STUDIA UBB SOCIOLOGIA, 64 (LXIV), 1, 2019, pp. 109-126
DOI: 10.2478/subbs-2019-0005

FORMS OF REMUNERATION FOR FREE SOFTWARE PRODUCTION: A REDUCIBLE COMPLEXITY[1]

EMRAH IRZIK[2]

ABSTRACT. How producers of free digital goods can be compensated for their labour is a major topic of debate and controversy in Free Software and related fields. This paper analytically disentangles the multiple modes of remuneration in operation in Free Software and presents the implications from a political economy perspective. The outlook of autonomous commons-based production in information goods is situated in relation to capitalism. In the process, certain conceptual contributions are made regarding the nature of information goods and the commodity form.

Keywords: Information society, free software, digital production, commons, capitalism

It is often asserted that since the seventies capitalism has entered into a new stage, variously described as "postindustrial", "informational", or "knowledge-based". This economy is characterized by an increasing emphasis, in terms of value-added, on the input of high-quality knowledge produced by high-skill labour in the production process. Qualitatively, it is also an era where the creation of a significant amount of wealth comes about through what Manuel Castells has dubbed "knowledge acting upon knowledge" (Castells, 1996), in the sense that intellectual effort applied to existing information and previous knowledge results in a new, highly sought-after, higher composition of knowledge. This knowledge, to various extents, can either be privately monetized in commodity form as intellectual property and used as a means of rent-seeking, or become part of the new, digitally representable commons which is shared, immaterial and inexhaustible, distinct from the classical, exhaustible commons like land and water. Under contemporary conditions, the profit principle dominates for the most part, and the distinguishing characteristic of knowledge-based capitalism has been that knowledge and information are

[1] This article is based on Chapter 3.1 *Concentric Circles of Remuneration* of my unpublished doctoral dissertation titled *Free Software as a Commons: Between Informational Capitalism and a New Mode of Production*.

[2] Guest Lecturer in Sociology, Babeș-Bolyai University Cluj-Napoca, email: *emrahirzik@protonmail.com.*

transformed into a restricted, monopolized, commodified factor of production. The fruits of knowledge-based labour as embodied in works of science, software, literature and art are monopolized via an expansive regime of intellectual property (IP). IP maximalism is legitimated on the individual level by reference to romantic notions of authorship (which is then assumed to be alienable and therefore transferable from the author to an intermediary such as the publisher), and on the collective level by assuming that strong IP protection promotes development by offering the only viable course for compensation.

It must be noted that once produced, knowledge and knowledge-embedded goods in turn act as the materials of labour necessary for the subsequent round of knowledge production, as each is a partial and constantly evolving embodiment of the accumulated cultural and intellectual riches of society. This fact, which has been latent in all eras of production, is only fully expressed in the contemporary knowledge economy. It significantly erodes the distinction between what is a consumer good (Department II product or means of subsistence in Marxian economics) and what is a capital good (Department I product or means of production). The consumer product is now increasingly also a means of production; the consumer is supplanted by the user (user-producer, prosumer, etc).

The strategy of profit-based enterprises so far has been to enclose knowledge on the basis of IP laws, trade secrets and employee regulations so that artefacts of the knowledge economy can be treated as if they were a rival good in the market, akin to material products of industry. A rival good means that one person's use of it necessarily bars another person from using the same, which is not the case for the artefacts of the knowledge economy. On the more ontological level, since knowledge is by its nature inalienable, in the sense that one does not part with it upon transferring it to another, it must be made a commodity by force of law, by way of restrictions placed on its reproduction (copyrights) and implementation (patents), so that knowledge can be treated as if it were alienable, in conformity with the logic of capitalist property. The commodity-form taken by these digitally representable knowledge goods, or digital artefacts, appears as *beyond fetishistic*. It does not merely substitute the appearance of relations between objects for what are in fact relations between persons. It denies, in the process, the material reality of the object (digital duplicability) in favour of legal fiction (copyright restriction). The digital artefact in commodity form not only appears to have value by virtue of its intrinsic physical properties rather than by virtue of being a product of social labour; it is furthermore only able to maintain this appearance because it comes into the hand of its buyer attached to a prohibition. In this sense, digital artefacts as commodities should be properly seen as embodiments of a *tabooistic* economic relation.

There is however, a counter-movement that proposes and practices a different knowledge economy against this arrangement in contemporary capitalism. This is a current which is advancing the Knowledge Commons, advocating and practicing the release of human knowledge in all its creative forms as global public goods, aiming to make this wealth open, shareable and accessible to all. There are diverse actors and avenues relevant to this process. In the software realm, this new intellectual commons takes the form of Free Software, as christened by Richard M. Stallman's GNU Manifesto in 1985.

Free Software (FS)[3], which offers a concrete alternative to the IP regime, is a particular way of organizing the production and distribution of software. It is based on social collaboration and free copy sharing, carried out by the coordination of a number of individuals that form an interactive community of producers and users. In FS, the human-readable programming code, called "source code," is made open and freely available under a freedom-guaranteeing legal license, together with the compiled binary computer software packages, which cannot be modified as is, but can only be executed by a machine. This allows anyone to develop the software by improving existing components and deriving new software, in addition to freely using it. FS thus constitutes an open commons; it is non-proprietary, in other words, held in common by all.

