ow—Labour in a Post-work, Posthuman society

André Fincato, May 2016.

Outline

Chapter #0, Introduction

What does it mean to work, when work is no longer a necessary human activity to make a living?

At the same time of a resurgence, in the last period, of an interest in post-work societies, driven by Universal Basic Income and full automatisation of labour, there is, in parallel, a more systematic proliferation of tracking processes happening on the internet, out of which web platforms and companies extract surplus value from users' activities for their own benefits.

While the two phenomena might not have any relation with each other, a question I'd like to pose is: if thanks to a Universal Basic Income, ideally I won't have to worry about covering the basic needs to set up not a decent, but a good-enough life [1], how would this affect the situation on the internet? Would a post-work society be mirrored also in this space?

The transition I am interested in exploring, is the passage from the post-fordist worker to the internet user, which began in last decade: while the former label has been used to describe a set of employment positions (ranging from care workers to designers, journalists, to call centre employees) that constitute, for the most part, the group of precarious, part-time, project-based jobs; with the latter subject, the internet user, I identify the new iteration of worker part of the capitalist society we live in. Both subjects have in common the increase of mental and or affective labour as determining factors in the type of work performed: whereas in the case of the post-fordist worker, the required amount of this labour is specific to the kind of job performed (e.g. maximum level for a care worker, less demanding for an academic), the internet user is a subject ontologically defined, in its very formation, by a mix of affective and mental tasks. Unlike the post-fordist worker, an internet user of the commercial internet is almost always asked to pour him-herself in the platforms he she makes use of:

> Social media trap us in a tautological loop, in which we express ourselves to be ourselves to express ourselves, trying to claim better attention shares from the people we are ostensibly "connecting" with. (…) It doesn't matter what we say, or if we came up with it, when all that matters is the level of response. In this system, we don't express our true self in search of attention and confirmation; instead attention posits the true self as a node in a dynamic network, and the more connections that run through it, the more complete and "expressed" that self is.
>
> Rob Horning, Preemptive Personalization, The New Inquiry, 2014.

Which forms do labour take on the internet, from a user perspective? Though not the only one, the most pervasive is the technique adopted by platforms that most of us, users of the commercial internet, check everyday (facebook, gmail, instagram, whatsapp, twitter and so forth), through which they collect informations we share with other users, in order to sell them to advertisers and present them back to us in the form of tailored ads. [2]

In dealing with the multiform action of tracking, what I want to explore is a platform called GitHub and the tool upon which it is built, `git`: tracking, in this case, doesn't happen as a nonaware and malicious background process, but rather as a deliberate and voluntary activity part of the application software workflow.

GitHub (github.com) is a website and a platform where users can share, contribute and download code-based projects: a ‘social network’ for people who use `git` in their workflow. `git` is a version control system for text-based files, which is, a software that, once installed, tracks every change a user makes on one or a set of files, within one or more folders she he decided to look after. This is useful to keep track of the work done so far, especially when it is fragmented in a team, while also going back and forth within the project history, if something went wrong or a user needs a specific file now deleted.

Whereas other internet-based platforms, like Über and Amazon M Turk, are explicit in their intents to be a job-offer for someone looking for work, github.com keeps itself in the grey area of being for both personal and professional use.

As much as a programmer working at a company can use it for his her job, a weekend website maker can use it as well for the same purposes. Being `git` only an added element to one’s workflow, the reason for creating an account on github.com are highly subjectives, and totally in line with the spirit of the creative post-fordist worker (where famously ‘work is play and play is work’). As a user, the personal takes over the sphere of the professional, transforming both into one smooth space.

In the act of imagining an expansion in the usage of `git` and GitHub as a model to organise labour on the internet, I wonder how the possible effects of the above cited post-work, Universal Basic Income-based society would effect this sphere of existence.

By post-work society, I refer to a cultural, social and economical shift in the understanding of labor, where human beings are freed of the need to provide themselves with basic material needs, hence from the action of sustaining themselves. A Universal Basic Income, implemented at a global level and regardless of citizenship status (Nick Srnicek and Alex Williams, Inventing the Future, 2015, pp. 119), would provide the means to start this transformation, being it part of the four pillars outlined by Srnicek and Williams to move beyond capitalism [3]. Post, then, refers to the act of moving away from the action of work as the defining activity of one’s life (both in the form of making one’s life materially possible, as well as fulfilling one’s needs of meaning and direction in such life). While in books such as ‘Inventing the Future’ (Nick Srnicek and Alex Williams, 2015), there seems to be a total automatisation of labour, and in particular of what they refer to as ‘shit job’ [4], what they actually mean is a diminution of tasks human beings have to perform within a job, rather than the full replacement of it by machines [5]. In short, they reinforce the thesis of a more tight collaboration between human beings and machines, while in the process of moving toward a post-capitalist society [6].

The reframing of the importance toward the activity of work as a life-meaning vector, also includes the questioning of the driving force through which «each thing, as far as it can by its own power, strives to persevere in its being» (Frédéric Lordon quoting Benedictus de Spinoza in Ethics, part III, Willing Slaves of Capital, 2014, pp. 14). This is the concept of conatus expressed by Benedictus de Spinoza in his book Ethics, in this text understood as both a force present in each thing constituting the world, which permits them to survive and let them being what they are; but also, since their strive to persevere in their being, conatus as a drive to foster a process of self-iteration and of always-becoming over time. Framed in this way, how can the spinozian conatus be applied to the internet user?

The separation between performing work to sustain one's life, and working for a different set of reasons and values, becomes central in analysing the iteration from post-fordist worker to internet user. This separation of intentions, indeed, is already present on the internet nowadays: most of users' interactions and value-production are made for free, be it because part of invisible value-extraction mechanisms (e.g. social networks); or as a genuine interest in contributing to a common good (for example wikipedia). With the advent of an hypothetic post-work society, this shift might get stronger.

The other element defining the environment of the project I am working on, is the concept of the posthuman turn. As a set of theories describing the decentering of the human being as measure of all things in the world, the definition of user given by Benjamin Bratton (Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 287) is of significant importance: a user is anything able to produce data and exchange it, through the help of interfaces, with other users. This definition appears fundamental when talking about labour on the internet between users: the human user is only one of the possible entities this position can be inhabited by, opening it up to users with animal, plant, machinic, mineral, algorithmic (and so forth) natures (ibid, pp. 272–273).

Data assumes the role of raw material through which performing communicative acts between users of different natures: finding an interface to get in contact with other users becomes essential. Tracking, defined as a process of keeping tab of what happened and what is happening, can be used to build new kinds of these interfaces. Furthermore, the action of tracking on `git` and github.com, traverses both the sphere of ethics and labour: what are the implications between users, when performing this action, and how does this affect one's work? Overall, how does tracking organise such activities?

Chapter #1, User

From the post-fordist worker to the internet user

There is still a strong desire amongst freelance workers able to spur them to take a solo, independent career and, in doing so, avoiding to be under a boss.

