EDICIONES COMPLUTENSE

Sociología del Trabajo
ISSN-e: 2603-9710

PRESENTACIÓN

Del script a lo ágil: el trabajo en los call centers y la industria del software en Brasil[1]

Henrique Amorim
Sociology professor at the Federal University of São Paulo (UNIFESP) ✉
Guilherme Guilherme
PhD candidate at the Federal University of São Paulo (UNIFESP) ✉

https://dx.doi.org/10.5209/stra.105581 Recibido: 23-09-2025 • Aceptado: 25-11-2025

ES **Resumen:** Este artículo examina la prevalencia de la taylorización y la auto-taylorización en las industrias del software y en los *call centers*. Los *call centers* surgieron a partir de la precarización, arraigada en los principios taylor-fordistas. En contraste, el desarrollo de software parecía encarnar la autonomía y la creatividad, representando el "sector de alta gama" del trabajo vinculado a las TIC. A través de una investigación empírica y entrevistas con trabajadores en Brasil y Francia, analizamos los mecanismos de gestión y control en ambos sectores. Los hallazgos revelan una radicalización de las prácticas tayloristas en los *call centers*, caracterizada por el uso de software de monitoreo y la adopción del sistema de producción Toyota. Por otro lado, los programadores de software enfrentan metodologías ágiles y un control objetivo mediante softwares de gestión, que posibilitan la parcialización y la metrificación del trabajo. Los autores sugieren que, en el horizonte del trabajo en las TIC, se observa una tendencia marcada por la estandarización y la prescripción, donde incluso las profesiones aparentemente intelectualizadas, como la programación de software, se ven afectadas.
Palabras clave: Taylorización; trabajo digital; industria del software; control y prescripción; self-taylorization

ENG From script to agile: work in call centers and the software industry in Brazil

ENG **Abstract:** This article examines the prevalence of Taylorization and self-Taylorization in software industries and call centers. Call centers emerged from precarization, rooted in Taylor-Fordist principles. In contrast, software development seemed to embody autonomy and creativity, representing the 'high-end' of ICT-related work. Through empirical research and worker interviews in Brazil and France, we analyzed management and control mechanisms in both sectors. The findings reveal a radicalization of Taylorist practices in call centers, featuring monitoring software and Toyota production system adoption. Conversely, software programmers face agile methodologies and objective control through management software, enabling work partialization and metrification. The authors suggest that, on the horizon of ICT work, there is a tendency marked by standardization and prescription, where even seemingly intellectualized professions, such as software programming, are affected.
Keywords: Taylorization; digital labour; software industry; management; auto-taylorización

1. Introduction and methods

Discussions about the emergence of "immaterial labour" and "digital labour" based on the insertion of information and communication technologies (ICTs) in the productive and labour processes have given rise to several theories and thesis about the end of typically factory and manual production (Gorz, 2010; Castells, 2010; Lazaratto, 1996). According to those, we would be facing transformations that are leading to the formation of a society fundamentally distinct from the so-called industrial society. Among the symbols of this transition are the productive processes in which prescription of tasks, control and subordination over work would be fated to disappear, and would promote, at the end, the fading of taylorist and toyotist forms of production organization.

We can find a critical bibliography debating this idea, dedicated to demonstrate that the Call centers, despite relying on ICTs, exhibit a pronounced incidence of Taylorist principles. Notably, their workers, despite

[1] This work is supported by the São Paulo Research Foundation (FAPESP): Doctorate/Continuous Funding Stream grant #2020/15078-9; Auxílio Regular grant #2018/00531-0 and The Brazilian National Council for Scientific and Technological Development (CNPq/PQ), grant 311116/2023-0.

utilizing computers and telecommunications as labour tools, perform repetitive tasks with standardized routines prescribed by management (Bain et al, 2022; Venco, 2006; Dutra, 2014; Braga 2013). According to Braga (2013), Call Centers pose a "setback" against the theses of post-industrial/informational society, representing a comeback for Braverman's (1974) general argument of a tendency towards partialization and standardization of labour.

In our findings, more than being a "setback" and just reproducing the taylorist logic, the call centers radicalize and deepen this logic when we notice: 1) the monitoring of the workers made through software deepens control when it comes to productivity (goals), to the workday supervision (working time and breaks) and to verifying the complete and perfect application, by the worker, of the *script* prescribed by the management; 2) this *script* is an ultimate synthesis of task prescription; 3) the presence of aspects of Lean Production/Toyota management, such as teamwork, collective goals, self-taylorization[2] or self-management and the quality control made by the worker himself / herself furthers management capacity.

A clear example of this accentuation can be seen in the *login* system. The worker is required to log in a "digital time clock" with his password every time he needs to enable his workstation. But this goes beyond simple control of entry and exit of the working day. The login software enables the management to verify, with accuracy, each break the worker takes, whether for eating, washroom, or for any other reason. More than this, based on this digital control resource, it is possible to constrain the worker when the prescribed time for the breaks is exceeded.

A further radicalization is the adoption of team-based work organization, inspired by the Toyota Production System (TPS) used in automobile assembly, now applied to call centers. This approach creates strong peer pressure, as individual call center agents who fail to meet targets (e.g., call time, sales) jeopardize their team's collective productivity goals.

In summary, the organization of the labour process at the call centers is based on the reproduction (and augmentation) of taylorist precepts (with labour fragmentation and rationalized control, now also realized through algorithmic management), added to the presence of TPS precepts of the typically factory-manual production.

This raises the question of whether this adoption of traditional labour management and organization methods extends beyond call centers to other ICT-based workers. Specifically, does software programming, an exemplary post-industrial occupation (e.g. Gorz, 2010; Lazzarato, 1996), fall prey to the same logic? Is it accurate to affirm that, on one hand, the call center agents have their work prescribed and standardized and, on the other hand, the software programmers rely on creativity and productive autonomy as their central productive force? Does the trend of standardized work processes and task division under ICTs apply even to those considered creative professionals?