The phenomenon of a digital commons emerging in a limited sphere of production under our general capitalist system raises the question of what the motives to create such a commons could be. As the creator of the Free Software movement, Stallman cites numerous such motives. Some are non-economic, such as fun, political idealism, admiration (prestige), feeling of community and hatred of Microsoft (or large software companies in general)[4]. Others are auxiliary to the dominant commodity economy, such as cultivating professional reputation (increasing chances of getting hired) and education (part and parcel of the training of future software workers). Finally, Stallman cites wanting a better program to use, gratitude and money as motives for writing Free Software. These warrant a closer look, because they encapsulate the modes of reciprocity that are involved in the creation of this universal digital commons: Self-use points to the concern of an independent producer, except this self-use immediately becomes shared use due to free digital reproduction. Gratitude points to the ethos of reciprocity that emerges in this contemporary gift-like economy. Finally we have “money”, which is the most controversial and the most interesting motive; what role money plays in a system where the product is not a commodity and no sale takes place is my subject of inquiry.

[3] "Free Software" (FS) is interchangeable with the term "Free/Libre Open Source Software" (F/LOSS) that is frequently used in the literature. I have settled on the original term Free Software, as defined by the four freedoms articulated by the Free Software Definition provided by the Free Software Foundation. https://www.gnu.org/philosophy/free-sw.html.

[4] Motives for Writing Free Software. https://www.gnu.org/philosophy/fs-motives.en.html.

How those practicing the commoning[5] of software, meaning the free software producers, can be remunerated for their labour is a huge issue of discussion surrounding Free Software. Coding, documenting, designing, publishing and publicizing may require little in the way of means of production, but they take serious amounts of skilled labour to actualize in a satisfactory manner, and the output satisfies only a tiny share of human needs. Therefore, the production of such a commons will have to either be based on a congruently limited share of labour time of each individual producer, or be remunerated extrinsically by society. In practice, both modalities are in operation simultaneously and inextricably. Historically, there have always been a plethora of ways of remunerating labour other than out of returns from the sale of a commodity: communal sharing, gift exchange, elite patronage, public donations, state sponsorship are some that come to mind. What may seem mundane from this broad historical perspective nevertheless merits close inspection due to the near-complete hegemony of the contemporary commodity economy which currently holds sway over not only the material lives of human beings but also over their imagination.

I see the remuneration mechanisms of FS as a series of concentric circles. The unique contribution of this approach is to definitively map out the contemporaneous economic models in operation for FS production: its internal functioning as the kernel of a new mode of production, its point of interface with capitalism, and its potential path to independence and generalization. The origins of each circle are also historically identifiable and have an order of appearance. Circle 0 originates in the small avant la lettre FS communities in the universities and research labs of the '70s, Circle 1 originates in the '80s with the Free Software Movement, Circle 2 originates in the appearance of FS companies in the '90s, Circle 3 originates with the rise of online donations and crowdfunding in the late 2000s, while Circle 4 remains a future prospect.

The innermost circle, Circle 0, is defined by individual FS producers producing FS for their own use, which subsequently is put online for the fortuitous use of others. Circle 1 operates on the level of the community of all FS producers who benefit from each other's work; it may involve explicit cooperation, and the sharing practice contains within it a form of fair reciprocity. Circle 2 functions at the point of interface of the FS mode of production and the larger capitalist economy with Copyleft[6] playing an important role. Circle 3

[5] Julie Ristau. https://www.onthecommons.org/work/what-commoning-anyway.

[6] Copyleft is a clause in FS licensing which requires derivative works based on FS to also be FS. This restriction on placing restrictions on code is the subversion of copyright law to serve ends opposite to its spirit. Copyleft is enforceable by copyright law because the original author (the holder of copyright) is setting these terms, allowing all software freedoms except the freedom to restrict the freedom of others.

consists of rising direct compensation mechanisms functioning between FS producers and the larger public that uses FS. With the analysis of these already functioning circles complete, I argue in favour of basic income as a new, potential Circle 4 mechanism which can further accelerate the generalization of the FS mode of production.

The expansion of the circles provides a mental image to aid our understanding of the expanding economy of FS. But it is not a solely analytical tool. It also follows the empirical, historical path of FS as its remuneration models diversified. Often, a similar trajectory may also be observed over the lifetime of an individual programmer as the programmer moves between academic settings, from FS volunteering to employment by FS producing corporations, to becoming an independent producer of various forms such as freelancer (e-lancer), donation recipient or public patronage beneficiary, or even a FS start-up entrepreneur.

New outer circles in the concentric circle model do not constrain or immediately negate[7] the operation of the inner circle mechanisms, but include and supplement them. At a given moment, a higher circle operation such as an employee producing FS code for a capitalist company may be simultaneously producing for their own need (Circle 0) and this may also prompt unrelated other individual coders to collaborate and/or spontaneously reciprocate (Circle 1). Circle 0 and Circle 1 operations may be at one point aided by a Circle 3 mechanism such as a crowdfunding campaign. In a hypothetical future where basic income is realized (Circle 4), a FS coder may continue to work as an employee as well (Circle 2). Therefore, we cannot observe these complex practices in delineated fashion, but we can analytically identify them. As the number of individuals participating in FS grows by the inclusion of new people in the outer circles, the inner circles also expand, strengthening the system as a whole. Consequently, it will not do to think of the movement as a mere tacking on of outer *rings.* Each quantitative (more individuals in a given circle) and qualitative (formation of a new circle) expansion furthers the displacement of the market in favour of the commons but with distinct logics that are often operating simultaneously.