Only for then being under a variety of bosses, being that a company, a client, they themselves even, and so forth.

Post-fordism arose in the early 1970s as a transformation of the current production model of the time, the fordist method applied to the factory, whose pillars were crumbling down due to factors such as:

> a) diminished profitability of production activities due to labour demands and technical restrictions (problems with managing longer and longer production chains); b) globalisation of economic currents disrupting the management of national economies; c) growing share of social transfers in public spending that propels the inflationary spiral and fosters distributive conflict; d) differentiation of consumption patterns demanding a broader range of supplied use-values.
>
> Igor Pribac, Post-Fordism–a Contextualisation, from Gal Kirn, Post-Fordism and Its Discontents, 2010, pp. 29.

The above are the outlined theories developed by the Regulation School, whose most prominent voice was Michel Aglietta, when during the 1970s, the international oil crisis, amongst other things, was slowing down the capitalist regime of production.

There were two more schools of thoughts, contemporary of the Regulation School, that sketched out other hypotheses in relation to what has began to be called, from the 1970s, post-fordism: the Neo-Schumpeterians and the Flexible Specialisation approach. Whereas the first was confident that big technological advancements were enough reasons to transform whole societies (religiously finding hope and building faith out of technology), the latter stressed the co-existence and opposition of both mass production and the rise of the specialised one as the means of the undergoing mutations in the economy of the time. For the Neo-Schumpeterians, a lot of agency was given to technological changes, assuming that «socio-political institutional frameworks depend on [these changes]» [7]. The 'Flexible Specialisation' hypothesis, instead, found in the shift of consumers' desires for a diverse range of products, and consequently a transition from mass-industrial production to, more and more, small, highly specialised and flexible firms, the reasons originating the undergoing transformation in the market at the time. Unlike the Regulation School's four core argumentations reported above though, both Neo-Schumpeterians and Flexible Specialisation's theories visibly lacked of «the economic-social-political core of the transitional changes» [8] at the basis of their postulations. Either they saw technology or socio-economic factors as the only driving vectors of change, instead of taking into account them both, and including politics as well.

Fresh of the free liberation of desires coming from 1968, and together with the application, for the first time, of neoliberal politics, post-fordism repositioned in the figures of information producers and workers a new faith for the extraction of surplus value.

Common examples of these new job positions are the ones of the journalist, the academic researcher, the designer, etc.

But since the production and usage of informations means indeed the production of language, also figures like the one of the call-centre worker are part of this new sphere. And more over, it is also important to remember all the area of care work, meaning affective and emotional labour performed mainly by women in different contexts: waitresses and shopkeepers, care workers, nurses, masseuses, prostitutes, and so on.

Affective and emotional labour are not exclusive of these jobs—they are present, in different quantities, in many other form of labour—but for sure they are the most visible and often forgotten type of employments, when talking about post-fordism.

A common element that many, if not all, of these jobs share is the value given to flexibility. When during the late 1960s and early 1970s flexibility was requested by students-finding-a-job and workers, as part of the transformations taking place in the job sector, it was seen by these subjects as a positive and important achievement. They could finally manage, on their on terms, their life and work, they could set their own working schedule, they could decide to have a short-term contract and spend three months at a job, three months in another job, then five other months, for example, travelling (or fulfilling personal aspirations). Ironically, this same lust for flexibility was also happily accepted by the employers. And the capital shifted, in the capturing of the workforce, from the traditional form of brutal, physical energy, sealed in each and every worker; to the working-desire, the desire for work they all had. The desire for self-realisation and, together with this, the energies produced through intellectual-mental labour and from the emotional and affective one. This passage marked a transition in the employers: from mainly demanding physical energy, to ask for mental and affective energy on top of the physical one.

> The capitalism the [anti-capitalist] protesters fought against in May '68 was the capitalism of the old spirit. (…) what was set up as an alternative to capitalist production, imagination, life, diversity, freedom, individuality, were the very features offered by the ideologists of new capitalism to form the new spirit of capitalism.
>
> Zdravko Kobe, Three Theses on Post-Fordism, from Gal Kirn, Post-Fordism and Its Discontents, 2010, pp. 151.

Workers were happy not to have fix jobs. This meant going against a long tradition imposed by their families, for which you had to look out for the job of life—which was essential to build indeed a family, a shared life, and reaching a form of stability. Instead, post-fordism introduced words such as part-time job, flexible hours, and in so doing also the idea of precarity: a life marked by a continuous precarious state of existence, due to a lack of job security.

For many job positions, such as, for e.g., the one of the journalist, the academic, the designer, etc. these new elements could have been taken in such a way that, after some years of paying one's dues, they may have reached a better position at the office they were working at, or maybe even setting up their own practice. But what would have changed after five years of working at a call-centre? Or as a care-worker for and with elderly and disabled people?

One of the biggest critics toward the concept of post-fordism is the fact that it is too localised to very specific geographical areas (for example, Italy and Japan), to stand as a general paradigm shift on its own. For this, many theorists did oppose to such a framework, saying that the fordist production model was only undergoing a structural phase of renovation, rather than living its final crisis. In the same way, other affirms [9] that the 2008 crisis marked one more milestone in the reversal of the current capitalist, fordist-driven, model of production.

A general common element traversing the whole "renovation phase" was the introduction, and more widespread usage, of computers and databases. From the beginning of this economic turn

—which for the most part developed in some areas of the western world, as noted above, but can now be seen outsourced to countries of the south-east Asia such as India (cfr. click-farms [10])—the popularisation of computers and databases were, for sure, essential elements in defining a new working environment, slowly then becoming part of society at large.

But, leaving unfulfilled the expectations of the Information Society theories' supporters, who dreamt computers would have given more leisure time to workers and consequently would have improved their engagement in politics, «information technology entered all production sectors—from service to manufacturing industries—and became their integral part without significantly altering the power relations within them». [11]

When in 1997, Maurizio Lazzarato wrote a text on *immaterial labor*, that umbrella term was quite useful to try to define and bring to light a new iteration of what post-fordism became after twenty-five years of development.

> The consumer is no longer limited to consuming commodities (destroying them in the act of consumption). On the contrary, his or her consumption should be productive in accordance to the necessary conditions and the new products. Consumption is then first of all a consumption of information. Consumption is no longer only the "manufacturing" of a product, but a real and proper social process that for the moment is defined with the term communication.
>
> Maurizio Lazzarato, Immaterial Labor, 1997.

Nowadays, that description has become obsolete, because it is so much embedded in the dominant societal order in which we live, that nobody is out of its reach. No more it belongs only to a specialised subset of professions (see, for example, the creative industries), for the most part contributing to these economic processes, but to anyone who is, in some form, part of society.

If in moving toward the post-fordist worker subject, Capital began to be interested in accumulating desires—in the form of information production of some kind—while getting a bit more closer and intimate with the subjective sphere of a worker (mental labour, emotional labour, and so on); now, with the internet user it is a matter of simply 'living' and doing the things you do in your everyday life: Capital needs only that to extract surplus value from your activities.