It is common to find, as we mentioned, authors that present ICTs as the "carriers" of a new society (Mason, 2015; Varoufakis, 2017) where work is creative and workers finally recovered the "know how" over the totality of work processes and from whom would be demanded not only a technical formation, but also a "vernacular-knowledge"[3] (Gorz, 2010). However, if at first something similar to "handicraft work" occurred among the software programmers, mostly in the 1960s in the United States, where the software production was still initiating (Cusumano, 1989), soon were developed new productive processes capable of systematizing work, divide it into different tasks and prescribe it to the workers as a routine and serialized process. First, the linearity of work processes is consolidated, based on "waterfall" software production, with, as Miguez (2017) observes, evident fordist inspiration. Thus, a work script is designed with steps that must be performed in sequence.[4] After, in the end of the 1990s and especially after 2001, with the publication of the "Agile Manifesto"[5], the agile methodologies acquire prominence, making the management process more accurate, basing it at the workers' self-management in the software production.[6]

Although those methodologies, also known as *Lean Digital,* are pointed out by the management manuals as a methodology that "empowers" the workers (as analyzed by Guilherme, 2020), supposedly giving them autonomy and creativity, researchers like Schumacher (2007) and Hodgson and Briand (2013) question this based on the analysis of so-called "creative" industries using these methods. That was also object of our empirical research, which showed us that the software developing labour, under the agile methodology logic, is monitored very accurately, either "virtually", based on specific softwares designed to control the work

---

2 In summary, management, as the objective rationalization of work, is internalized by the collective of workers. That does not mean that the direct and external control of labour process is missing, but, "(...) there is a double subordination of labor to capital, first to the precepts, rules, functions, and strategies imposed (...) and second to the subjective persuasion that ends up creating internalized management in the worker " (Amorim and Grazia, 2021: 12).

3 Gorz uses this expression to refer to the "cultural background" of the immaterial worker that would be necessary to this new form of production. Companies would demand all the worker's life experience, his accumulated knowledge including at leisure moments. That would be right the opposite from what Taylorism demand from the workers.

4 In general, the productive sequence of the waterfall model is: Requirements definition → Project → Implementation → Tests → Maintenance (Sommerville, 2011).

5 In this "manifesto", 17 developers, CEOs and managers list some principles and values that must guide the process of agile work. Based on this manifesto, most management manuals are published, and the main work management methodologies are formulated. Available on www.agilemanifesto.org. Accessed July 2024.

6 In the process of being adopted by software factories, these methodologies are being spread by a process of productive standardization and also by "evangelists" (corporate slang) who "preach" their use in courses and consultancies. Those evangelists are dedicated to "spread the word" of Lean Digital.

process, as well as by a self-control and self-management process very close to that from the Quality Control Circles of the automobile's assemblers.

The advance of management over the software programming's work, especially alongside tendencies of prescription with standardization and further task division in the work process are, as we will develop in the following items, opposing evidence against those theses that advocate the immunity of this kind of work to the capitalist management.

Given this brief introduction, it seems important to specify that our fundamental goal is not to make a simple comparison of labour as distinct as those of the call centers workers and software developers. This study critically investigates the standardization and fragmentation of work processes in ICT-mediated sectors, exploring its broader applicability to ICT-facilitated production. We comprehend that fragmentation of tasks within production processes serves two essential purposes: it allows capitalist management to retain the ability to combine the labor of various producers while simultaneously weakening the political power of workers and their capacity for collective organization. Thus, deepening the fragmentation of labor is not solely a strategy to accelerate the extraction of surplus labour. It also aims to disrupt the working class and its forms of resistance, whether expressed through political parties or labour unions. Nevertheless, our analysis reveals that ICTs do not inherently promote taylorization neither fragmentation; instead, ICT workers face labour management strategies akin to those experienced by workers under capitalism.

Our findings result from empirical qualitative research with the call center workers and the software programmers, in a field that was divided between Brazil and France. Based on semi structured questions as support, we interviewed, in what comes to the call centers, 10 working men and women, distributed in different positions: active callers, the receptive, the *backoffice* and the team supervisors[7]. In what it refers to the software programmers, 22 were interviewed. Equally supported for a semi structured guide, we interviewed workers who are testers, technical leaders, IT managers, *scrum masters*, and programmers, besides the "preachers" (or evangelists) of agile methodologies[8].

2. Management by teamwork

Are you happy with your job?

"I am, but I work too much." (Programmer/*Scrum Master*, Paris, 35 years old)

To start the analysis of the work management in the two professional categories discussed in this article, we will use as *an entrance door* the organization of work through teams. As we stated in the introduction, this is a fundamental aspect of *Lean Manufacturing*, or TPS, and it is present both in the agile methodologies at software production and at the call centers. Through the approach of teamwork in the working sites, more accessible at first sight, it is possible to go further in our analysis on other work management aspects among the call center workers and software programmers.

Some authors have already analyzed the composition of work teams in lean production (e.g., Malaguti, 1996; Antunes 1998). The formation of teams headed by leaders who are constantly pressured to meet goals lead to an increasing productivity. Such logic imposes upon the workers a "performance control" of their workmates, which establish a workers' led management among themselves and self-management of each worker. The imposition of goals (even if they look like self-goals decided by the team, as it happens with the software programmers), ends up establishing a self-control of the worker himself/ herself and a mutual control between the workers and their teams. In plenty of cases, it is possible to notice *competition* among workers, which Malaguti (1996) analyses under the idea of "professional pride".

With this work dynamic there is an *apparent* decrease in direct supervision and work's surveillance to the extent that workers recognize themselves as responsible and co-authors of production. Andy Friedman (1977) named it "Responsible Autonomy", questioning the content of this supposed autonomy[9]. The tasks' coordination becomes an internal activity to the collective of workers, and productivity becomes a collective result[10]. As this teamwork method became popular with the TPS expansion outside the automobile industry, notably its use can go beyond "material" production. It is the case of work in call centers.

---

7 The active workers are responsible for making the most varied contacts (offering products, collecting money, advertising, for example); the receptive workers receive calls, generally from the SAC (Customer Service); *backoffice* is responsible for the continuity of the service after the first contact.

