notes about individual processes, and general organizational material like calendar entries.

The Chromebook repair

Out of all the repairs that I did, there is one that stood out the most to me. During March of 2025 I took part in a workshop organized by Boris Smeenk, an artist and tutor of the school's interaction Station², which, during that year, received a donation of Chromebooks from a company that collects discarded electronics.

A Chromebook is a laptop produced by Google in collaborations with several big manufacturers of computers - like Acer, Samsung, HP (see Figure 2) - that runs ChromeOS, a web-based operating system. These laptops have been introduced in 2011 and were designed mostly for online activities like accessing Google applications like Gmail, Google Drive, Google Docs etc. through the use of ChromeOS, as an interface to connect to these services (Pogue, 2012).



Figure 3: The Chromebooks we repaired during the workshop

They became popular in educational environments for their affordability (their price ranges from 150\$ to 300\$ depending on the device) compared to other laptops or tablets. In America, for example, in 2012 around 27,000 computers were leased or sold across 41 states, meaning hundreds of schools were using Chromebooks in their classrooms (Temple, 2012).

In 2023 though the educational institutions that became reliant on this cheap family of devices had to come to terms with the fact that they were not built to last, as several models stopped receiving software updates, leading to schools littered with e-waste and needing to replace their expired laptops with new ones (Seshadri, 2023).

2 The Willem De Kooning Academy defines a station as a workspace. Inside of the school there are different stations which are dedicated to accessing specific tools and get support to develop student projects. The Interaction station is one of these workspaces with tutors that offer help with projects that cover a wide range of technology.

Apparently Chromebooks have a set Auto Update Expiration date (AUE) which is visible inside of the Settings menu, under a dedicated section that reveals for how long software updates will continue which reads: "This device will get automatic updates until [specified date]."

In an official support page Google states that "older Chromebooks eventually stop receiving new software updates when hardware can no longer support the latest versions" (Google, no date). In the same support page Google then offers a couple of solutions if automatic software updates are not an option anymore: to either upgrade to a new Chromebook, linking a shop list page with the latest products, or to continue using the device knowing that security updates will not take place anymore. (Google, no date)

This was also the case for us, even if the hardware of the Chromebooks we worked on was still functional, because they were given an expiry date, they were dismissed by the schools they came from.

Based on my previous experience, I was expecting something similar to what I was used to, like replacing some parts or changing ChromeOS for another operating system, but it was not as simple as that this time. It was by far the most complex repair I've ever done, the workshop was organized in four sessions in order to evenly organize the work that we needed to do, as it involved both hardware and software tinkering and a good dose of luck.

The original source of this repair method was the Mr. Chromebox docs page, which is a project run by a developer which documented extensive information about ChromeOS, including methods to change the Chromebook operating system, which is a way to recover the old Chromebooks models that stopped receiving software updates (Mr. Chromebox, 2024a).

Mr. Chromebox's website is written with dense technical information concerning all the possible Chromebooks models, so for the workshop we crossed referenced the official documentation and the one Boris made, condensing the general main steps we needed to do (Smeenk, 2025)

In this back and forth throughout the sessions, we had to remove various types of locks from the Chromebooks. These locks, which are referred as "Write Protection", were both physical, in the form of a screw on the motherboard that we had to remove, and abstract, in the form of a custom script we had to download that disabled the software lock (Mr. Chromebox, 2024b).

Throughout this lengthy process we ran across several alerts that warned us of the dangers of what we were doing, but after some work we managed to install another operating system, taking ChromeOS out and installing Linux by inserting a USB stick with an installation file, making the Chromebooks regular computers that can still be used.