It is this the connection between the post-fordist worker and the internet user: data is now relevant for the Capital, not only when produced at work and for some very specific purposes, but as data itself, no matter the kind of environment in which it is born. Some value is produced out of it anyway. The precarious position often embedded within the post-fordist subject, even in the position of unemployed, now makes sense. Everybody at home can be capitalised by only browsing the internet, because home has now become your phone and computer (your personal devices), rather than a physical fixed site. And labour has been dematerialised both in terms of time, space, and also action.

Conatus as labour, labour as making interfaces

Data is essential to re-understand the user subject. This position implies an expansion of the entities capable of taking part in this role, opening it up to animal, mineral, plant, machinic, algorithmic and bot user, besides the human one. Data becomes the common ground that enables human user to interface with non-human and in-human users.

For this reason, there must be a shift at the ontological level in the language we use to talk about it: ‘organism’ is incapable of including all the different natures of which the user may be composed of. What does inhuman and ahuman imply, then? Not a desire to define these other natures against the human’s one, through a process of negation with which producing a definition of each entity. Rather, it mostly helps to remind us that the human user is only one possible variant within the user position, stressing a movement of expansion that finds superfluous the connotation organic/inorganic, and instead sites in the terms ‘operative, operational’ and ‘interoperability’ a new kind of language.

It is essential to understand how this shift, from an understanding of user as a living being to one in which its way of being in the world is through data production (*being operative*), traces a line to the concept of conatus.

If conatus is «the effort by which ‘each thing, as far as it can by its own power, strives to persevere in its being’» [12], then does this relate to the user subject in any way? What is the underlaying effort a user must do in order to keep being a user? For user is a position an entity inhabits, and each of these entities have already a conatus of their own even before entering the user position.

So what do we mean by «strives to persevere in its being» in connection to a user? Does a user strive to keep producing data and interfacing with other users, in order to avoid falling in dead, non-operative moments? Is there such a thought from the perspective of a user?

There is another, complementary, definition of conatus that goes more into the idea of iterative, ever-becoming processes. This definition might relate much more to the user position, as something than permits an entity to inhabit differently his her its surrounding and own being. Spinoza says:

> In this life therefore we endeavour [conamur] above all that the body of infancy be changed [mutetur] into another body which is capable of a great many things, and which is related to a mind which is very much conscious of itself, of God, and of things. (Ethics V Proposition 39)
>
> Stephen Connelly, Conatus: political being and Spinoza, 2015.

With this extended spinozian definition of conatus, the striving to persevere in one's being, of surviving, means also the continuous effort to adapt and iterate over time. In a sense, then, the question might not only be how much of an effort users have to do in order to keep striving to persevere in their beings, but also how much they are aware of the streams of data production and interaction they are part of, because this will influence their ways of being operative all the time.

In the user's position, labour is not always a conscious performative act, but an ever-happening activity, no matter the effort one's put in it: once performed, it can be channelled to a specific outcome when the outputted data is used for this or that other reason, task, aim, etc.

Being operative, in this frame, means then not being alive, or being on/off. Instead, it's a matter of how to build interfaces with other users and producing a different output with the convergence of two, or more, fluxes of data.

In the context of a post-work society where human users will live with a Universal Basic Income, how will the idea of conatus might change? With the rising and acknowledgment of many other users, will it turn into a drive for human users to strive to keep being relevant in their inter-operativeness amongst other users?

> Worse than being seen as an enemy is not being seen at all. As Eliezer Yudkowsky puts it, ‘The AI does not hate you, nor does it love you, but you are made out of atoms which it can use for something else.’
>
> Benjamin H. Bratton, The Black Stack, e-flux.com, 2014.

Looking at it in this way, for human users conatus will not be a matter of striving for one’s own sustaining, but striving for building different and better interfaces through which connecting with all the rest of users, in forms that destabilise humanist understanding of *being* in the world, of language and accountability, and so forth.

Conatus becomes a new form of labour, addressing a personal strive to stay together with others, and therefore, in how to use one’s data production to re-position oneself in the world. Interfaces, in parallel, assume the role of survival mechanisms to cope with other users, while being operative as one.

User agency

How does user agency manifest itself?

Unlike one might assume, an entity (algorithmic, mineral, human, etc.) does not correspond to a user in a 1:1 ratio: a subject takes place in the user position fulfilling a set of actions, but in doing this, it is being transformed because of that very position. This means that while a human actor, in becoming a user, changes his her being while being shaped by this new position as well, he she doesn’t align completely to that position: she’s not that user, but something else born from the aggregation of parts of her and some other agents constituting that particular user position. For this, that human agent retains all her qualities as human, and bring some of them, depending on the tasks at hand while inhabiting that user position, with her.

To give an example: when we send an email, the user is composed by a part of us (the one thinking and writing the text, choosing a recipient, etc.), the email client and the service provider, and all the other components that made this action possible.

For these reasons, user agency doesn’t correspond only to one entity’s desires, but it is shaped by many different agents, forming an array of desires and positions often visible in ways we at first don’t recognise.

> Economies of prostheticization are in fact important to understanding the posthuman User position, but their shape is not that of a one-way concentric radiation, from human into apparatus. It is, rather, a crisscrossing field in which humans themselves are just as likely to be the prostheses, and the apparatus is just as likely to be the User that prostheticizes the human, as the other way around.
>
> Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 273.

For example: in recent years the tracking activities employed by many web services have become part of public opinion's general topics of discussion. Keeping the same logic described above, it is easier to see how often the intentions we have and want to put into actions—say sending a message in which we talk about particular things—are being captured, read, processed and spitted back to us in the form of tailored ads, tracing a direct relation between the topics of our conversations and the products suggested to us.

Agency in this context then is, like the user position, computational. Data is the common language that permits everyone taking place in the user position, to look for and create interfaces, hence languages, through which to talk and read and listen to other agents, part for example of a platform.

It is clear how, more often than not, we are part of composite users [13]. But instead of thinking about them as 'other users' and so 'other human beings', we need to apply the word user to a more vast variety of entities that, for instance, we normally consider tools. Computation makes this easier to understand, because it transforms everything into data-readable and executable formats. Conversely computation, in the process of destroying humanist values and anthropocentric orders of the world, makes no exception or favour to anyone.

Interfaces are the places where disputing how user agency is being read and performed. Also for this, they become integral in the redefinition of what is life, what is labour, and the transformation of these two spheres into the one of operations, operational and being operative.

Building status and building history

Whereas oral language tends toward the creation of missing links (black holes) in the recalling of a fact we used to know, and for this it takes over someone's memory, computational language—even better than written human language—has a tendency to track everything. It keeps traces of what it has been processed and put them into indexes. Sure, it's possible to edit those traces, and in so doing producing alterations between data sets, but that's not the point.