8 The testers are responsible for finding mistakes or *bugs* in the programming structure of the product; the leader elaborates the software conception that needs to be produced based on the client's demand and then distributes it for his team; the *scrum master* is some type of team's manager, and he supervises and coordinates the workers individual labour and, finally, the "preachers" as we've said, are the ones dedicated to "spread the word" of *Lean Digital* through training, lectures, videos and podcasts, publicizing the agile methods. We describe some of these functions in more detail in the next item.

9 For the author, there is a maintenance of managerial authority, since the workers identify with the competitive aims of the company. He also points out that, if the initiative of diminishing direct control comes from the management, it will probably result in greater managerial control over the labour process as a whole.

10 To Marx, capitalism is constituted when the same individual capital employs simultaneously a given number of workers. Marx calls this form of labour, in which many work in a planned manner, side by side and jointly, "Co-operation". For him, co-operation increases the individual capacity of workers' performance, achieving more than a simple multiplication of their isolated and individual efforts, thus creating a collective productive force that is not paid by the capitalist either (Marx, 1982). Teamwork, therefore, within this same logic pointed out by Marx, intensifies this aspect of labour exploitation present throughout the history of labour in capitalism.

In our empirical research, we have noticed that everything in these companies is based on teams, each team dedicated to the same task. For example, all the workers of a given team in the active calls sector are responsible for selling internet and data packages for phones. They all must perform following the same *script,* aiming at the same objective: selling and, at the end of a certain period, reaching the goals previously established. The displacement of workers between different teams and functions may occur, but it is not common. The "rotation" necessary to the call centers seems to come from the high numbers of *turnover* found in this sector. According to Venco (2006), Braga (2012) and the report from *The Global Call Centre Report* (Oliveira, 2006), in Brazil, in a one-year period, almost all workers within a call center are renewed.

In our interviews we found many mentions that account for the effectiveness of teamwork in management and work control. For instance, a worker (in active sales, working for a cell phone company) described to us how rivalry between the working teams was encouraged to enhance productivity. Each team has their own "battle cry", intoned as a form of self-affirmation against the other teams at the moment a goal is reached. According to his words, "*this thing gets into the people's minds: [people say] 'We must do it for the team, we need to reach the goal, and you must come to work. We MUST reach the goal, the team NEEDS you'"* (Call center agent, São Bernardo do Campo, 28 years old).

Although in appearance and in the company's discourse such mechanisms are "ludic" resources, with the objective of integration and even for fun at the workplace, in practice they reinforce discipline, sometimes by causing embarrassment to workers. In this sense, another example came from this working team: for each sale, the worker responsible for it would sing a song, based on a *pagode* (traditional brazilian music rhythm) from the 1990's, which begins in the call center's version with "*there is one more sale to celebrate".* At this moment, the other workers of the team were allowed to mute their phones for a second to sing along with the song, making *backing vocals,* saying *"you may celebrate, you may celebrate".* The practical effect, beyond the supposedly fun at the workplace and integration of the team, is as follows: every working man or woman knows who *chants the most* and *who chants the least* that song. It means, therefore, knowing *who sold the most* and *who sold the least,* making clear for all workers who raise the team's goals and who *pushes the team down*. As the worker summarized: *"(...) it creates a bad atmosphere for the person who does not sell. It is the same as if guys put a sign on you that says: 'this person does not sell" (*Call center agent, São Bernardo do Campo, 28 years old).

The pressure to *"not let the team down"* and to *"help"* achieving the collective goals acts on the workers very strongly. Analyzing this context, we believe the call center's work organization is beyond simply taylorist: it goes through a self-taylorization process of the work's collective, at the pace that, through these management practices, it is also founded in a form of compulsory persuasion of these workers, based on a coercion-consent game which acts based on an obligation form of the call center agents in relation with their working teams.

When it comes to software programmers, Lean Digital methodologies, as the name suggests, emerge as a managerial practice adapted from various elements of Lean Manufacturing to digital production. In other words, the toyota production system is used in software development to flexibilize production, reduce costs and intensify work, increasing productivity.

As it happens in the call centers, one of its organizing principles is the formation of work teams. Managers and workers call those teams "*squads",* composed of a few different roles, such as *Scrum Master*[11]*,* the technical leader, the programmers and the testers, (there might be variation of these names and teams' composition according to the company). It is essential in this methodology, as one of the workers said, the "*diverse roles inside a squad (...) each one aiming for a specific part. In the end, at the final delivery time, everyone is out there on a Friday night hoping it works out, for each one has done a piece of it and will put everyone's labour together."* (Technical leader, São Paulo, 30 years old). Thus, the labour is based on «(...) *multidisciplinary teams in a customizable line of production to the different kinds of software*» (Amorim & Grazia, 2021. p. 9)

While at the call center the goals are based on simple metrics, such as the average time of calls or a certain number of sales, the agile methodologies, theoretically, organize working processes that are less quantifiable than those established by the call center metrics. However, according to our empirical research, the agile methodologies make it possible to fragment the labour process and divide it in small production cycles, also easily verifiable.

At each new project, following most of the *Lean Digital* techniques, programmers are gathered, and the tasks are distributed. From this point, the *Kanban* - through communication cards[12] or *softwares* that

---

[11] *The Scrum Master* is a manager in companies using *Scrum* to organize the production, work processes, and, above all, organize the working men and women. This agile methodology advocates a set of steps for programming software, forming multidisciplinary teams in a production that can to the different types of software demanded by customers. In short, we are talking about a way to organize workers that has as its backdrop the need for adaptation and flexibility of work. The worker must, therefore, follow a certain type of conduct, as a kind of play in which different roles are assigned to the programmers, the client, the scrum master, and the manager (Amorim; Grazia, 2021). The *scrum master* has the role of ensuring the correct functioning of the Scrum and the software to be developed.