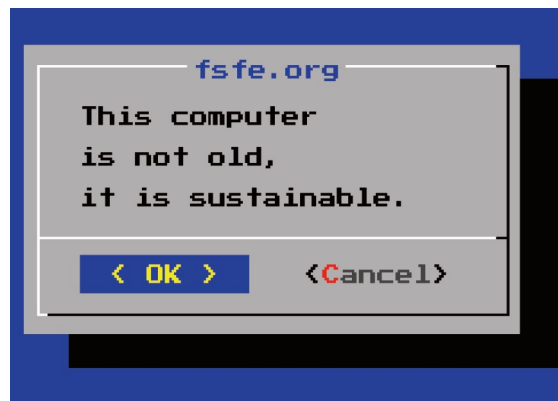


Figure 4: My computer is not old, sticker design by Lisa "Mullana" Schmidt

I could not believe that this much work was needed just to make these computers usable again and that they remained unused for who knows how long until they ended up on our tables at the Interaction Station.

This is just one out of many examples of planned obsolescence in software. As noted by researcher Lea Mosesso “Software should theoretically not suffer from natural wear as does hardware, being able to run endlessly once installed. However, in practice, we see a growing number of empirical examples where high performance requirements, cancelled support and update services, malfunctioning software, new social usage or technical shifts, and other software related problems speed up the obsolescence of otherwise functioning devices.”(Mosesso et. al., 2023, p. 2).

Another more recent example is Microsoft's decision to drop support for Windows 10 in favor for its other operating system Windows 11 in October 2025. "Upgrading" to the newer version of Windows is possible only under specific minimum hardware requirements, if these are not met, users needed to either enroll in an extended support program, that would last until October 2026 or to "simply" buy a new computer.(Microsoft 2025)

To counter Microsoft's plans, an international campaign was launched on May 28th 2025 called “End Of 10”, offering an online community of support in the form of a networked list of repair cafes, independent shops and organizations that can help with making the transition to a new operating system, stating that “the cheapest and most environmentally-friendly computer is the one you already own” (The end (Of windows 10) is nigh!, 2025).

After the repair workshop, we were told we could keep the computers we repaired if we wanted to. I initially was not convince since these computer were clearly designed for replacement, which we clearly saw when we looked inside of them. All of their components are soldered to the motherboard, so if any of them (like the main storage or the RAM) fails, it means that it cannot be taken out and simply replaced (see Figure 5).