At a structural level, computational language transforms the substantive 'archive' into a verb. As a consequence, that's what we began to do as well, as human users.

`git` works in the same manner, with the difference that the user has to comment after one, or a bunch of, edits performed to a set of files, briefly attesting what happened in making those changes.

This action is called `git-commit`, and it helps to log more clearly what was updated, deleted and revised, throughout the history of a project.

This mechanism of tracking and commenting, tracking and commenting, forces the user to start adopting a particular mindset—or at least confronting him herself with a thinking mode very well defined (and maybe at odds with his her own).

`git-commit` was conceived as a useful option when working on distributed software projects (e.g. Linux), because for each new improvement or feature added to the codebase, all the rest of the team could easily see what was changed and proposed, then deciding if approving it or not.

If on its original form it works as a way to make someone responsible for his her actions, especially when in relation to a group-based project, the same `git-commit` can be used to add messages with no real connection to the work executed, and building a different, parallel history of its development.

In the context of becoming more aware of how this builds one's accountability, injecting a disconnected and 'other' narrative to describe a very precise but different set of decisions produces confusion and anger in the next user reading it afterwards, because he she will find disconnected comments left by a previous user. At the same time, it's a useful thinking to foster unwanted or non-conceived actions (using the commit text to talk in a Q&A manner with the rest of the team; or, doubling one's activity between coding a website and writing a narrative about it while developing the project; etc) and freeing this `git` command from its normative function to different parameters and output. To expand `git` to more and more users whom purposes are different than the prescribed ones (e.g. writing prose instead of writing code), means stretching and re-contextualise a tool built and conceived to maximise particular tasks, and push it to its most useless and deferential applications. To see where and when it breaks. To create otherness and a more varied understanding. This very phenomenon is one of the elements that defines usership:

> Repurposing, by transforming former ends into new means, neutralises the sacred. In this respect, usership is synonymous with the act of profanation. The useful, indeed the used in general, is profane.
>
> (…)
>
> ‘Repurposing’—a term that captures both usership’s paradoxical idleness (no need to add anything new) and its transformative dynamic (putting the given to new purposes).
>
> Stephen Wright, Toward a Lexicon of Usership, 2013, pp. 50, 56.

What is important to remember, once again, is that this is as well a mechanism to build one’s accountability within a project. Though on github.com users don’t receive marks of any kind (points, or stars, or karma, and so forth), they are very much responsible for every contribution they perform. Since every commit is viewable by anyone accessing the url of that project (except for when a repository is private, but that’s almost never the case for projects users want to spread around), with this action it’s possible to see what happened in the development of a project: when and who did what, and how, and for which reasons they accomplished that.

The process of working on a project therefore, doubly binds with both the process of building a history of that project, and at the same time of building a level of accountability for each contributor who takes part in it.

As an uncommon choice for a social platform’s architecture, on github.com what appears first and foremost are lists of project repositories, then the users working on them. With this shift in the order and visibility of a user, also comes a reverse of hierarchy in the content order of the website: the work is of utmost importance, and users are tiny bits humming in the in-betweens.

Github.com’s personal user page nonetheless, has become a portfolio website for many developers, and also for designers dabbing primarily with code. Much better than any linkedin profile, a GitHub’s user page adamantly reports number of user’s projects created, number of projects that a user has contributed to, popularity of his her projects across the GitHub community (or should we call it group of users? Is there any real connection per se on GitHub?), and complete activities (hours, number of commits, project history, issues, pull requests,…) for every project.

Better than any punching-card and office supervisor together could ever do, a user is voluntarily tracked by `git` and actively filling in any useful (or just any) detail left blank for him her to complete.

Chapter #2, Platform

Why github.com?

Github.com is a bizarre platform, because it acts as a possible shift in the understanding of labour, while presenting itself, at the same time, as a perfectly integrated layer in the infrastructure of every big and small company.

Drawing from the free and open source software ethos, for which free circulation, non-copyrights, collaboration and remixing of code are the founding values of how to frame and to do work, github.com implemented one of the most used tools to put in practice this approach, `git`, at a world-wide level.

Whereas GitHub the company is born to be sustainable and making profits, selling for this reason private repositories (online space for projects that are visible only to the user who created them, and to others a user gave access to) and custom integrations with one’s own private servers, at the same time they are spreading a particular set of ideas all over the commercial internet, some of which are very useful in producing a cultural shift on many areas of what labour means nowadays.

Labour on the internet, indeed, often happens as a non-deliberate act. Think of the CAPTCHA system for example: a task where users have to prove they are human beings, by typing a short combination of letters and numbers from an image they are provided with. reCAPTCHA uses the typing users perform to prove they are human, to scan and digitise paper books more accurately than a robot could do. Many entities (Google Books, The New York Times’ archives) have used this tool to successfully convert physical archives of books and what not, into digital files. If we perceived the CAPTCHA as a riddle, then reCAPTCHA is a form of labour disguised as play.

GitHub, instead, creates a space where work is shared, complementary, remixed, discussed and improved.

In relation to a possible future scenario defined by a post-work society, platforms like this could be used as a rough blueprint to organise labour in a way, perhaps, more suitable for certain kind of production executed for reasons other than earning a wage. Value will be produced and identified anyway, but will differ from money.

Though they are the first ones not following what they offer (as described below in regard to GitHub's copyrights over its own software), at a practical level a GitHub user has to adapt his her workflow in relation to an environment born from a programming background. Most people joining github.com are part or hang around this circle of production, still, such aspect is essential in drawing a line between labour, internet, and its users.

As computation, understood as the process of resolving problems through the execution of some kind of calculation that follows specific parameters, replaces (or doubles) and transforms more and more parts of human life, data becomes the common material produced out of these processes. Far from suggesting that data takes the place of everything that build the world we live in, disembodying bodies of different kinds and abolishing materiality, I am more concerned and curious to explore how it affects these bodies and how we can read it and use it to readjust our understanding of what constitutes, in this case, work.

If used *en-masse*, GitHub would be able to familiarise its users to many of the mechanisms taking place on different levels of what makes a human life-turned-into-human-user, in the twenty-first century.

For example, data tracking, processing, reading, extracting and its usages is part of what defines the transformation from individuals to dividuals:

> In the societies of control, on the other hand, what is important is no longer either a signature or a number, but a code: the code is a *password*, while on the other hand the disciplinary societies are regulated by *watchwords* (…) The numerical language of control is made of codes that mark access to information, or reject it. (…) Individuals have become 'dividuals', and masses, samples, data, markets, or 'banks'.
>
> Gilles Deleuze, Postscript on the Societies of Control, 1992, pp. 5.

Becoming a mix of codes instead of retaining one's personality might have seemed the only outcome characterising the new society of information, about twenty five years ago. But why not trying to repurposing `git` as a tool, to better understand what other meanings, functions and outcomes tracking might have?