[12] Visiting companies that use *Kanban,* we could verify the use of large glass "murals" in which, in front of the whole team, tasks are set by using post-its that indicate what and when they must be done. The spaces to be filled in the board according to this technique are: TO DO → DEV → TEST → DONE. The administration and management manuals of the software teams' work usually call this model *Pull System*. According to the very conception of *Kanban* the "empty space" ahead "pulls" the accomplishment of the previous task.

performs a similar function, are used to distribute tasks among the programmers. Thus, according to Amorim and Grazia (2021), *Kanban,* in the software industry, has a different function from that one of the TPS automobilist industry, being used not as a stock control technique, but as a working management technique within *Lean Digital*, to organize and control the constant cycles of production and delivery of software.

When it becomes possible to break the labour process into smaller tasks and distribute them among the workers, it also becomes necessary to manage the execution of each one of these tasks. First, a prior estimate of the work demanded is made for each one of the tasks predicted at the beginning of a new project. At this moment, software programmers, in the same room with their managers or *Scrum Masters,* must stipulate the specific working time for each task. The tasks can be scored, for example, from 1 to 20, where 1 is the simplest to be done and 20 the most complex ones. From these two extremes, it is evaluated, by comparison, the complexity of each one of the tasks which must be performed. In practice, there is:

> (...) a joint evaluation... developers and the *Scrum master* in the room, we vote and see what the average points of each task are. If one person voted too high, or someone too low, he must explain the reason why he thinks he is so different from the majority (Software engineer, Campinas, 35 years old).

This technique is also known as *planning poker:* each worker stipulates the grade for each task and, writing on a card, everyone puts their cards down on the table at the same time. Based on this process, it is possible to cause some embarrassment to workers so that they reduce their expectations of "working time" regarding the execution of each task at the moments of *planning:*

> Let's say that I'm kind of led to say it will take me 20 days [to accomplish a task] because I know the person [supervisor] wants it fast, because I know it usually takes people 20 days to do it, because I know if it takes me longer than this, they will think I'm fooling around... (Programmer, São Paulo, 23 years old)

Besides this initial planning meeting, daily meetings are also very common. Such meetings are usually 15-minute events in which the programmers update their team and managers on the stage they are on their tasks. As one interviewee suggests: *"(...) we call daily meetings in order to know what each one needs that day, what he intends to do and if there is anything in his way (...) these are the things we do in a daily basis"* (Startup programmer, São Paulo, 30 years old).

With cohesive and self-managed teams, having in mind the accomplishment of the tasks and the finalization of the product, it is possible the structuring of a management that fantasizes itself with a discourse of autonomy, freedom and flexibility in favor of the worker. The role of the supervisor, for instance, whether by any of the names by which he presents himself - *scrum master,* project manager, *tech lead* - would be "(...) coordinating tasks, determining time and motivating the developers (...) *a boss who neither gives orders nor waits for the directions".* He must «*(...) try to prevent the developers from delaying their tasks, and encourage them*» (Miguez, 2017: 47). When we interviewed an IT manager and preacher of the agile methodologies, he told us that the central point to management is to create "(...) *an empowerment of your own team",* since "(...) *the idea is that the leader, like me, do not need to keep telling the team what to do. They do these arrangements among themselves"* (Brasil, 45 years old).

Kunda, at the early 1990's, had a similar finding when analysing the pioneers of what he called Engineering Culture in the tech industry. Doing qualitative research, one of his interviews synthetize what we are referring: «They have to want to. So you have to work through the culture. The idea is to educate people without them knowing it» (Kunda, 2006: 5). In this sense, we point to the self-taylorization of these workers as an important managerial aspect, aiming to constitute a self-control and self-management process that is based on a relation of coercion, convincing, and, at the same time, consent. As it is, the individual or collective worker learns/internalizes elements that are proper of taylorist management as regulator elements of his own work and based on them, acts in a proactive way in the consecution of management goals. As one interviewee pointed out: "*(...) I always come in front of my manager with a proposal: "I want to do this, are you with me?", instead of waiting for the managers to decide for me (...)"* (Programmer, Paris, 35 years old).

The conformation of a team made by self-managing workers, who perform interdependent tasks is part of the process of building consensus within Lean Production practices, and, as we observed, is also present in production based on Lean Digital. In this way, agile methodologies and labour organization through working teams allows a self-control of these teams and a permanent contact with the supervision, to "improve" (intensification of *cadence and movement)* the work processes, through suggestions from the workers themselves, and to monitoring the progress of the prescribed tasks, even though this management does not look like an external control to the worker.

Teamwork and Agile in the software development, therefore, are central elements of a production that radicalizes the prescription of work and its taskification, either through self-management, proper to the structuring of teamwork, or through a more direct supervision, in daily meetings and through softwares. These elements, although at different levels when compared to the call center teams, which work is based on *scripts,* synthesize a first approximation of the call center workers labour in relation to that of the software programmers. While among the first ones, the strategies and techniques of control and management seem more coercive and direct, among the last ones - although still coercive - these strategies and techniques are based, as we said, on a compulsory persuasion**,** in which the work prescription done by the management pretends to look like autonomy.

3. Customers and work control

In both cases, at the call center and at the software industries, another managerial element is the active participation of the client. In a context of work projectification[13] and outsourcing, work and its rhythm and intensity are increasingly based on arrangements between contracting companies and the management of the contracted companies.

Notably, "clients" does not refer to the final consumers, but rather to the contracting companies—the corporate clients of call centers and software firms. In the case of telemarketing, these are the companies that hire the call center service; in the case of software development, they may be external firms or internal departments, in the cases which software is developed "in-house". The figure of the client is thus characterized as part of the dynamics of work. In the case of call centers, the hiring customer is "at the top" of a pyramid that exerts pressure over the call center workers. Starting from the goals stipulated by contract with the client, there is a cascade effect of the supervisors over their working teams in order to accomplish these contractual goals. In short, the external customer pushes the outsourced company; its management does the same with the coordination, which pressures the supervision and these, at last, pressures the call agents. Venco (2006), in that sense, points to a pyramid of coercions with cumulative effect, which would determine the pace of work.