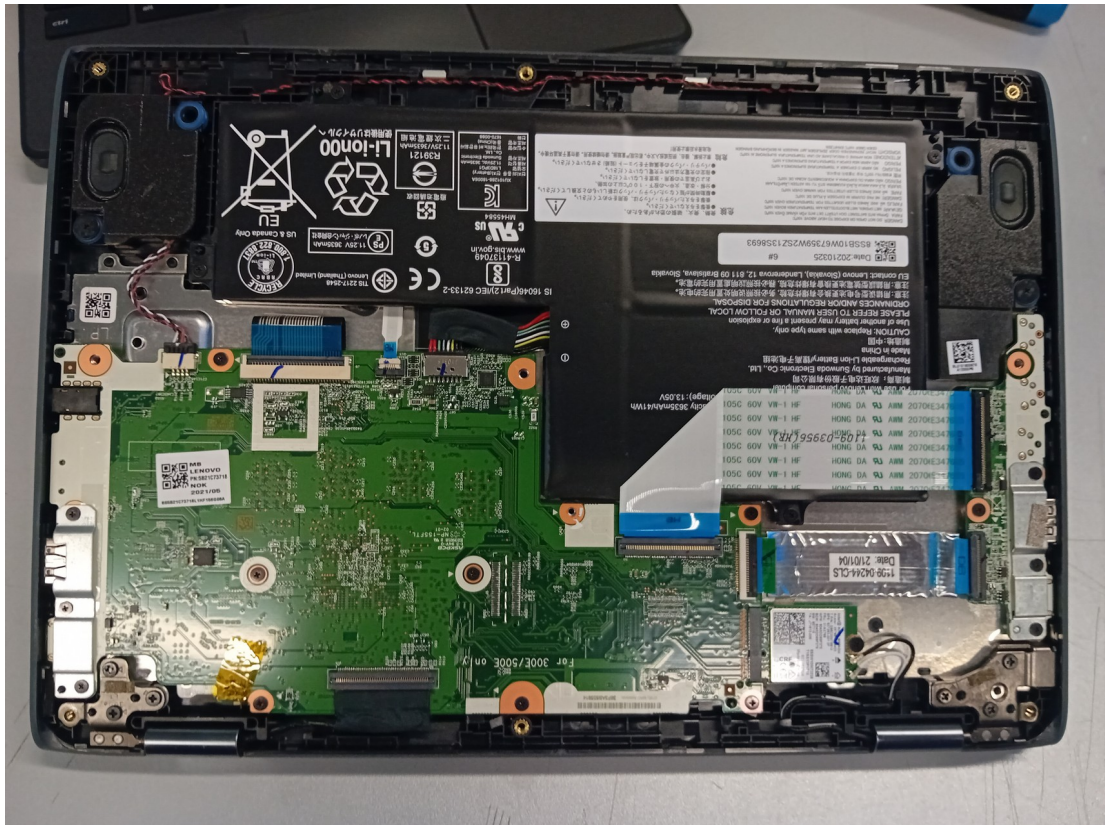


Figure 5: A top view of an opened Chromebook.

Despite these observations, I viewed its limitations as a possibility to bring practical my research further. I decided to reuse this 10 year old laptop as my primary working device, both during my studies and in my own personal time, rediscovering my daily usage of a computer within the hardware limitations of the Chromebook³.

Dismantling software

Old hardware, old software

As I started using the Chromebook I repaired as my main device, I noticed how having limited computational resources impacted my usual habits of working with a computer.

Opening the browser took more time, watching a video online was almost impossible, and in general working with a graphical interface on any program was more cumbersome, sometimes even leading to crashes. Because I was stubbornly interested in reusing this computer, I started looking at alternative software programs, in order to learn different ways to do the same operations (ranging from making simple graphics, making presentations and generally using the device for daily needs) I would do on a more powerful computer, but with limited available resources.

³ Since the Chromebooks were designed for accessing data online, it has little to no storage on the computer itself: 16 GB total, which after installing a new operating system with a graphical desktop environment and some programs became 2 GB for me.

While looking for alternative tools, I ran into a blog post from a system administrator whose nickname is yctct, that talked about old software highlighting how the same wasteful cycle of technological innovation that I started perceiving for computer devices takes place also in software. New applications are pushed in spite of old ones all the time, even if the latest tools might also be the first to be forgotten (yctct, 2024). In their post, yctc lists the programs they use, and by specifying the date of their release, they show that some of them are from the 70's.

Old software, like old hardware, is not necessarily obsolete, in fact most times it still works and it is still used in the background of a lot of today's services, an example being cURL, a program written by a software developer named Daniel Stenberg in 1996, which is used for communication worldwide allowing for data transfer over the internet (IVA's Gold Medal 2025, 2025).

Given that I started using old hardware, I started looking into old software as well, and I did so by reading a lot of new forms of technical documentation, like software manuals, forum threads and technical cheat sheets.

Reading software documentation

When I was learning computer repair, I went through forms of technical documentation that were mostly image based (videos or articles with pictures of the operations I needed to do). In those cases, the barrier I often ran into as a newcomer was that I needed to have the correct tools at hand for the specific situation (like having the correct screwdriver bits), but when I started reading software documentation, I found myself way more under prepared and confused.

As I installed Linux on my Chromebook, I started learning about it and free software tools I gneral. Since my device has very limited resources, what I was curious about and continued researching were mostly text-based tools, as I could not simply install many programs with a graphical user interface on my computer, I had to open the terminal and find equivalents⁴.

It was interesting to learn to interact with a computer through text, but it was definitely not intuitive. This is mostly due to the fact that these text-based tools, like the terminal itself, come from a different era of computing. Until the 70s, computers were something available only to a few, they were not in households but in research labs, bigger in size (as big as entire rooms) and they were operated much differently than today without the use of icons and images, so the way that the tools that come from that time are documented follows mostly conventions that today are very far from common computer knowledge.

As I do not have a background in computer science, I faced the fact that even if this information is publicly available, like in the case of manuals and documentation of free and open source software, this does not necessarily mean that this is information is for everyone to read or understand, as the main difference I immediately felt was the amount of technical jargon used, mixed with secret implications. Examples of this can be not writing the full context behind a tool, like who made it and for who it is intended, or not specifying clear

⁴ The terminal is a program that represents the space of the computer through a text based interface.

ordered steps to achieve a goal, which created multiple walls of incomprehension for me at first.