Since tracking processes, in the form of check-ins with a smartphone app in all the places a user visits or wants to mark for his herself, in making lists of things to buy or read for later, in tagging someone or something—in a sense, in the act of indexing one's reality—has become a common thing amongst general human user activities. And since that tracking is the fundamental activity of using `git` and github.com—as a form of measuring and reporting labour activities which took place before. What are the processes put in place when this activity is performed? In a sense, how does computation, a form of information traceability, look like when it becomes part of one's existence?

Terms and conditions: machine users

To create an account on github.com, the requisites are: 1) to be at least of a teenager age (13 yo) and human, 2) providing a name and email address, 3) using one account per user, unless it's for a machine user [14]—in that case it can be hooked up by as many human users as needed, and 4) taking care of one's password and uploaded content. As a bonus, a human user has the option to create one machine user account.

On GitHub machine users are acknowledged and given some options. To avoid spam or hacking, and in a way also to implement a norm of human self-responsibility, these accounts cannot be automated in their creation. They have to be set up by other human users. A bot can't create a bot account on github.com.

The main reason for wanting to use a special machine user account, is to automate tasks when working with more repositories at the same time: speeding up processes, updating a range of submodules [15], etc.

In the overall exploration of what it means to move towards a posthuman society, where the human is not replaced by better, more advanced, or different entities, but rather realises his blindness in putting at the centre of the world the anthropos as the supreme being, GitHub's acknowledgment of machine users looks emblematic: it reinforces GitHub's tagline that its platform is 'How people build software', assigning proper user agency only to human users. Rather than a question of enslavement and or submission, which sounds almost arrogant in its embedding of souls into machine users (cfr. animism), the point is how to find a common ground between human users and machine users.

For this, I don’t yet see problematic GitHub’s need for human authentication when creating a new machine user, because the platform is born to organise and shared human labour first of all. But slowly, it could be used to actively promote a gradual transition toward a common language of data and its computation between human and machine users.

When human users assign machine users with tasks they have to complete, the two positions turn into a composite user 16. This very fact asks for a shift in the way human users see their position as the privileged one, and it proposes a reframing of their relations with other users through less human-centred design solutions. Again, a human being entering the position of user makes him her leave behind many particularities constituting one’s personal self, and let him her retain others, to inhabit a subjectivity of a different kind. When I assume the role of a user, I am not my full self, but portions of it depending on the task at hand. The common ground I think might be necessary to look for, between human users and machine users on github.com, is due to the fact that they are both constituted by some shared elements: a set of available actions and data. The nature at the basis of each user is determining in the way they interact with each other, but should not become a ‘box’ that helps to avoid better and different interoperability between them:

> (…) putting nonhumans into that User position should be seen as a temporary station at best, but perhaps a means to invent different kinds of agencies, not just mimicking this degraded human. We must save the nonhumans from being merely humans, so that they could show us a different way for us to be both human and not.
>
> Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 274.

The rising of bots to help human beings with mundane, boring, repetitive, or very complex computational tasks is telling in the measure each of its human creators use them for.

On a more basic level, many of this bots are often dummies and troll experiments, set up for example, to make a user’s followers growing, or spamming all repositories containing a particular software component in their architecture.

In this case, human user agency still prevails all over. But, as it happens on Twitter since a while, where bots of different scopes are populating the platform and enriching the streams of many human users who follow them—from literary experiment, to snarky cut-up tweets, to any other variation in topic and form 17—which shape could this take on GitHub?

A shift in the current situation, would imply a recalibration of usership. To be very brutal, we could see all technological infrastructures as using human users not as agents, but as components of larger mechanisms. Usership is defined both by using as well as by being used. Being operative is exactly that: transmitting something, receiving something else. In these two actions, informing and deforming the material being passed on.

Github.com is born as a way to spread, facilitate and automate this operativity. Setting up rules where human users are the gatekeepers of the platform, makes sense in a culture still very much imbued with humanist values: it is, nonetheless, a process that could bring human users closer with users of other natures. To give 'only' human users permission to add machine users, encounter them and being in relation with them is maybe the way to go in this initial phase: not to make this a reason to let human users tame machine users, but, effectively, the other way around.

Terms and conditions: Issues and Pull Requests

Part of GitHub's architecture consists of specific features added as extensions of `git`'s existing structure.

The main sections around which github.com is built are:

- Code
- Issues
- Pull Requests

Code is the default section and displays: a) the commit history of the project folder—a list of the files sitting in the root level of a folder, with a brief text next to each file/folder describing which change has been made; b) the readme file—the text of presentation for the project, usually giving a general overview of the work and some technical informations of how it has been built and how to use it 18.

While the commit history is at the basis of `git`'s architecture, and it's possible to visualise it when working with `git` on the Terminal as well; the readme file becomes so much central only on github.com, acting as the welcoming text for a new user visiting the project page.

On GitHub users are free to `fork` (cloning to their account) every project repository they have access to (since repositories can be public or private), and with this action using and or contributing to each project they have an interest in. This action is part of a general usage of the platform, that consists in working on a project in one's local hard-disk and synchronising it on github.com when new changes are made. Forking is the entry point to collaborate on someone else' project, commenting and improving it, while still being able to use it for one's own needs.

This fluid exchange and improvements of code is part of the two most used and quintessential features of GitHub, *Issues* and *Pull Requests*.

Issues gives users the possibility to report problems of any kind that they have encountered in the usage of a particular project repository. It serves as the help & assistance centre of a project, with the benefit than anyone who wants to contribute to it can take part. In fact, the creator of the repository is not the only one in charge and or in power of that project.

The other useful area, parallel to the Issues section, is *Pull Requests*.

Whereas in Issues a user would, and should only post, problems, in Pull Request users can suggest new features and improvements for it. It's not that uncommon for a reported bug or missing feature, to transition from Issues to Pull Requests and being transformed into a feature request. At its best, this structure lets swing from the two sites of discussion very smoothly, improving either what exists, or pushing for new expansions.

There are three more tabs that builds each project repository on github.com, *Pulse*, *Graphs*, and *Wiki*. Pulse shows a dashboard with all the working activities which took place so far. Graphs is a tab with the list of contributors to the project and other statistic diagrams displaying how operative the project has been until that moment. Wiki, is a place where extending the readme file (the introduction text of a project) with more detailed informations. In describing the peculiarities of github.com though, these three tabs, while useful, took less weight in the success of the platform.

Being Issues and Pull Requests the two areas of conversation and exchange, one think of how to avoid bureaucratisation on the platform. There are tons of software that sits on top of github.com to better organise, filter, assign, label, and resolve issues on GitHub.

Similarly, the amounts of ad-hoc solutions, fragmentation of the apparatus into smaller tasks, and cyclical proposals coming from GitHub users, suggests how the Issues tool has become essential within the platform and, exactly because of this, in need of many face-liftings—if not of a structural rethinking.