The succession of the circles is towards increasing the *extrinsic autonomy* of the FS producing labourers within the historical *interim* between

[7] If the progression of the FS mode of production continues and generalizes, in the medium term Circle 2 may be replaced by Circles 0 and 1 on the one hand, and Circle 3 on the other. This is an inherent possibility because of the contradiction between FS and capitalism. The shifting of individual FS producers between these circles is the reflection of this contradiction between the two modes of production on to class positions. In a utopian future where labour in the production of digital artefacts is predominant over labour in analog production (through automation), Circles 0 and 1 may also displace Circles 3 and even 4. This would imply a classless, post-scarcity economy.

capitalism and the potential future generalization of the FS mode of production, while the system moves in the same direction as a whole: achievement of economic self-reliance. This movement follows an S-shaped (sigmoid) curve, where Circles 0 and 1 provide minimal extrinsic autonomy, whereas Circle 2 provides almost full extrinsic autonomy, and is then only slightly improved again by Circle 3 (and a hypothetical Circle 4). In other words, the movement is towards the realization of self-reproduction within the FS mode of production; it is the movement from proto to full mode of production and goes hand in hand with the formation of digital producers as an independent class. This is *not* accompanied in the interim by a linear increase of *intrinsic autonomy* from the perspective of the individual FS producer, in the sense of choosing what to work on under which form of governance and with what regime of regularity. I rather claim that intrinsic autonomy follows a well-shaped (inverted bell-shaped) curve where in between the two cases of maximal intrinsic autonomy which are the first and final circles, extrinsic autonomy comes mostly at the expense of the intrinsic. The two forms of autonomy converge at maximum as we move to Circle 3 and beyond.

The addition of each new circle of remuneration to the system increases both the mass of use-values produced (amount of useful software), and thus the non-capitalistically satisfied needs of society, and the number of individuals (communities of software producers and also users) with a stake in the life of the system[8]. As the number of individuals engaged in the sphere of FS in various capacities increases, the *cultural* influence of FS increases as well, giving it the character of a social movement. This is reflected in both the explosion of academic interest in FS, peer production, open-source and the digital commons, as well as the interest of left-wing political movements and mainstream media commentators, whether business-minded, critical or utopian. The cultural influence of FS is sowing the seeds of a mass *political* consciousness of FS, which has already been reflected in the programs of progressive parties, the Pirate Party phenomenon, and FS-related activism.

[8] It is important to note that this constantly increasing production of new software code must continuously compensate for "bit-rot"; the deprecation and degradation of old software code due to the constantly co-evolving software ecosystem. Thus, we are looking at a field of work that is not simply cumulative, but which is always chasing moving targets in order to stay relevant. "Finished" and "complete" software projects are rare things. Constant improvements, iterations and maintenance requirements characterize the field, demonstrated by versioning. There is however an element of decadence in the world of proprietary software which increases the chances of the FS competition catching up: the inclusion of "anti-features" such as DRM and tracking mechanisms which make the software less useful to the user while more profitable to capital, planned obsolescence, as well as prematurely pushing out buggy, half-finished releases due to cut-throat competition among proprietary vendors.

In the tradition of the Free Software Foundation's (FSF) Four Freedoms Definition of FS, which begins with Freedom 0, I denote the innermost circle in my model as Circle 0. This circle consists of FS produced by an individual coder for their own personal need. This is what Eric Raymond has referred to as "scratching your own itch". The crucial point is that upon completion, the code which is produced in this manner gets shared openly, to the benefit of all others who may have the same need. This is the simplest form of FS production but at the same time it is where the *immanent principle* operating at the core of the entire FS edifice is visible in its purest form. It is a radical phenomenon brought about by the nature of the digital artefact: an individual producer of a digital use-value, by mere willingness to share, which comes at no additional cost to themselves, by the trivial means of digital copying and distribution over the Internet, automatically provides the use-values for potentially everyone possessing the same need for such a use-value[9]. This positive externality is the main driver behind the entire system, and grasping it is essential to understand its resilience and sustained expansion. Also, despite the partial parallel, this already sets apart the FS producer from the traditional subsistence producer who produces on their own what they will *individually consume.* Contrary to the material nature of commodified analog goods, in the material nature of the digital artefact, there is no contradiction and dialectical conversion between use value and exchange value. A contradiction between the individual and the collective does not arise on this point[10].