After some time, I learned how to decipher the language used in software manuals, by trying all of the examples they proposed first hand. I was obsessively executing all of these so called commands, failing a lot of them and repeatedly trying them again and again from scratch, causing all possible error messages to appear on the screen.

While looping into these exercises, I started to keep a lot of notes to help myself, especially when I discovered or understood something that I though was missing in the documentation I was reading. By doing all of this technical learning, I found that repetition was something I would do with a mindful presence, reflecting while making, especially on why I was doing these things in the first place. From this reflections, I realized that I was not learning all of these technicalities just for the sake of it, but I had a specific intent in mind, which was to attempt to keep in use a specific device I repaired.

During the process, though, I was loosing this initial focus and I was getting lost in the documentation: the more I learned about software, the more I felt a weird pressure that I had to continue learning, somehow feeling more insecure than when I started. This was more clear once I moved from reading software manuals to reading forum posts on platforms such as Stack Overflow. I was caught between the online realms of various communities that were very opinionated about computing, to the extent of having fired up arguments that revolve about which operating system to install, which software to run, which version etc. and I saw how these opinions formed some sort of status symbol. Something was off.

I slowly realized the implications of choosing an old device for daily use: to be able to use less computational power, I needed to deal with OS updates, package managers, manual installation of software etc.

Alternative approaches

During the course of XPUB, we experimented with many of the tools I was running into (such as the shell and programming languages like python and bash).

Because an important part of the course is to take into account "the implications of the technologies and tools we use, how the choices have an impact socially, environmentally, politically"(xpub, 2026) which in practice is done in many ways, most notably by using free software⁵ for both classes and for self-hosting our own online infrastructure⁶, which got introduced to us at the start of the course.

This infrastructure consists of various websites, applications such as the Wiki and Etherpads⁷, and small Raspberry Pi servers given to students for the duration of their studies. We used them as a way to interface within the overall infrastructure of XPUB and outside of it, for example by using it as a publishing surface by making and hosting our own websites. The environment of the course was of great support to go through these feelings of insecurity and frustration that I felt.

As the tools we used and discovered throughout the course come from contexts that are far away from us, core tutors acknowledge the hardships that might come from not understanding, and they both introduce us to the value of listening to these feelings (Berends, 2025) showing us that there's a multitude of approaches that can be used outside of the

5 A piece of software is free when its code is published publicly, which grants people to study it, use it, modify it and redistribute it freely, contrary to proprietary software, that does not offer the same possibilities.

6 Self hosting consists in creating and maintaining servers to self host websites and online applications.

7 A collaborative writing application.

standardized ones, which is something I would have hardly discovered if I continued reading documentation on my own.

Guest tutors also played a big role in this for me, one example is Pippin Barr, an experimental game maker and teacher at the Concordia University in Montréal, who came to visit us in 2025. During the class we had with him we discussed together about documentation of our own artistic practices. He showed us some examples of his, as a game maker, he writes games with code and publishes his work on GitHub, a platform used to share and collaborate on software projects, but what I was very happy to see when he showed us projects like “It is as if you were on your phone”, a game about being on our phones all the time, was how he tweaks the standard format of code documentation of GitHub, which is a platform mostly used by developers. Apart from the source code of his work and the basic description of the project, he includes more than just instructions of how to use his software. He uploads many images of scanned doodles, journal entries, backstories in the form of anecdotes and other notes that over form his personal progress and behind the scenes. (Barr, 2025)

Another great inspiration for me was Doriane Timmermans, a web developer, artist and educator from Belgium. We met Doriane during the second trimester of our first year at XPUB, she joined the course as an external tutor and editor of XPUB’s Special Issue number 26⁸ that was themed around her research project called Declarations, which she describes as “an ongoing trans-disciplinary artistic research project into the poetic materiality of the CSS web-standard and its echoes on design and artistic practices.” (Timmermans, 2026)

The way Doriane shared her research on CSS, which is a markup language used to style and and present a web page (Wikipedia, 2026) was very different from the technical manual that describes the functionality of this tool. She put the focus on the use of metaphors, highlighting how the language properties of this tools have a poetical value, setting aside the rules of the software manual. During the trimester we spent working with her we approached web making as a craft, instead of constraining ourselves in the grids of traditional web design, we experimented with the medium following our own interests.