The implementation of a quick-access option to the emojis keyboard, when typing **:** + any letter or number, lets users dealing with Issues and or Pull Requests with ease, showing their approval or disregard of a particular opinion, or of a new implemented feature, with a couple of strokes.

With an open ended amount of users possibly interacting on many popular open source projects, the level of time wasted by the maintainers of these repositories just to check a new notification —reporting yet another user commenting with '+1/👍/me too'—grows huger and huger.

Since this way to show support is quite widespread, a group of GitHub users proposed to implement a proper voting system, able to automatically transform the above reactions into +1 or –1. These users, tired of how the platform functions, and having to manage big open source projects hosted on it, published an open letter directed to GitHub Inc, in the form of a new repository project. [19]

In it, they suggest how they would like the Issues tool to work, reaching a peak when they surf over the fact that, if github.com would be open source, they would be more than happy to help with the implementation of what they have been asking for. So many nervous grins flirting with a company which, in less then ten years, now hosts the majority of the open source projects on the commercial internet.

This perfectly resonate with the history of the open source community, at the schism-level with the free software group. Born as one and the same, and divided, in 1998, from the original group after almost fifteen years of being together, the new formed open source group stepped away from the free software movement for some disagreements at the ethical level. While often the two groups are taken for the same thing, they differ in their philosophy:

> (…) whereas free software is always also open source, open source software does not necessarily have to be free software. That is, software can be open source without granting its users the additional freedoms that free software guarantees.
>
> LINFO, Open Source Definition, 2007.

What are these additional freedoms? Open source software cares mostly that a program be at no monetary cost and that its source code can be inspected. At the level of what one can do with the source code, though, free software lets a user not only read it (as open source software does), but also modify it and redistribute it without restrictions—under the rule that it will remain free and modifiable also for other users.

For this, free software sees itself as a social movement advocating freedom in its multifold manifestations; open source software, instead, aligns itself to only a 'practical' development methodology, ignoring ethical concerns. [20]

Comparing this quick sketch of the differences between free software and open source software, it is clear how this positively re-affirms the open source movement's aspirations at the time of their split with the free software group: to appeal to the corporate, business-y environment. Making them use free software, indeed, was perceived as pitching a poorly made product to the 'high society', whereas the industry-informed ethos of the open source group was fitting much better, since their attitude privileged focusing on making well made products over dealing with ethical questions of any sort.

On their side, GitHub Inc. replied to the open letter published by the big group of developers mentioned above, saying that they not only are listening to their requests, but are also working on some extra features to rethink the Issues section on the platform.

That's how your roll it out.

git's decentralised structure and GitHub's centralised one

Github.com acts as an interface to `git`, making the activity of working together on the same project, and sharing new code, easier.

Even though it is possible to set up one's own server with `git` installed on it, and create a web interface to facilitate the exchange and collaboration of projects, what github.com offers is—as many other internet services and social platforms—the possibility to focus on what you want to do, and not investing time in the underlaying infrastructure governing those very activities.

Perversely, github.com has become the most recognised and main used website for sharing `git`-based workflows, as well as the biggest website hosting open source projects. The perversion lies in the fact that in becoming this, in the first case it is going against the very principles and architecture employed and suggested by `git`, while, in the second instance, it is finding itself in a bizarre situation.

The two main factors that put GitHub Inc. in the above position are:

1. centralisation of the decentralised-by-design git's infrastructure
2. copyrights over github.com's software

In its inception, and to put it cynically, `git` is decentralised because its creator, Linus Torvald, was in need to get patches from the various developers helping him with the development of Linux. Originally, `git` was built as a reaction to some problems the Linux project had while being hosted on BitKeeper—a popular Version Control System (like github.com is). After Andrew Tridgell, a popular computer programmer involved in the free software scene, tried to build a free version of BitKeeper (which is proprietary software) for learning purposes—as he put it, part of the Linux project was treated, by BitKeepers itself, to not being freely hosted on BitKeepers' servers anymore. For this, Torvald, in need of a replacement, decided to create a free and better implementation of BitKeeper to use for his own needs.

Version Control Systems are divided in those implementing a centralised architecture, and those adopting a decentralised and distributed one. In the former case, users have to work with the same copies of files on a central server. If the server has problems, users working with it will automatically suffer as well. `git`, instead, is a distributed system, where each user has its own copy of the same project, and for this, it is, potentially, also a decentralised one, as users are spread all over the network. It is only potentially decentralised, because it requires that each user would set up his her own `git` server, and would give access to it to other users as well.

Still, this multiplicities of Linux enthusiasts were perfect for Torvald: more contributions, more choices to pick from or to discard.

The problem with github.com copyrighting the software that makes it run is that, unlike any other repository hosted on its platform, GitHub's own project repository can't be duplicated and used, and improved, and modified—it's not even listed along the other projects on the platform.

Besides this being at odd with the fundamentals of `git`, it makes it very hard to create a relay effect for which, when github.com is having some issues, another 'copy' of it takes over and keeps the service up and running. If GitHub is down, a big chunk of the internet is down as well. Of course, for this to happen, first GitHub should be open sourced, whereas now it's not; and second, many GitHub users should invest a small amount of money in duplicating on their personal servers a perfect copy of the platform's software and make it accessible to other GitHub users. Ideally then, if we would stretch this current limitation (github.com's biggest one?), there would be an easy option, in the configuration panel of each user, through which setting up a copy of GitHub's software on one's own server (or with a hosting provider, as one would do when making a new website) by only following some easy steps.

GitTorrent, a project by Chris Ball, aims exactly at resolving this problem. Using BitTorrent as its peer 2 peer decentralised network infrastructure, GitTorrent, in its first experiment, would be able to create this relay effect I sketched above. What GitTorrent proposes, is to make `git` decentralised for real—since the necessary architecture is there already, but the infrastructure to do it still requires quite some effort to be well implemented and kept up to date.

In the case GitHub's servers were down, GitTorrent would permit the user of the former platform to still access it, because every user adopting GitTorrent, would automatically put online, in a peer 2 peer network (through BitTorrent), their local repositories. It's the same when you download a movie through torrent and then let the torrent application still on, so other users can munch bits of the movie's file as well, and complete the download process. This already happens when using GitHub: a user have (or should have) a local copy of the project on his her hard-disk, and should synchronise it with GitHub when making new progress. GitTorrent moves away from this infrastructure [21], by using a decentralised network architecture (BitTorrent) which relies on users themselves and the internet network, rather than on an unique site of storage.

As Bratton says [22], platforms, acting as a third site between the State and the Market, both centralise some functions and decentralise others. What GitHub did, was to centralise `git`'s distributed architecture into a common space for anyone to join. While this have brought many benefits, if only the general spreading and higher relevance of open source projects, on the other side it's a movement undermining the very reasons which made Linus Torvald leaving BitTorrent, for something else non-dependant on a central infrastructure.