This presence of the client inside the call center is so remarkable that, during our empirical research, we noticed the workers refer to their own teams by the name of the company contracting the call center. Thus, although the workers are hired directly by *Atento*[14]*,* they refer to their teams as "Bradesco team" or "Vivo team", for example[15].

This form of structuring the call centers is the synthesis of a process that involves working laws deregulation and standardization and disqualification of work. In an interview with a worker from the call center of *Orange,* a french telecommunications company that emerged from the privatization of *France Telecom,* we could notice some evidence of this process and work precarization as its consequence. This interviewee, once an employee of the State's company (since 1979) and a remaining employee after the creation of *Orange*, describes how a process of outsourcing the most repetitive and rhythm-intensive tasks took place to contracted call centers. These were transferred for hired call centers, some of them in former french colonies, such as Morocco and Tunisia. The interviewed points to a continuous cutout of *Orange* workers, based on a managerial restructuration in which a process of partialization of work ends up transferring the tasks' execution, simplified after disqualification and standardization of work processes, for outsourced call centers. This teleatendant, a technical staff member at *Orange,* observes this process in his very own job:

> They try to frame us with a script. (...) Putting questions in chronological order, which is very embarrassing for the people who were technical, once they have a *know-how* (...). What matters to the employer is to extract the workers' *know-how* so that things get standard, and everyone can operate the same way. This is the trend. (Call center worker, Paris, 57 years old).

At the end, he also notes that these outsourced call centers have worse working conditions when compared to the average working hours and average payment in France. According to him, the outsourced workers "(...) *have a higher intensification of work. They must answer more phone calls and make them shorter".*

We can note, therefore, based on the relation between task simplification and outsourcing, that the contracting company stipulates a quantity of sales, charges and average time goals for each call, imposing it on the workers who are now sub-hired. This relation is made explicit in the description of the pressure chain in a big brazilian call center:

> Our goals come from the Average Time of Service, which is 9 minutes. (...) there is some pressure over it because of the contract: if we don't reach the goal, we lose money... It is part of the contracts. We are pressured to reach this average. (Call Center supervisor, São Paulo, 26 years old).

The prior establishment of goals in the contract is presented by the call centers as a "plus", meaning an advantageous characteristic offered at the moment of competition for new contracts. A manual from SEBRAE[16] dedicated to support the formation of call center companies points at this aspect. For a call center to stand out from the competition and attract the targeted-companies, SEBRAE recommends, among other aspects, the "making of corporate contracts with performance percentages".

In this same sense, another "plus" which is usually presented, and directly causes pressure over the workers, is the possibility that the hiring company may verify, in real time, the work of the call center agents

---

13 Kalff (2017) identifies projectification as a concret form of work organization and the most effective for the needs of what would be informational work. Through projects, it would be possible to organize labour in a structured and disciplined way, at the same time in which there is a veiled coercion imposed on the worker under the form of self- responsibilization. Besides, projects are central elements of the productive processes of the companies whose production is based on the demand of other companies for specific products, as it is the case of the call centers and industries of customizable softwares.

14 Call Centre company that arrived in Brazil linked to Telefonica S.A., Spanish multinational corporation that began operating in the country after the privatization of telecommunications in the 1990s. Initially, Atento's unique service was Telefonica's demands, but it began expanding its clients. Sold to a US group in 2012, nowadays the company has 40 call centres in Brazil and approximately 75.000 workers.

15 Bradesco is the second biggest bank in Brazil. Vivo is a spanish corporation, that possesses concessions in internet, cable TV, phones and mobile.

16 Brazilian Micro and Small Business Support Service is one of the main entities in the country dedicated to fomenting entrepreneurship and competitiveness, being very influential over the constitution of certain characteristics of the market in Brazil.

who provide the outsourced service. So, work fiscalization and control have a triple dimension: there are intern supervisors; external supervisors linked to clients and self-management among the agents in the team.

Among the software programmers this logic is repeated. The customer is also a central part of the work organization. In many software industries there are specific working functions that aim to "build the bridge" between the customer (extern or intern)[17] and the developing team. It would be, as the management's terms say, a "translation" activity of what the customer wants to the software programmers.

It is from this initial demand, therefore, that the product is elaborated. Motim and Rodrigues (2018) observe that «(...) *the customer has no technical knowledge, but his interference is very incisive»* (154). This, according to the authors, interferes with the deadlines and in the product itself, since the conclusion of the product depend on the acceptance from the client. Hodgson and Briand (2013) had similar findings in another study. As they point out, the client set the policies and establish very clear guidelines concerning the delegation of authority. In their studycase, the company had to elaborate periodic reports on the progress of projects, pay scales and even the expense accounts.

Besides, one of the main aspects of Lean *Digital* methodologies, as we have already mentioned, focuses on delivering the software in parts or stages. That way, instead of an unique delivery at the end of the process, it is constant for the programmer the necessity of showing partial results to the customer at certain time intervals. This method ends up by elevating the client to the figure of an "external manager", once he follows, if not in real time, in very short periods of time, the whole execution of the product he bought[18].

As a matter of fact, this theme appears frequently in several managerial texts destined to software companies. In a document elaborated by *CI&T*[19], with the intention to teach, publish and universalize agile methodologies, we find several passages stating that companies that «*will remain in the game»* will be those that can «*(...) learn fast to deliver value to the customer constantly».* That way, companies should develop «*digital anti-fragility»,* based on «*less software».* In other words, companies should «*focus on developing only what meets the needs and solves the customers' problems, without wasting time with functionalities that will not generate value to the user experience»* (Oliveira, 2017).

Thus, in addition to providing greater flexibility and speed in the development of the digital solution, this form of workforce management aims to reduce the amount of necessary code. So, still according to CI&T manual, the frequent contact with the customer and the software development process is directed to «*simplifying the management»*, which favours a distribution of prescribed tasks to the programmers, after the conception of the product has been made by the management in contact with the customers. It is, therefore, a more standardized and verifiable software programming work process in its execution according to the periodicity agreed with the client.