Circle 1 in the concentric circles model of FS is the so-called gift economy relation among FS coders. I refer to this as the "so-called" gift relation because the nature of the digital artefact was unaccounted for or under-appreciated by those who asserted the identification. Gift exchange economies

[9] They have to be able to actually find out about it though. This discovery process may be commodified. Here are two examples: Certain sneaky, small parties sometimes venture to re-brand and sell FS items to customers who are unaware that they can acquire the genuine product for free online elsewhere. While unethical and frowned upon, this practice is technically legal and is in observance of FS licenses, because FS licenses allow charging for distribution. The scenario in mind in allowing this however was FS CD sales and not this type of scam. Another case is the App Store model that has come to dominate the smartphone and tablet computing platforms, where software installation is mediated by a gatekeeper (Apple's iTunes, Google's PlayStore, etc.) who may collect fees and/or commissions from app makers and/or users of the App Store, including for FS. These are cases of consumers' lack of information regarding alternatives in the market leading to the realization of rents (a market inefficiency). See the concept of the "Attention Economy" (https://en.wikipedia.org/wiki/Attention_economy) for further inquiry into such issues.

[10] The realization of the use-value of a digital artefact does not take the form of consumption, but instead the form of using a copy. Replication of the digital artefact among its users is just one more instance of the mundane operation of copying which computing as a whole is based upon. In its technical functioning, copying data over the network, i.e. between users, is not categorically different than copying data from the hard drive to the memory of a system that belongs to the same user.

by necessity operate on the basis of an item gifted to someone specific equalling said item being removed from possession in the process of exchange - whether immediately or after a delay - in favour of another. Digital copying, as is the case in FS, means any and all parties, who are mostly anonymous, maintain possession of the exchanged artefact simultaneously. Still, let us ignore this theoretical shortcoming for the moment because flawed as it is, the gift economy perspective is based on an important insight. Early theorizers of FS arrived at the gift economy model because they observed a community of vocational programmers freely sharing code predominantly among themselves (Barbrook, 1998, for example). This implied a practice of reciprocity, even if it was implicit and unstructured. The producer base of software was more or less identical to the user base. This of course was to change dramatically with the PC revolution, leading to a situation where the vast majority of users are not programmers, although their use itself partially contributes to the production of software through the indirect mechanisms of network effects and providing feedback.

Nevertheless, an essential positive externality was in effect in this "gift exchange" among programmers. As a thought experiment, let us imagine now the first act of code exchange between coder Alice and coder Bob, a hypothetical *ursprünglich* moment in the software commons. Let us assume that through their own labour, Alice and Bob have respectively produced codes Foo and Bar. Let us further assume that Alice and Bob are both producers of average skill and it took the same amount of labour time to respectively produce Foo and Bar, making them of equal value. Both interested in the use-value of each other's piece of code, Alice and Bob now engage in exchange, i.e. they provide copies for each other of Foo and Bar. Both are now in possession of both Foo and Bar. Concerning fairness in remuneration, both are now fully compensated for their efforts in producing their respective pieces of code. But something extraordinary happens here due to the nature of the digital artefact, which sets Alice and Bob apart from two simple commodity producers engaged in a direct exchange of equal values. The "exchange" is not a private affair as in the case of market exchange. The exchange occurs over a public network, and there have been no copy protections placed on Foo and Bar of either a technical or legal kind that would limit the exchange to Alice and Bob as the only authorized parties. The result is that now not only Alice and Bob, but in principle every potential user of Foo and Bar have also come into possession of the software. This is the secret behind what Eric Raymond identified as the "Magic Cauldron" (Raymond, 2001: 113-67) of open-source: a common stew, to which each contributes a small bit, yet is able to receive as much stew as personally needed in return, simultaneously and non-subtractively.

In a hypothetical assessment of fairness defined as equal exchange for groups larger than the most basic symmetric schema of Alice and Bob, we could make the following calculation: On the one hand, we must know the labour-time Alice contributed to coding FS (we ignore non-code forms of contributions for the moment). Then we make an inventory of every piece of FS that Alice uses which has been coded by other FS producers. For each item in the inventory, we assume we know the labour-time that was expended in producing the item and we also know how many copies of the item are globally in use. We divide the former by the latter to arrive at a per-copy value of the item[11]. We repeat this for every item in Alice's inventory of FS used, and add up the values and arrive at a sum. If the labour-time Alice contributed to FS is equal to this figure, we can reach the conclusion that the relationship of Alice to the community is fair on the basis of the law of value within a system of generalized reciprocity. With each new fair participant in the commons, the positive externality born out of the individual's socialistic "exchange" with the collective spreads throughout the system, as in the previous case of the one-to-one exchange between Alice and Bob.

This calculation of what goes on between Alice and the collective is all well and good, except for one problem: putting it into practice would be insanity. Not only would it prove utterly unfeasible to implement, it would hardly be desirable. We would need to install a draconian surveillance mechanism on each and every person's computer that would track every piece of installed FS on their system, as well as a mechanism that would track how much time they spend towards FS production. These would then have to be aggregated and constantly updated in real-time, accounting for the millions of hours worked and millions of installations of FS made every day. The end result would be a major disenchantment in the form of a number spit out on each individual's screen, stating their balance of account towards the commons, a quantified amount of credit or debt. And then what? Presumably after a certain period, a check or an invoice, followed up by enforcement, with all the nastiness that would go along with it.