GitHub's merit, while temporary, might be to have created the conditions for a massive transition toward a truly distributed and decentralised `git` network. But as with many other projects of the same kind, the desirability to start adopting, promoting, and co-developing GitTorrent, lies in how easy to setup and maintain it is. If the adoption is smooth, then there might be a higher level of interest in learning how to properly use and misuse a tool. 'User-friendly' should stop acting as a way to dumb users down in name of simplicity and, rather, be a synonym with agile entry-point: first quick adoption, then, based on the granularity of a user's knowledge, letting him her navigate confidently the tool no matter what. Giving too much options at the beginning might be fatal, still, they should be available when a user will reach a better level of mastery.

Since the thinking and making of new interfaces between users is the new conatus as work, when data and computation become the common material through which interacting between users of different nature, github.com is making a good job in expanding its platform to new users. In parallel, a new iteration of it in the form of GitTorrent, could bring and spread more literacy on a technical level, while implementing different ethical values at the same time. How would this set of values position themselves in the sketching of a posthuman ethics, though?

Chapter #3, Conclusion

In order to proceed with the above question, there is one more element to bring into account: the new geographies we live in as users. Benjamin Bratton proposes The Stack, both as a schema and a structure, to understand the overall population of different computational systems already in use and constituting the world we inhabit. The Stack refers to:

> a transformation in the technical infrastructure of global systems, whereby planetary-scale computation has so thoroughly and fundamentally transformed the logics of political geography in its own image that it has produced new geographies and new territories that can enforce themselves. Unlike modern political geography, which divided up horizontal maps, Stack geography also vertically layers spaces on top of one another. Instead of surveying all the various forms of planetary-scaled computation—cloud computing, smart cities, ubiquitous computing, massive addressing systems, next-generation interfaces, nonhuman users, and so on—as different genres or species of computing, each off on its own, this model locates them on layers of a consolidated metaplatform, an accidental megastructure. (…) The Stack is a model for thinking about the technical arrangement of planetary computation as a coherent totality, as well as a conceptual model for thinking the contradictory and complex spaces that have been produced in its image.
>
> Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 375.

A new set of ethics dealing first with the user position, then trying to move beyond it and embrace the posthuman environment of the user, will have to work in close relation with the geographies of something like The Stack. Computation has significantly become an ubiquitous process part of our reality and of the nature of the time we live in. To better operate in this new space and being able to explore the user position we can inhabit as humans, a *new literacy of computation* will be necessary.

In tracing lines between the post-fordist worker and the internet user, and the related working environments the two subjects operate within (respectively office-home-etc., using internet to communicate with a team or clients; and internet platform) I saw, very often, a lack in the understanding, knowledge and experience of this new position. With the intentions to foster processes that amplify the formation of such new sensibilities, a new literacy of computation would refer to, for example, the making of a new language through which describing what happens when we interact with other users (for instance, framing users as 'operative' and not as 'living organisms', as pointed out in Chapter #1); and, getting to know and mastering (while being mastered by as well) the tools we have at our disposal, both accidentally built or intentionally designed. This last one can be seen amongst the founding values that describe usership: a set of practices where using, reusing, misusing, repurposing, and so forth, have become a common framework of action specifically because of the rising of digital technologies. Usership, in this respect, might be seen as a continuation, into less specific environments, of the hacker ethos of learning something by dissecting it.

It is because of these reasons that `git` might also be an experiment in the act of defining what might a new and different literacy of computation look like. As a tool to track informations and build histories out of it, it can as well be used in a reversal manner and let human users better understand what are the other applications and ethical repercussions of this activity. As a response to the general public discussion about the collection of citizens' metadata from companies and governmental institutions alike, `git` might be used to reflect about the different architectures of data collection and distribution, re-staging on a personal scale systems that might be hard to grasp at first, when contextualise on larger environments.

At the same time, through a new literacy of computation, I look for ways to embrace complexities that are not user-friendly in the current meaning of the term, often handicapping my human-user framing, understanding and mis-understanding of them, and longing for tasting data, for being able to relate to it in modalities that go beyond looking at my stats on a website platform's dashboard, but for example realising that my own body is a mix of metadata of different kinds, starting from my own faeces:

> Grandparent toilet bowl stool diviners, you were right: considering the density of DNA from microbes, from one's own internal fluids, and from the DNA of the remaining food stuffs that one ate, the human stool is one of the most information-dense substances you

> are likely to regularly encounter. Bit by bit, your stool is far more information rich per cubic centimeter than the flash memory in your iPhone.
>
> Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 269.

Relating the understanding of my reality, body included, with data doesn't necessarily turn me into a computer. Inversely, it can help me to better relate with data and computation without making me forget the starting position from which I am experiencing it: a human body.

I think, overall, that a new literacy of computation, understood in this way, might also help to finally realise that we don't have to take either the positions of machines (as in, human beings are machines processing data and provided with functioning organs), or the one of nostalgic human beings trying to restore a supposedly once existed balance that governed the world, between the natural world and us.

We live surrounded by machines of different kinds, with data and its computation that has become essential in the functioning of our reality, but not for this we have to overcome our nature of human beings and strive for delirious transhumanist positions—where the mastery of the world is in the hands of human beings with implanted prosthetics components to augment their sense of power and control around them. Posthuman, for how I intend it, doesn't mean beyond human, or over the human. Posthuman means human beings at the side of the world, as one side of the world, because there are many other entities inhabiting it. Rather than disparaging what makes us human beings, it's maybe a matter of iterating ourselves over what keeps us from finding ways to relate with other users. Machinic, plant, mineral, animal, algorithmic. They might ignore us, but we might realise we are in need of them.

Bibliography

In order of appearance:

- Srnicek, Nick, and Alex Williams. Inventing the Future: Postcapitalism and a World without Work. Verso, 2015. Print.
- Horning, Rob. "Preemptive Personalization." The New Inquiry. 11 Sept. 2014. Web. 31 May 2016.
- Huff, Jason. "American Psycho." Jason Huff, —American Psycho. Web. 31 May 2016.