The modes and tones she used during the time we spent together were similar to the ones of a close friend sharing knowledge they are passionate about, something that is recurring in her practice and that takes full form in her HowSpots: “inclusive and welcoming pedagogical moments that take place in someone’s home to share situated knowledge about everyday technology for everyone (especially your friends) and claim (some form of) digital autonomy .”(Timmermans, 2025)

I joined one session she organized during August of 2025 on the topic of archiving things we care for on the web. We met at our friend Chae’s house and started the session by introducing ourselves to the group of people that came, sharing our experiences with archiving and why we joined this session. Even that simple introduction started many interesting threads, everyone’s personal anecdotes enriched the topic a lot, and the whole thing in the end was more about sharing our experiences and struggles that come by dealing with technology rather than finding technical solutions. The importance I found in this form of sharing, was the possibility to exit the “logical” frame of how people make use of technology opening up space for a different discussion.

8 At XPUB each Special Issue (SI) consists in a three months long thematic seminar on a given topic curated by an editor. Each SI ends with a publication, which can take various experimental forms.

Forming my own tech pedagogy

Learning, documentation and play

Precisely because my background is from another field, the one of arts and humanities, I did not need to feel intimidated by software best practices when dealing with technology, yet at the beginning it was really hard to understand.

When I was coming back to reference technical documentation after some months, I started to notice that I wanted to move away from it, as in the end I was starting to use something else to make sense of what I was learning. Inspired by Doriane's influence I was able to accept where I come from and own my amateur position, using it as a strength and positive element of my own work, I was able to go back at following my intuition.

It finally came a time when I wanted to share my discoveries with others, but I did not want to lean on the technicalities of the things I learned as my main focus, so I moved away from tutorials and guides and reflected on my influences, such as Massimiliano and Doriane.

Massimiliano introduced me to a type of practical learning that jumps straight into action, without relying too much on absorbing theoretical notions before hand. I used to think of learning as a solitary experience before meeting him. In the past I would first spend some time getting familiar with theory before and only after "getting it" I would try to apply what I learned through practice, trying to think of any possible outcome before, but thinking this way created a sort of self imposed threshold, that determined when I was or not allowed to take some action. Massimiliano's teachings were something I carried with me when continuing to practice repair and also when I started reusing the Chromebook I fixed. Doriane showed me a way to put my own sensibility in the foreground to better relate to something completely distant from my world, leaving room for listening to my own intuition and using it as a way finder to conduct my explorations.

Both Doriane and Massimiliano share their knowledge socially, paying great attention to the situated part of their work, which takes place in a local context they create consisting in being together with others and listening what happens relationally between the people that are physically present at the moment of their teachings. These threads lead me to refer to the sphere of pedagogy, in which I found an interesting learning theory called the head, heart and hands learning. This method, introduced by Swiss pedagogue Johann Heimrich Pestalozzi during the 18th century, is about the belief that when we learn things, we use our head, heart and hands, the head being a metaphor for the cognitive domain of learning, the heart the affective one, and the hands the psychomotor. Traditional education gives most attention to the head, the rational part of our learning, which consists in storing information, remembering it and synthesizing it, but taking into account the other two domains of learning, which are represented by our feelings and our bodies, is something that allow for all the these three domains to unite into what is considered to be holistic learning (Gazibara, 2020).

The importance of play

Re-reading my notes revealed that in the loop of reading and writing, not only I was forming my own documentation, similarly to what happened with my repair and maintenance wiki page, but also my own tech pedagogy, that would prioritize curiosity when navigating technology.