It is inside this logic of *"less software"* and constant deliveries that the MVP, *Minimum Viable Product,* appears*.* According to the same document, the MVP must have a sufficient value to be used (that is, it must present some active functionalities to be used) and still create a "feedback cycle" with the customer capable of guiding its future development.

For us, more than the "transparency", this relation provides a *managerial just-in-time,* not only of the delivery of the product, but also of monitoring and management of the labour force. The "quality control" and the external supervision occur constantly throughout the productive process. It is periodic: daily, in the case of the daily meetings of the programmers with their managers, but it is also constant between the work teams and the client company, starting at the Minimum Viable Product (MVP) and following each partialized delivery of "value" to the "client"[20]. By this description, we want to demonstrate that control over work can be concretely realized in a periodic manner based on the delivery of small parts of the project under development.

However, besides this periodic verification, performed at each delivery, real time monitoring is also present, with control over the execution times of the tasks. Such control might be executed through specific softwares that distributes and monitors tasks execution. As we said before, the *Kanban* itself, in some companies, is applied by software programs used to manage the working teams. We found, in our empirical research, several mentions to softwares with this function. One of the most mentioned was *Jira Project Administration;* a software that promises, according to its creators, to *unleash the power of Agile"*[21]. Based on it, it is possible for the management to monitor the progress of every programmer in their tasks.

In short, the structuring of the project according to the *Lean Digital* and the division of tasks through *JIRA* or other similar programs follow a very similar script: the project manager, (or *Product Owner* or even the technical leader, depending on the company), in a direct and constant relation with the client who demands the software, outline its general characteristics and perform the division of the work among the development

---

17 Many times, the "customer" for the IT team is inside its own company, in the case of IT sectors/departments inside companies whose function is other than the software development. To this production, the specialized publications call *Software in house.* The companies dedicated to the software production, hired by other companies to develop them, are usually called *Core IT.*

18 As described by a software engineer about the follow-up that the customer can do on the projects: "*I think that for the customer, the best thing is transparency. He can measure the progress of his project every two weeks, or by sprint (...) he can have better and faster feedback" (*Software engineer, Campinas, 35 years old).

19 *CI&T is* a brazilian software company located at the technological hub of Campinas/SP and is part of *Agile Alliance,* an organization sponsored by many companies of the sector dedicated to spreading agile methodologies.

20 According to workers and managers vocabulary, "value" is a part of the product to be developed which, however, is already measurable or "verifiable" after every *Sprint*.

21 This program was developed by the australian company *Atlassian* and released in 2004. Its function is to manage the team work for software development. Available on: https://www.atlassian.com/br/software/jira.

team, following the precepts of agile methodologies. These teams work with deadlines for different parts of this software, being submitted to the periodic supervision - daily meetings, weekly meetings - and real time supervision, through softwares that help in the metrification of the programmers' work.

This renews the taylorist principles of separation between work conception and execution, a dichotomy thought to be obsolete, particularly in light of ICT-based work processes. A technical leader of a programming team, when describing his work routine to us, approaches this separation in a very didactic way: "I work with a group of developers, so I work more at the technical leadership part*, I have more contact with the direct customer (...) and I make some technical understanding of it and, after that, I gather with the developers and they take it from here".* This technical understanding is the very conception of the program to be developed. This technical leader, having as reference the customer's needs, does a pre-elaboration of the product. And he goes on: "*Nowadays I rarely put my hands on the coding process. What I do most are the tests of concept and understanding with the customer"* (Technical leader, São Paulo, 30 years old). Therefore, his work is focused on the product's conception and its prescription, as tasks, to the working team.

The interview of an IT manager who went through the entire transition of the company from waterfall methodologies to the implementation of *Lean Digital*, addressing the programmers' reaction, corroborates the above. According to him: "*It's hard for the programmer to think in a fast environment which is broken in small tasks.* [They complain] *"Oh, but I don't know what the whole project is".* The interviewed, then, concluded, simulating his answer to these inquiries: "*You don't need to know the whole project. The person who does the planning needs to know the whole project"* (IT manager, Curitiba, 45 years old).

For us, it is possible, based on these elements, to present another characteristic that approximates the work at a call center to the one at a software company from the point of view of the tendencies of work management: the *script.* At the first, it is an important instrument of attribution and prescription of tasks. It does not only prescribe phrases or solutions expected for the attendances, but also indicates speech modulations, simulations of emotions through the voice, forbidden terms and other prescriptions, such as that the worker should never turn off the phone before the customer. As one of our interviewed states: "(...) *besides being repetitive, each person that you answer, you need to have the so- called 'smile in the voice'(...)"* (Call center agent, São Bernardo do Campo, 28 years old).

The supervision ensures compliance with the script through monitoring tools: real time verification softwares; post monitoring the recordings of the workers' attendances; the control done through the individual and team evaluations. Even though it is not possible to affirm that the instructions given by the software programmers' management to the programming team have the same level of control and prescription of the *scripts* formulated at the call centers, they are formulated based on the same principle, that is, the project is previously defined by the management together with the customer.

Similarly, just as call centers establish average call times through contractual agreements, software companies' customer projects serve as benchmarks for the work processes of software programmers. In the words of a project manager:

> This definition stage is important exactly for this: we sit down, assemble some documents together, including screens, texts, functions, everything as complete as possible, and we sit down and validate it with the customer. From then on, we already have a direction to base our development on (Project Manager, Minas Gerais, 32 years old).

This "direction" may, in certain work processes of software programmers, be similar to a division and prescription of programming phases in tasks previously structured, as we stated before. In this sense, we remember the following line collected in an interview, provided by a programmer who, once promoted, became responsible for the requirements analysis of the company and, in order to do that, he also conceives the projects to be developed:

> I make the system's screens; the developer just looks at the screen and tries to do the same. And that makes his life much easier, *the product is already born as correct as possible.* (...) I give it to them "chewed", telling what must be done, where to do and when to do. The guy just needs to keep the ball rolling (Project Manager, Fortaleza, 30 years old).