- GitHub. GitHub—Build Software Better, Together. Web. 31 May 2016.
- Craig. “Shit Job.” Urban Dictionary. 07 Sept. 2004. Web. 31 May 2016.
- “Automate Now? Robots, Jobs and Universal Basic Income A Public Debate.” YouTube. UWE Philosophy & Post-Europe Project, 21 Dec. 2015. Web. 31 May 2016.
- Lordon, Frédéric, and Gabriel Ash. Willing Slaves of Capital: Spinoza and Marx on Desire. Verso, 2014. Print.
- Bratton, Benjamin H. The Stack: On Software and Sovereignty. MIT, 2016. Print.
- Kirn, Gal. “Post-Fordism and Its Discontents | P-DPA.” PostDigital Publishing Archive. 03 Nov. 2010. Web. 31 May 2016.
- Arthur, Charles. “How Low-paid Workers at ‘click Farms’ Create Appearance of Online Popularity.” The Guardian. Guardian News and Media, 02 Aug. 2013. Web. 31 May 2016.
- Lazzarato, Maurizio. “Immaterial Labor - Maurizio Lazzarato.” Immaterial Labor - Maurizio Lazzarato. Web. 31 May 2016.
- Connelly, Stephen. “Conatus: Political Being and Spinoza.” Critical Legal Thinking. 16 Mar. 2015. Web. 31 May 2016.
- Bratton, Benjamin H. “The Black Stack.” E-flux. Mar. 2014. Web. 31 May 2016.
- Wright, Stephen. “Toward a Lexicon of Usership.” Museum of Arte Til Tools Comments. Web. 31 May 2016.
- Deleuze, Gilles. “Postscript on the Societies of Control - Gilles Deleuze.” Postscript on the Societies of Control - Gilles Deleuze. Web. 31 May 2016.
- GitHub. “Managing Deploy Keys.” Managing Deploy Keys. Web. 31 May 2016.
- Lunden, Ingrid. “Pentametron Is A Twitter Poet That Gives Bots Some Literary Cred.” TechCrunch. 13 Jan. 2013. Web. 31 May 2016.
- Lanfear, Robert. “Roblanf/phypapers.” GitHub—Build a twitter literature bot in three steps. 25 Feb. 2014. Web. 31 May 2016.
- Fincato, André. “afincato/ow_draft–1." GitHub. 04 June 2015. Web. 31 May 2016.
- Various authors. ”Dear-github/dear-github." GitHub. 15 Jan. 2016. Web. 31 May 2016.
- The Linux Information Project. “Open Source Definition.” Open Source versus Free Software. 27 May 2006. Web. 31 May 2016.
- Pasquinelli, Matteo. “Autonomous Commons.” P2P Foundation. 28 Jan. 2008. Web. 31 May 2016.
- Ball, Chris. “Announcing GitTorrent: A Decentralized GitHub.” Announcing GitTorrent: A Decentralized GitHub. 29 May 2015. Web. 31 May 2016.
- Brain Bar Budapest. “The Politics Of Platform Technologies Neither State Nor Market (A Bit Of Both) - Benjamin Bratton.” YouTube. YouTube, 01 July 2015. Web. 31 May 2016.

Footnotes:

[1] Since a Universal Basic Income «must be a *supplement* to the welfare state rather than a replacement of it»—Nick Srnicek and Alex Williams, Inventing the Future, 2015, pp. 119. ↩

[2] A project dealing with algorithmic ads suggestions based on the content users produce is American Psycho, by Mimi Cabell and Jason Huff: «This book was made by sending the entire text of Bret Easton Ellis' American Psycho between two gmail accounts page by page. We saved the relational ads for each page and added them back into the text as footnotes». ↩

[3] The other three being: fully automating the economy (full-automatisation of labour), reducing the working week, achieving a cultural shift in the understanding of work (Nick Srnicek and Alex Williams, Inventing the Future, 2015, pp. 108). ↩

[4] From urbandictionary.com: «An unfulfilling, tedious waste of 8+ hours of the day, five days of the week, that you only persist with in order to pay the bills and the ever mounting debt that comes from having to increase your spending in order to entertain yourself outside of working hours as you are so brain dead from the hours you spend in work.» ↩

[5] See this panel titled 'Automate Now? Robots, Jobs and Universal Basic Income A Public Debate', 2015: at time 1:25:30 they clarify what they mean by full automation — https://www.youtube.com/watch?v=ShmbzDceuYo&feature=youtu.be&t=1h25m30s. ↩

[6] Still, the advancement of machine automatisation also brought what is referred to as 'shadow work': self-check out devices at the supermarket, self-service petrol stations, self check-ins at the airport—all tasks that on one side give the impression of a more agile service, but, on the other, have the effect of cutting on personnel and assign a considerate amount of labour to the customer. ↩

[7] Igor Pribac, Post-Fordism–a Contextualisation, from Gal Kirn, Post-Fordism and Its Discontents, 2010, pp. 28. ↩

[8] Ibid, pp. 38. ↩

[9] Igor Pribac, Post-Fordism–a Contextualisation, from Gal Kirn, Post-Fordism and Its Discontents, 2010, pp. 31. ↩

[10] A click-farm is a place, usually a physical location, where barely-paid workers click on facebook's like, twitter's retweet, etc. all day long to create value for specific user profiles, companies, and so forth. It could be an online shop asking to gain visibility, or a private account trying to become popular, etc. These workers, instead of adding one small

piece to the production line of a product, do exemplify at best how what most users do on a less extreme scale: clicking on the share button of the most used platforms of the commercial internet to generate 'value': that is, revenues for the companies hosting the service (the the social platform) and the advertisers selling on them. See Charles Arthur, How low-paid workers at 'click farms' create appearance of online popularity, The Guardian, 2013. ↩

[11] Igor Pribac, Post-Fordism–a Contextualisation, from Gal Kirn, Post-Fordism and Its Discontents, 2010, pp. 35. ↩

[12] Frédéric Lordon quoting Benedictus de Spinoza in Ethics, part III, Willing Slaves of Capital, 2014, pp. 14. ↩

[13] «[But] the incorporation of many different types of actors (human, machinic, bots, animals, infrastructures) into recombinant User assemblages, all with differing regular morphologies and temporalities, means that inside the User position itself, rather different kinds of platform logics between individuated actors can occur». Benjamin H. Bratton, The Stack—On Software and Sovereignty, 2016, pp. 284. ↩

[14] From github.com's FAQ, a machine user is described as an account created by a human user, and set to execute automatised tasks on a GitHub project repository. ↩

[15] A submodule is a repository (a project folder) linked to another repository, as if it was a sub-folder of the latter. ↩

[16] See footnote #13. ↩

[17] See, for example, this article on a Twitter literary poet bot, and how to build a twitter literature bot in three steps. ↩

[18] For an example and dissection of the readme file on GitHub, see ow_draft–1, a text part wiki-how, part manifesto, that I made to explain how GitHub works and to outline some of the general thematics of usership. End of the year exhibition, Designing Democracy—Sandberg Instituut, 2015. ↩

[19] Here the submission—Dear Github. ↩

[20] The free software movements falls short, though, when they have to explain their ethical reasons and burning passion for freedom as an absolute value, in particular in regard to their implicit ideology on the subject of copyright, copyleft and creative commons. As Martin Hardie claims, in fact, «the logic of FLOSS seems only to promise a new space for entrepreneurial freedom where we are never exploited or subject to others' command. The sole focus upon 'copyright freedom' sweeps away consideration of the processes of valorisation active within the global factory without walls». Matteo Pasquinelli, The Ideology of Free Culture and the Grammar of Sabotage, 2008, pp. 6. ↩

21. The project is already quite ambitious and covers—as well as made use of—other areas and technologies: for example it employs blockchain to distribute usernames to adopt on the network. Here the article. ↩

22. Benjamin Bratton, The Politics Of Platform Technologies, youtube.com, 2016. ↩