Throughout all of the repetition what really got me through the process of going through the strict constructs of software was humor and playfulness, something that I initially developed in my notes as a way to make sense of what I was going through. Once I grasped this, I tried putting play in the foreground and used to write my personal documentation. As I spent much time inside of the terminal, I made a zine that voiced and valued the experience of getting lost inside of it (see Figure 6), adding a general narrative form that would make room for my puzzling feelings, unsuccess and other personal elements like drawings of myself in a comic format that accompanied the text, adding my voice in the form of comments.



Figure 6: "Lost in terminal" personal documentation.

Play is an activity that is given importance to only when we are growing up during our childhood. After that, it feels like with time it loses its key role when entering adulthood, starting to fall in the background, becoming something considered secondary or not necessary at all.

But if we consider play as a methodology for learning, we can realize how much we have passively learned through it. In our childhood we commonly get introduced to many concepts by playing, by having moments of "make believe" we take up roles that we are not familiar to with fearless engagement. In these moments, we explore how it could be to be a builder, a scientist or a dinosaur by using our imagination to enact them. On the contrary, adult life is divided into much more specific roles and tasks, which ends up in feeling like you cannot

approach the unknown with the same ease, we hardly think that we can do x if we do not have some prior experience with it, forgetting that the knowledge we have today is something that we did not know of at some point.

I rediscovered play as an interesting methodology for learning, and I use it as an element that supports a positive feeling about exploring something new, passing through playfulness and creating scenarios where imagination can be used to discover something and also imagining how it could be (as in how it could work differently than it really does).

A key part of playing is the set of rules that are in place: some conditions that allow a player to move towards their goals. Playing by the rules in the context of using text-based programs can mean, respecting a fixed syntax, paying attention to spelling mistakes, using special characters to delimit operations, like adding a “-” to use a specific option of a program, or writing text in a specific order, to use the program and its available options. Not respecting these rules will result in different error messages depending on what went wrong. Playing by the rules will not return an error and complete an operation, like copying or opening a file. When I thought of rules in a broader sense, I started asking myself how could the same rules that in my experience defined restrictions could be presented in a way that would turn them around, transforming them into possibilities within the limits of their workings.

Instead of focusing on sharing my learning from theoretical concepts behind them, I wanted to develop made up scenarios that would try to keep into account emotions and facilitating their expression be them positive, like satisfaction, or negative, like frustration. These scenarios would be something I set up openly, to make sure that I would not necessarily force a determined one on others, hoping to leave some room for interpretation and play.

I was interested in sharing the stories behind my experiences to evoke emotions in a social context, inviting others to take a look and experience my research.

Lost in terminal

I applied my methodology of play when I turned my “Lost in Terminal” zine into a workshop about making use of terminal commands to imagine a collective shared space (Castro Chaponan, 2026). I hosted the first iteration of this workshop on November 2025 for my peers at XPUB, making use of our shared server that we used during our studies.

I asked my classmates to think of it beyond its functional use of a shared resource, something closer to our studio or the wiki for example.

During the session we gathered into a room and connected to our server from our own devices, keeping in mind the metaphor of the server as a space. I also made a connection between files and folders, which in this case would represent objects and rooms. After a brief introduction of the technical rules needed to move around inside of the server (which we interact with using the terminal) we shifted our attention towards the act of making rooms and objects, that would make up spaces where we could either use to relax together, or where we spend time gossiping with each other.

Hinting at the server as a hangout space was a metaphor that worked well, I saw my peers having fun in creating absurd objects like, “lavalamp-from-kruidvat” and “huge-coffe-spill”

filling many rooms with them (see Figure 7), they also had a fun time finding out how they could steal objects from each other's rooms.

They faced the confusion of the terminal by playing inside of the constraints and the rules of the syntax of the shell and even experienced the rules of ownership of the files and folders they created, working around the permission to modify or move them.

I used the technical know-how as pretext and a base for something else, encouraging others to make reference to their imaginations and feelings using individual knowledge to make our collectively shared space. It was more like a session of fictional writing than a tutorial about the terminal.