In both working processes, the customer's figure, who demands the call center service or a software, is a constant presence and an organizer of the working processes, having influence, consequently, in the rhythm and in the working conditions. Moreover, based on this relation, the concept of each project is made between the hiring company and the management of the hired company. At this moment, boards, *scripts* or models to be followed by the workers are elaborated.

In addition, in the context of outsourcing it is also important to highlight the role of certifications such as those from the International Organization for Standardization (ISO) and the Capability Maturity Model Integration (CMMI), which are key requirements for companies seeking to compete in the software production market. As Castillo (2007) demonstrates, behind the virtually mandatory adoption of these certifications lies a process of labor deskilling and the routinization of software development.

Even though the speech of autonomy and flexibility are the center of the agile methodologies' "ideology", we believe we have pointed to several aspects which situate such methodologies as a management form of toyotist inspiration, which intensifies the work and tends to partialize it in smaller and standardizable tasks.

In these terms, it is impossible to disentangle the work, the intensity of labour exploitation, and the individual and collective accountability of programmers from the management/client relationship and from competition between software industries.

4. Conclusion

We highlighted some points we consider providing basis for a possible comparison of how the management actuates over the work at the call centers and over the software programmers' work. A comparison that sought to point to how the presence of taylorist and toyotist management has been reproducing, stating clear the prescription of work as central characteristic of the work mediated by the ICTs.

Through the analysis of these two professional categories, apparently very distinct, but approximated by the tendency of prescription with standardization and partialization of work, our aim was to get critically into work's management territory. On this ground, teamwork is a central element for subordination of workers' collectives. From this initial prism, it was possible to go through other mechanisms of work management in these two professional categories, showing how work mediated by the ICTs follows the same logic which directly operates over the collectives of work and that underlies the typically industrial production since the introduction of machinery and that now goes deeper, trespassing the borders of production considered intellectualized or non-manual.

On one side, the work at the call centers is similar to taylorist typical production and uses its precepts, given the prescription's intensity, the cadence dictated by the "machine" (the calls' distributor), the *script* and the externally prescribed goals, imposed either by the call center and the external client. In addition, the ICTs enable a deepening of this productive logic as they make possible the real time monitoring; subsequent monitoring (recordings) and the accurate verification of working time, accounting for attendances and breaks. In that way, our conclusion is that work management at call centers is radicalized, to the extent that they reproduce and deepen the precepts of taylor-fordism, in addition to introducing the toyotism principles: working teams, collective goals (besides the individual ones) and the engagement of the worker, in his self-responsibilization for the work and supervision. On the other hand, regarding the software programmers' work, we believe that the management and control is also constituted of elements of partialization and tasks' distribution, and is also formed by self-management practices, through delegating productive responsibilities to the work collectives and calling for their adhesion, either for coercion and consensus, to the managerial techniques present in the agile methodologies.

Based on teamwork and anchored in the autonomy ideology, the work collectives establish goals, times and movements, closely monitored by the customers, through periodic meetings. These meetings have a double function: 1) the programmers establish time and goals to do each task, in the presence of all the members of the team and their supervisors (which tends to "encourage them" to estimate shorter times for the completion of each task); 2) the work management follows, in real time, the work processes, when workers, together, must report their performances in their tasks. The meetings are, however, coercion mechanisms, cloaked in autonomy. The engagement and consequent self-responsibilization of the programmers, therefore, composes a central strategy to control and to increase productivity.

This self-responsibilization of the work collective allows the company not to impose a despotic scenario as much as the one present in industries with predominance of manual labour and production of physical goods. The playful work environment - even if the "ideal type" of work environment in software programming exists more in some authors' imaginary about the industry than in the concrete reality of most of its workers - also does the function of organization and engagement of the programmers. These playful environments are only possible due the presence of self-management and self-taylorization as managerial patterns. Led by the technical leader, *scrum master* or *product owner,* working teams who manage themselves, the software programmation may acquire, therefore, autonomous and flexible appearance[22].

This process is anchored in a certain type of profile and conduct that the worker must follow, which is necessary to the *flexible* production based on projects. As reported by a *start- up* programmer we interviewed: "*(...) there is a Silicon Valley culture, it comes to us from movies, books, articles (...) that leads people to think this way. (...) They want to be the new Steve Jobs, the new Mark Zuckberg, the new Bill Gates"* (*Start-up* programmer, São Paulo, 30 years old).

We reinforce, although, that the labour force management is not *only* achieved for misleading elements of the work relation and supervision. We tried to point out the existence of a management that, in contact with the customers, elaborates the software to be developed and then transmits it to the programming teams. Besides, the constant monitoring of productivity of these workers and the tasks' execution is made possible by using softwares which elaborate indicators about the work every member of the team does. There is, also, as we have pointed out, face-to-face supervision through agile methodologies in daily meetings.

In short, there are at least two set of factors which organize the programmers' work: the first one is "external" to the working processes, but clearly affect them: competition determines the amount of time in which every phase of the software development must be done; the second one is internal to the working processes and is based in this "outside" pressure as it organizes all the work, time, tasks, goals and processes in the search of higher productive efficiency and accuracy, in order to reduce working time and to increase productivity. The prescription, therefore, is the basis to the definition of metrics and cadences.

---

22 In our interviews we found, for instance, workers who, in order to finish certain phases of the software programming, work overnight at the company. A "confraternization" atmosphere, with pizza and beer, that makes working time look little like working time. Kept the qualitative differences, we recall, however, how much the work organizations at the teleservice headquarters also use "ludic" aspects to increase productivity, either by the responsibilization of the teams over the results and goals, or by the productive competition among workers.

The agile methodologies in the organization of the software programmers working processes, therefore, are a form of advancement of the management of capital over labour. In this sense, the partialization of work, after the previous conception of the final product, point to a reproduction of the serial character of software production. Thus, although many times dressed by an apologist discourse of freedom, autonomy, creativity and collaboration, the fundamental structures of capitalist industrial production are reproduced.

Finally, we conclude that the prescription and standardization of work, which is observed in these two professional categories, even though they have different origins, wage averages and autonomy levels, reflects a tendency which might be generalized to a set of jobs that are mediated by the ICTs. Such tendency operates both among software programmers and call center workers, at the pace in which the processes of simplification and disqualification of work are reproduced.