The point of course is not to actually account for and guarantee such fairness in practice but to transcend it, in the sense that over time and across a large number of individuals, the principle roughly holds without conscious intervention. In fact, the power of the communistic FS mode of production is demonstrated precisely in its tolerance of individual cases of "unfairness": unlike traditional material commons, those who maintain a relatively one-

[11] The more general purpose the software, the more users, hence the per-copy value of the software tends towards zero. For certain software used in small niches that takes a large amount of labour to develop however, the per-copy value will remain non-trivial. This could possibly explain certain holes (for the time being) in the currently existing gamut of FS solutions, as well as very high price tags on their proprietary counterparts. Examples that come to mind are game engines, professional CAD-CAM software, and Non-Linear Video Editors.

sided relation to the commons do not have a subtractive effect – in fact, recalling the comedy of the commons in which each participant increases rather than decreases a common resource[12], they have an "unfairly small" yet still additive effect. The FS ethos therefore prioritizes the maximum satisfaction of needs instead of obsessing over equality of contributions. We settle for a subjective fairness that leaves it to the operation of the moral urge to reciprocate that springs from the individual's conscience and sense of appreciation, what Stallman referred to as gratitude, which proves to be enough to sustain the system. External motivations for contributing are highly useful but not essential for the system to function.

The real limitation in Circle 1 remuneration *in this historical phase of transition* is not the issue of fairness, but the in-kind nature of remuneration that takes place[13]. Only software needs can be met within this circle (or in the case of digital production in general, only needs for digital artefacts). Although the spending of individuals who benefit from the digital commons is reduced by the amount that they would have spent otherwise on proprietary digital artefacts (software, e-books, digital music files, etc.), they cannot pay for "food and rent" through the operation of Circle 1. They would have to either do something outside of FS to earn money as well and limit the time they spend on FS production, or engage in the higher circles of FS remuneration.

What has truly propelled FS from a small "gift economy" among programmers towards the cyber-communism (Barbrook, 2000) we now observe was the Personal Computer revolution. The PC revolution started in the late ‘70s, and exploded in the early ‘90s with the advent of the home internet connection. PC's meant that non-programmers would own their individual computers and use them for tasks other than programming. In fact, personal computer became ubiquitous in production, used in every sector of the economy. The software accompanied the hardware, creating a vast market for what used to be called "packaged software": binary-only software that does not include accompanying source code. The emergence of Microsoft, which focused on serving this personal computing software market, was part of the same trend. The PC revolution has been a massive democratization of computing, which created swathes of computer users that vastly outnumbered the number of user-programmers. The role of programmers shifted from serving each other in academic research facilities where a form of avant la lettre FS had emerged, towards serving mere users. The dominant form this service took has been the market-based proprietary software model, i.e.

[12] Carol Rose. The Comedy of the Commons.

[13] If we were to imagine a future society where most or all production is the production of digital artefacts, all remuneration could be in-kind, and the issue would not arise.

selling licenses for usage of binary software that came without accompanying source code. However, FS adapted to this new terrain, where a massive discrepancy exists between the number of FS coders contributing code to the software commons and the number of users benefiting from it. It is this new terrain that provided steam to what I call the Circle 2 of FS remuneration.

Circle 2 of FS consists of the contributions made by coders who are employed by an entity such as a corporation to produce FS because it furthers the commercial success of the corporation in a related field, or by a government which employs FS producers with a variety of economic or political motivations. A FS worker operating within Circle 2 is not concerned with what mechanism the capitalist, the government or other entity has devised to benefit from its spending on FS development; the relation of the entity to the FS producer is wage labour. This means that a category of FS programmers is compensated with money rather than with the exclusively in-kind exchanges of the previous circles.

Let me point out that this in general does not reduce the size of Circle 1. Even when individual contributors in Circle 1 are recruited by entities into Circle 2, they are naturally replaced by other newcomers. Furthermore, the volunteer phase of an individual FS coder may in fact have been motivated by expectation of future employment in Circle 2 as a result of volunteer work to begin with, a process analogous to the role internship plays and Stallman refers to as professional reputation. With the Circle 2 mechanism, individuals who at some point might otherwise have to abandon FS or greatly limit their contributions can be sustained. Also, there is the likelihood that the recruiters will hire volunteers to continue working on their existing project (where they have demonstrated their competence), rather than assigning them to different work.

There is a widespread tendency to see capitalist sponsorship of FS as proof of its capitalistic character, but this is a superficial conclusion which ultimately proves fallacious. Capitalist investment in FS production does not turn FS into commodities. Neither are the use-values seized by capitalists and made exclusive in any other way. Capitalist contribution to the FS commons is often a result of the contradiction between the interests of a particular capital and capital as a whole, where one capitalist is willing to reduce the total size of surplus value produced if this enables them to capture a bigger slice of the now reduced whole. Not to mention the fact that the particular capitalist may be dragged into FS production by the desire to build upon already existing Copyleft'ed code (to produce a derivative work).