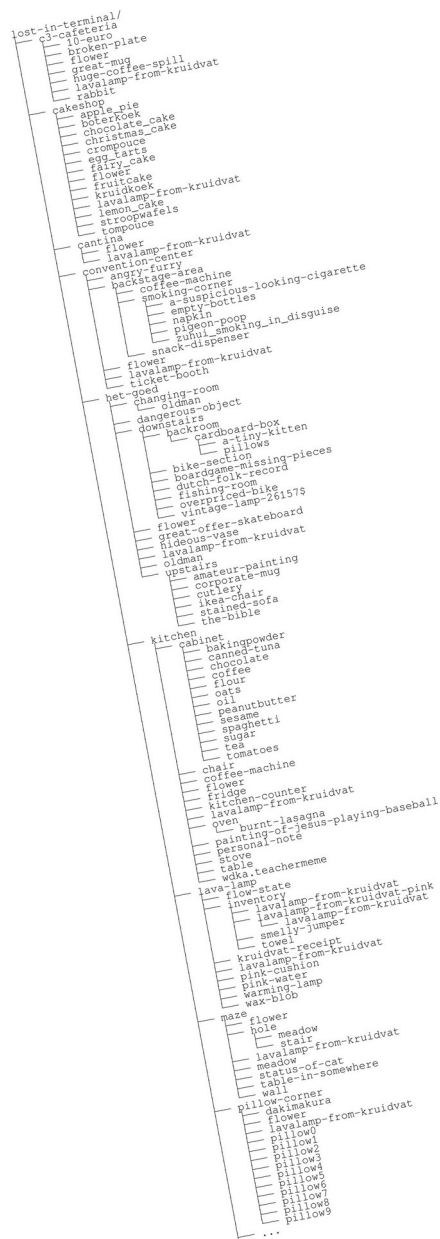


Figure 7 : "Lost in terminal" workshop documentation.

I chose the terminal because I know that it is a tool that can be very confronting, personally, the first time I saw it I felt exactly how I felt the first time I opened a computer, it was scary. From my peers I got a general feedback that it was nice to move the focus away from doing things the “right way” (being in the correct folder and executing the right commands), towards doing them the way we saw more fit, to embrace the feeling of being lost and not knowing, but also not being so scared of it. This had a positive effect, and turned the space of the terminal one of possibility, rather than restrictions. Continuing forward, I see myself repeating this workshop in different contexts, with different audiences, to explore what iteration in different groups might bring up. I also will use the 100th time methodology as something that I think can be applied to various other activities, such as the re-purposing of discarded devices.

Conclusion

Practicing repair and reuse of electronic devices is something that brought me to test my ideas on technology and helped me define a different technological scale other than the big tech one we are used to see so much, one that takes shape by curiosity and hands first experience, rather than standardized best practices.

With the methodology I formed with “the 100th time” I invite others to look beyond the use of technology as dictated by industry standards, to be skeptical of the myth of unlimited resources and to reclaim tech through play and experimentation.

I would like to encourage an exploration of what is already available today, as a way to learn about how much space technology already takes into our lives and how much our actions can help discover ways in which we redefine autonomously the amount that we want to keep and the one we want to disregard.

Images

Figure 1: Top view of the netbook I have found, opened.

Figure 2: Computer fans before and after cleaning.

Figure 3: The Chromebooks we repaired during the workshop.

Figure 4: Schmidt, L. (2023) My computer is not old. Available at: <https://fsfe.org/contribute/spreadtheword.html> [Accessed March 12 2026]

Figure 5: A top view of an opened Chromebook.

Figure 6: “Lost in terminal” personal documentation.

Figure 7: “Lost in terminal” workshop documentation.

References

010 kalender (2026). Available at:

<https://mastodon.social/redirect/accounts/113305026247163073> [Accessed April 29 2026].

Barr, P. (2025) it-is-as-if-you-were-on-your-phone. Available at:

<https://github.com/pippinbarr/it-is-as-if-you-were-on-your-phone> [Accessed April 2 2026].

Berends, M. (2025), Prototyping 2024/2025 intro Available at:

https://pzwiki.wdka.nl/mediadesign/Prototyping_2024/2025_intro [Accessed April 9 2026].