As far as software programmers are concerned, the standardization of their work also gives rise to the possibility of precariousness through the "platformization" of work, since digital platforms in full rise make it possible to contract several tasks related to software development, without any formal link between contractor and worker being created, as well as without any rights arising from the employment relationship.

References

Amorim, H.; Grazia, M. (2021) "The precariousness of immaterial labour: self-taylorization in the Brazilian software industry", *Latin American Perspectives*, Thousand Oaks, v. 1, 1- 17.

Amorim, H; Moda, F.; Mevis, C. (2021). "Empreendedorismo: uma forma de americanismo contemporâneo?" *Caderno CRH*, 34, e021018.

Amorim, H.; Guilherme, G.; Moda, F.; Pelegrini, J. (2021) **"**O Empreendedorismo Contemporâneo ou uma Forma de Mistificação das Relações de Classe", *Contemporânea*, v. 11, n. 3.

Amorim, H. (2017) *Trabalhadores do Imaterial Precarizados.* In: Amorim; Souza (eds.). *As classes sociais no início do século XXI.* São Paulo, Annablume. p. 111-126.

Antunes, R. (1998) *Adeus ao trabalho?* Ensaio sobre as metamorfoses e a centralidade do mundo do trabalho. Campinas, SP: Cortez.

Bain, P., Watson, A., Mulvey, G., Taylor, P., & Gall, G. (2002). "Taylorism, targets and the pursuit of quantity and quality by call center management", *New Technology, Work and Employment*, *17*(3), 170–185. https://doi.org/10.1111/1468-005X.00103

Braga, R. (2013) A vingança de Braverman: o infotaylorismo como contratempo. in: *Riqueza e Miséria do trabalho II***.** São Paulo. Boitempo Editorial.

Braverman, H. (1974) *Labor and monopoly capital.* Monthly Review Press, 1974 (25th Anniversary edition, 1998).

Castells, M. (2010). *The information age:* Economy, society and culture, vol. 1, The rise of the network society, 2nd ed. with a new preface. Wiley-Blackwell.

Castillo, J. (2007). El trabajo fluido en la sociedad de la información: Organización y división del trabajo en las fábricas del software. *Revista de Trabajo*, *4*, 125–141.

Cusumano, M. (1989) "The software factory**:** a historical interpretation", *IEEE Software***.** Volume: 6 , Issue: 2.

DIEESE. (2017) Terceirização e precarização das condições de trabalho. Condições de trabalho e remuneração em atividades tipicamente terceirizadas e contratantes. Nota Técnica n. 172, março 2017.

Dutra, R. (2014) *Do outro lado da linha:* Poder Judiciário, regulação e adoecimento dos trabalhadores em call centers. São Paulo: LTR.

Gorz, A. (2010) *The Immaterial*: Knowledge, Value and Capital. Seagull Books.

Guilherme, G. (2020) *O trabalho dos programadores de software*: o desenvolvimento de uma nova gerência do trabalho? Master's thesis in Social Sciences - Escola de Filosofia, Letras e Ciências Humanas, Universidade Federal de São Paulo, Guarulhos. Available at: https://repositorio.unifesp.br/handle/11600/62130?show=full

Hodgson, D. and Briand, L. (2013) "Controlling the uncontrollable: "Agile" teams and illusions of autonomy in creative work", *Work, Employment and Society*, *27*(2),: 308–325. Available at: https://doi.org/10.1177/0950017012460315.

Kalff, Y. (2017) "The knowledge worker and the projectified self: domesticating and disciplining creativity", *Work Organisation, Labour & Globalisation,* v. 11, n. 1, p. 10.

Kunda, G (2006[1992]) *Engineering culture:* Control and commitment in a high-tech corporation Second edition Philadelphia: Temple University Press.

Lazzarato, M. (1996) Immaterial Labour. In: Hardt, M. and Virno, P., Eds., Radical Thought in Italy: A Potential Politics, University of Minnesota Press, Minneapolis and London, 133-147.

Marx, K. *Capital,* Volume I. Penguin Books, 1982.

Mason, P. (2015) *The end of capitalism has begun.* Available at: https://www.theguardian.com/books/2015/jul/17/postcapitalism-end-of-capitalism-begun.

Motim, B. and Rodrigues, M. (2018) Trabalho, autonomia e identidade entre os desenvolvedores de *software:* uma análise sobre a "natureza" e as condições de trabalho em TI. In: *Flexíveis, virtuais e precários? Os trabalhadores em tecnologias de informação.* Edited by M. A. BRIDI and J. LIMA. Curitiba: 143-172.

Miguez, P. (2018) O trabalho e as novas tecnologias: uma abordagem do trabalho informático segundo as teses do capitalismo cognitivo. In: *Flexíveis, virtuais e precários? Os trabalhadores em tecnologias de informação.* Edited by M. A. BRIDI and J. LIMA. Curitiba: 41-60.

Oliveira, M. (2017) Desenvolva a antifragilidade digital. In: *Lean way to digital success.* Campinas, CI&T.

Oliveira, M. *et al.* (2006) *Relatório da indústria de call-center no Brasil (2005).* The global call center industry project. São Paulo: ABT.

Sommerville, I (2011) *Engenharia de Software,* 9ª Edição. Pearson Education,

Schumacher, L. (2007) "Immaterial Fordism: the paradox of the industry", *Work Organisation, Labour & Globalisation.* Vol. 1, n. 1: 144-155.

Varoufakis, Y. (2017) *Capitalism is coming to an end, says one of the world's most respected economists.* Available at: https://www.independent.co.uk/news/world/europe/yannis-varoufakis-capitalism-ending-obsolete-former-greek-finance-minister-artificial-intelligence-a8006826.html.

Venco, S. (2006) "Centrais de Atendimento: a fábrica do século XIX nos serviços do século XXI", *Revista Brasileira de Saúde Ocupacional*. 31(114) 7-18.