This process generally follows the following pattern: FS, like all software, may provide a foundation for the complementary commercial business of selling "support and services" around it. All other things being equal, a company that can sell both software licenses and support and services will make more profits than a company that releases its software freely and restricts its profit generating activity to selling support services. All other things *not* being equal, it is of course possible that a FS producing company may in the end make more profit than a proprietary competitor. This may occur in the following manner: If the FS product is more or less on par with the proprietary rival, by virtue of releasing its software for free, the FS company will have a much larger user base that may translate into more users who go on to buy support and services predominantly from the FS producing company, which, as its producer, is likely to be the company with the highest expertise on that FS (even if not a monopolist). This and similar second-order mechanisms for making a profit in no way negate the fact that in the domain of the relevant type of software per se, profits will have been reduced and the market will have shrunk in favour of the free commons. This means that the wealth of society increases dramatically (more users accessing the FS due to zero price), and as a result, new needs are created, the size of a separate market, the market for support and services for the software at hand, grows. The net result to society is an increase of wealth when compared to the system of selling software licenses, which is characterized less by the creation of wealth, and more by the transfer of wealth from one group in society to another, i.e. rent.

The Circle 2 model allows economic independence to the FS contributor so that they may pay rent and buy food while contributing to the commons which benefits everyone. This is a good thing. Corporate or state direction of FS projects does mean, however, that some freedom in organizing the productive activity and defining its goals (the question of what to work on and how) must be surrendered to the corporate or state managers. The economic independence of the FS developer in this circle comes with managerial strings attached.

Personal economic sustainability in Circle 2 may thus come at the cost of a degree of *alienation* in the work setting, which is the characteristic feature of all wage labour. The mere fact that a labourer is paid a wage in order to produce FS as opposed to proprietary software cannot negate alienation in the production process, when it is managerially organized by corporate, governmental or other non-self-constituted entities. Even within the wage relation, FS does, however, have a tendency to *reduce* alienation compared to analogous proprietary software production. There are two factors that effect this amelioration in the condition of alienation in *FS production under external management:* the first is that the product by definition remains a commons, so producers are not alienated from the fruit of their own labour in exchange for the wage. The wage is received in addition to access to the product that is produced. The second is that FS production has to be open to some degree to the collaboration and contributions of a larger community. To have it otherwise would substantially defeat the purpose of engaging in FS production for the entity – a major objective for the entity is to benefit from free external inputs instead of developing a solution where the entire development costs would have to be internalized. Therefore, any FS project will engage to some degree in dialogue and partnership with the community in the way it organizes production, taking into account the needs and wishes of this community, of which the waged producers will comprise a (major or minor) subset. If this interaction is deemed to be dysfunctional by the community as a whole, the project's success will be jeopardized and may be threatened by the appearance of a fork. This ever-present pressure of community opinion and the threat that the community will vote with their feet in FS production provides a check on the amount of managerial fiat that a FS producing capitalist or state entity can exert on their waged FS producers (the same applies to any kind of leadership in FS projects). The managerial fiat will be diluted by influence from the community, in other words some characteristics of Circle 1 production will be felt in Circle 2 operations as well.

The interests of the sponsoring entity may overlap with the interests of the public fully or to a partial extent. This is a matter of the use-value of the produced FS. I intuit that as a general rule, the public good will be furthered by capitalist-sponsored FS in similar fashion to the way public goods produced by the capitalist state (such as roads etc.) benefit not only the capitalists but the population as a whole (assessing exactly who benefits how much can be difficult). In cases of government sponsorship (whether at the local, national or international level), the harmonization of public and governmental benefit will be dictated by politics (which no doubt is influenced by economics) rather than direct economics.

Circle 3 contains FS remuneration schemes where FS producers are funded by their users voluntarily and directly, without formalized procedures such as contracts or reviews of work performed. This remuneration is essentially in the form of donations and it can be seen as an example of collective patronage. Donations are made to FS coders either prior to or after/during (as software is rarely "complete", there is no clear "after" but rather continued development) the initial work of development takes place. The simplest form of donation is usually facilitated by posting a bank account number, PayPal button or other electronic currency id on a FS project website. Donors may sometimes receive notices of appreciation such as appearing on a ranked list of donors on a web page. Those that donate over a certain amount may also receive tokens of gratitude like swag items or their name appearing as patrons in the “about” dialog of a software.

As the donation model has gained traction and with the general proliferation of freelance work and the start-up phenomenon in the larger economy, innovations have taken place in the facilitation of donations. There are micro-tipping systems such as Flattr where the donor pre-allocates a certain amount of monthly donations in their Flattr account, from which donations are drawn in proportion to how many times the user clicks the Flattr buttons on various recipients' web pages during the month. The pay-what-you-want model pioneered by "The Humble Bundle" game sales is another variation on the donation model, where a purchase action is required, but the amount paid can be as low as a single dollar (the requirement forces the user to break donor-inertia).

Crowdfunding has emerged as a systematic method of pooling donations for projects that are in initial or ongoing development. There are a few variations of crowdfunding. In the Kickstarter model, work is premised on the prior promise of donations. The project often presents an introductory video and page explaining what the project aims to be, and a certain target sum of money and a duration limit to gather the donation pledges is stated. If the targeted sum for donation pledges is reached within the given time frame, the project is undertaken. When compared to the usual circulation of commodities in the market, this form of crowdfunding reverses the production first, sales second approach with a seek funding first, delivery second approach. It also shifts some of the risk of enterprise on to the backers because they cannot evaluate the finished product before purchase. In the case of crowdfunding of already existing FS projects, however, the risks are much reduced because trust has already been established and distribution is instant upon completion. In the Patreon model of crowdfunding, "patrons" pledge recurring donations to projects instead of the one-off model of Kickstarter. This increases certainty and regularity of income for the producer compared to impromptu donations. In the more micro-system of Bountysource, users of software post specific feature and bugfix requests and set a monetary bounty for their completion, which is awarded to programmers who complete these tasks.