Castro Chaponan, C. (2025) The maintenance and repair page. Available at:

https://pzwiki.wdka.nl/mediadesign/User:C3/The_maintenance_and_repair_page [Accessed April 5 2026].

Castro Chaponan, C. (2026) Lost in terminal. Available at: <https://git.xpub.nl/c3/workshop-lost-in-terminal> [Accessed April 10 2026].

De Cinque, M. (2026) Digitalsherpa. Available at: <https://www.digitalsherpa.it/chi-sono/> [Accessed April 6 2026].

Gazibara, S. (2020) “Head, Heart and Hands Learning

- A challenge for contemporary education” *Journal of Education Culture and Society*, pp. 71-82. Available at: <https://doi.org/10.15503/jecs20131-71-82>

Google (no date) Use or upgrade an old Chromebook. Available at:

<https://support.google.com/chromebook/answer/7052113> [Accessed April 29 2026].

Hertz, G. and Parikka, J. (2012) “Zombie media: circuit bending media archaeology into an art method,” *Leonardo*, 45(5), pp. 424–430. Available at:

https://doi.org/10.1162/LEON_a_00438

IVA’s Gold Medal 2025 (2025) Daniel Stenberg – the architect behind a global

communications backbone. Available at: <https://www.iva.se/en/published/ivas-gold-medal-2025-daniel-stenberg--the-architect-behind-a-global-communications-backbone/> [Accessed: April 1 2026].

Temple, J. (2012) Google's Chromebooks making big school push. Available at: <https://web.archive.org/web/20120127024806/http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2012/01/25/BUF61MTEAP.DTL> [Accessed April 20 2026].

Le Cuniff, A. (2022) About, Mental Nodes. Available at: <https://www.mentalnodes.com/about> [Accessed April 10 2026].

Le Cuniff, A. (2020) "The Only Way to Learn in Public Is to Build in Public", Mental Nodes. Available at <https://www.mentalnodes.com/> [Accessed April 5 2026].

Microsoft (2025) End of support for Windows 10, Windows 8.1, and Windows 7. Available at: <https://www.microsoft.com/en-us/windows/end-of-support> [Accessed April 19 2026].

Mosesso, L., Maudet, N., Nano, E., Thibault, T. and Tabard, A. (2023) "Obsolescence paths: living with aging devices," *2023 International Conference on ICT for Sustainability (ICT4S)*, pp. 13–23. Available at: <https://doi.org/10.1109/ICT4S58814.2023.00011>.

Mr. Chromebox (2024b) Firmware write protection on ChromeOS devices. Available at: <https://docs.mrchromebox.tech/docs/firmware/wp/> [Accessed April 28 2026].

Mr. Chromebox (2024a) mrchromebox.tech. Available at: <https://docs.mrchromebox.tech/> [Accessed April 7 2026].

Pogue, D. (2012) Laptop buyers should pay some attention to the chromebook, Pogue's Posts Blog. Available at: <https://archive.nytimes.com/pogue.blogs.nytimes.com/2012/11/29/laptop-buyers-should-pay-some-attention-to-the-chromebook/> [Accessed April 28 2026].

Seshadri, M. (2023) Chromebook software expirations leaving schools without laptops, EdSource. Available at: <https://edsource.org/updates/chromebook-software-expirations-leaving-schools-without-laptops> [Accessed April 20 2026].

Smeenk, B. (2025) Repurposing Chromebooks. Available at: https://wiki.ias.wdka.nl/wiki/Repurposing_Chromebooks [Accessed April 6 2026].

The end (Of windows 10) is nigh! (2025) End of 10. Available at: <https://endof10.org/press/2025-05-28-press-release/> [Accessed April 12 2026].

Timmermans, D. (2026) Declarations. Available at: <https://declarations.style/> [Accessed April 10 2026].

Timmermans, D. (2026) HowSpot. Available at: <https://howspot.ungual.digital> [Accessed April 5 2026].

xpub (2026) Available at: <https://xpub.nl/> [Accessed April 29 2026].

yctct (2024) Use old programs. Available at: <https://yctct.com/old> [Accessed: September 5 2025].