The beauty of donations when coupled with FS is that each individual decides how much to give themselves, taking into account their own ability to pay. This is much nicer for the user than the proprietary alternative of a one-size-fits-all price tag which will be set at a revenue-maximizing level, shutting out those who cannot afford it. One drawback of donations for users is that it may not be easy to figure out just how much to give to what, which could lead to donor-fatigue / donor-cluelessness. Nurturing the sense of community between producers and users, as well as accounting transparency in FS operations and further systematizations of donation mechanisms can go a long way towards solving this. Another issue may be the relative difficulty of generating donor interest for non-user-visible FS projects. Donation sharing and kickback schemes between upstream and downstream FS projects are being put in place to alleviate this issue. A sore spot in these advanced donation systems so far is that the platforms are capitalist intermediaries which take a cut out of the donations. An obvious solution is for the FS community to produce non-profit or cooperative alternatives to these existing platforms. A non-profit clone of Patreon called Liberapay is one such attempt. This is a young field which is still seeing major innovations and the best models will be settled on with time[14].

The significance of the improvements in donation schemes and the development of a culture of patronage among the public for FS is that it provides a foundation for FS programmers to take a major step towards becoming a class of independent producers without relying on the mechanism of selling their products as commodities on the market. This means the coupling of extrinsic autonomy with an upswing in terms of intrinsic autonomy because FS producers who can fund their work through collective patronage can self-manage their own organization of production, without the bosses and managers in Circle 2.

This sums up the circles of remuneration for FS that have emerged up to now within the constraints of existing society. It is important to keep in mind that Circle 1 (which contains within it Circle 0) is the defining form of remuneration of FS as a mode of production because it is unmistakably stamped with its own internal logic. If the progress of FS and similar digital production models is not blocked by reactionary forces, the ultimate long-term historical trajectory will be towards the complete dominance of Circle 1, which is an idiosyncratic form of what Marx called the higher stage of communism (Rigi, 2013; 2014). Circle 2 is a product of the interaction of the emerging new mode of production with the old capitalist mode of production. While Circle 3 is a step towards breaking out of the capitalist mode of production, it is still of a transitional nature. The more human labour in production as a whole moves exclusively into the realm of producing digital artefacts through increased automation, the more relevant the FS mode of production will become and the more acute will be its contradiction with capitalism as a historical system. This will take a while, though. In the meantime, we need to keep thinking about the transition. One idea for accelerating this transition is to imagine the institution of a universal basic income as a Circle 4.

"A basic income is an income paid by a political community to all its members on an individual basis, without means test or work requirement" (Parijs, 2004: 8). There is a virtuously circular logic behind my argument. The already existing phenomenon of FS proves that individuals are producing use-values for society as a whole without necessarily being motivated by money; they tend to do it as a matter of self-realization. As a corollary, they deserve

[14] See Platform Cooperativism by Trebor Scholz for an introduction to the issue of capitalist platform intermediaries and emerging P2P cooperative initiatives intending to replace them.

being accommodated by society in a way that does not reintroduce alienation[15]. On the flip side, by easing the provision of their basic livelihood, the introduction of a basic income will allow more individuals to cross the threshold of economic security in order to be able to participate in FS. Under a regime of basic income, which is an elegantly simple demand, all kinds of free digital production would be boosted by new participants, and as of yet unforeseeable new instances of free digital production may appear.

Another beauty of introducing basic income to a world in which FS and free digital production is ascendant would be that it would not jeopardize the operation of any of the lower circles, although it would strengthen the position of the labourer in Circle 2 and Circle 3 by increasing their options. "Give all citizens a modest, yet unconditional income, and let them top it up at will with income from other sources" (Parijs, 2004: 7).

There are already instances of support for the idea of basic income in the FS and Peer Production literature[16]. Perhaps even more significantly, however, software programmers themselves seem to be arriving at the idea. In a recent essay on his blog titled "Funding FOSS", software programmer Noah Kantrowitz points to the "non-capitalist system" of basic income as "an end game solution". "Provide a basic standard of living so people that want to dedicate themselves to enriching society can do so without putting their own needs in jeopardy"[17]. While Kantrowitz approached the issue from the perspective of funding FS, I find it equally exciting that other software programmers are pointing to basic income as a logical solution to the contemporary looming problem of structural unemployment[18]. Acknowledging the role software plays in inducing technological unemployment, software developer "Jason" of the blog "Practical Elegance" published a post titled "Confessions of a Job Destroyer" addressing the issue:

> We (programmers) all are, on some level or another; we're taking mundane repetitive tasks and automating them with code. In a perfect world, we would be hailed as heroes, freeing the toiling masses from their humdrum routines to engage in more ennobling pursuits… but there's that pesky issue of needing an income. (...) This, gentle reader, is where I make the argument for a basic income. It's just common sense as the amount of socially necessary labour decreases with each

[15] Not every deserving contributor to the digital commons can currently make a Circle 2 or Circle 3 mechanism work for them.

[16] Cosma Orsi, 2009; Jakob Rigi 2014. Bauwens 2005. See also http://p2pfoundation.net/Basic_Income.

[17] https://coderanger.net/funding-foss/.

[18] http://blog.p2pfoundation.net/why-the-tech-elite-is-getting-behind-universal-basic-income/2015/02/24.

> passing year. (...) I'm a job destroyer, and I love what I do. Now if only we had a rational economy, I could stop having mixed feelings about the net effect of my work[19].

I expect that the demands for a basic income and reflections on the expansion of the FS mode of production will increasingly coincide. This hypothetical Circle 4 remuneration mechanism could prove to define the penultimate phase of information society on its path towards "fully automated luxury communism"[20]. Its realization, however, will require mobilizing the cultural influence of FS in order to express it in the sphere of politics, which could take a while. It would also provide a forward-looking solution to concerns with making sure that the capitalists contribute their fair part to the digital commons, because progressive taxation would provide at least part of the funding for basic income. Even more importantly, basic income would leave the communistic cultural experience in the FS mode of production undisturbed; no regressive introduction of the wage or equal value exchange. Furthermore, by looking out for the workers who lose their jobs due to the march of software-based automation (not to mention the software developers employed by proprietary vendors who could lose their jobs due to FS competition), basic income could be the quintessential "non-reformist reform" (Gorz, 1968) demand that unites the “bit-twiddlers” with the rest of the proletariat in a long march towards a post-class society.

REFERENCES

Barbrook, Richard (2000). Cyber-Communism: How the Americans are Superseding Capitalism in Cyber Space. *Science as Culture*. 9 (1): 5-40.

Barbrook, Richard (1998). The Hi-Tech Gift Economy. *First Monday*, 3 (12).

Bastani, Aaron (2019). *Fully Automated Luxury Communism*. London, New York: Verso.

Bauwens, Michel (2005). The Political Economy of Peer Production. *100 Days of Theory.* www.ctheory.net/articles.aspx?id=499.

Castells, Manuel (1996). *The Rise of the Network Society. The Information Age: Economy, Society and Culture.* Vol. I. Oxford: Blackwell Publishers.

[19] https://web.archive.org/web/20161022094048/http://decomplecting.org/blog/2013/03/11/confessions-of-a-job-destroyer/.

[20] See http://www.theguardian.com/sustainable-business/2015/mar/18/fully-automated-luxury-communism-robots-employment for this humorously serious proposition, and Aaron Bastani, Fully Automated Luxury Communism, 2019.

Free Software Foundation (2018). *What is Free Software*. https://www.gnu.org/philosophy/free-sw.html

Kantrowitz, Noah (2015). *Funding FOSS*. https://coderanger.net/funding-foss/. Merchant, Brian (2015). *Fully automated luxury communism*. https://www.theguardian.com/sustainable-business/2015/mar/18/fully-automated-luxury-communism-robots-employment.

Orsi, Cosma (2009). Knowledge-based Society, Peer Production and the Common Good. *Capital & Class,* 97: 31-51.

Parijs, Philippe (2004). Basic Income. *Politics & Society* 32 (1): 7-39.

Peer to Peer Foundation Wiki. *Basic Income*. http://p2pfoundation.net/Basic_Income.

Practical Elegance Blog (2013). *Confessions of a Job Destroyer*. https://web.archive.org/web/20161022094048/http://decomplecting.org/blog/2013/03/11/confes sions-of-a-job-destroyer/.

Raymond, Eric (2001). *The Cathedral and The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Sebastopol: O'Reilly and Associates, Inc.

Rigi, Jakob (2013). Peer Production and Marxian Communism: Contours of a New Emerging Mode of Production. *Capital & Class,* 37 (3): 397-416.

Rigi, Jakob (2014). The Coming Revolution of Peer Production and Revolutionary Cooperatives. A Response to Michel Bauwens, Vasilis Kostakis and Stefan Meretz. *tripleC,* 12 (1): 390-404.

Ristau, Julie (2011). *What is Commoning, Anyway*? https://www.onthecommons.org/work/what-commoning-anyway.

Rose, Carol (1986). The Comedy of the Commons: Commerce, Custom, Inherently Public Property. *The University of Chicago Law Review,* 53: 711-781.

Schneider, Nathan (2015). *Why the Tech Elite Is Getting Behind Universal Basic Income*. http://blog.p2pfoundation.net/why-the-tech-elite-is-getting-behind-universal-basic-income/2015/02/24.

Scholz, Trebor (2016). *Platform Cooperativism*. New York: Rosa Luxemburg Stiftung.

Stallman, Richard (1985). *The GNU Manifesto*. https://www.gnu.org/gnu/manifesto.en.html.

Stallman, Richard (2010). *Free Software, Free Society.* 2nd ed. Boston: GNU Press.

Wikipedia. *Attention Economy*. https://en.wikipedia.org/wiki/Attention_economy.