Working under the clock
The pitfalls of having a fixed schedule at work

Roberto Enrique Chavez Rodriguez
Managing Software Development
Innopolis University
Innopolis, Russia
r.chavez@innopolis.ru

Abstract

The world is currently moving in such a direction where people, and particularly software engineers, can work for a company not only by going to the company's offices, but also remotely. Furthermore, established work shifts among these engineers seem to be antiquated for how projects evolution leads modern development of software products. In many cases, working a determined schedule may not indicate that software engineers work effectively. Rather, such a metric could hinder IT companies from analyzing real productivity of their computer science employees, and new performance and efficiency metrics must be determined.

***Index Terms*—engineering, performance, goal**

I. Introduction

Nowadays programming has become a skill that is not dependent in time, i.e., a software program could be written with quality in less or more time, depending on the ability of the programmers who develop it. However, if not managed properly with the time worked by the developer, this fact may trigger in losses for the company and a lack of efficiency in the development teams. It is common that companies fix a determined schedule on their employees, working in general 8 hours a day, but what if a programmer can finish their daily work in less time, what if in more? Having a fixed schedule may affect the performance of the specialist [1]. Another issue is how programmers look into the work of others. Sometimes they might complain about a colleague that receives the same salary but attends work only for a few hours, while they work under full schedule. They might think that this is unfair, and therefore they may also start working fewer hours or they might start to give some work time to personal things to do while still being at work, reducing their overall effectiveness. That is why that it must be determined if working under schedule or under skill brings the best benefits to a company that works with different computer scientist who have different levels in their abilities.

It is clear that in the field of Information Technologies and Computer Science, programming is a skill that is not dependent in time, i.e., a software program could be written with quality in less or more time, depending on the ability of the programmers who develop it. However, if not managed properly with the time worked by the developer, this fact may trigger in losses for the company and a lack of efficiency in the development teams. This paper aims at contributing to the analysis of whether working under schedule or under skill brings the best benefits to a company that works with different computer scientist who have different levels in their abilities.

II. Literature Review

A Bannai describes that eight hours of work in a day might be too much for an average worker [2]. In experiments carried out by Scandinavian Journal of Work show that working 8 hours consecutively for an extended period of time might incur in negative effects for health, particularly for the elderly. But working that many hours in a row does not bring negative consequences to old workers, but also to the young ones, where excessive work shifts trigger in depression and some psychological problems for these employees.

In fact, in several areas of the world different measures have been already taken to counteract the negative issues that come along with working the typical eight-hour work shift in the industry. Among these measures, there is the most known known one, the reduction of work hours, which was applied in many countries [3]. In the countries where such measures were implemented, research found that the level of performance did not vary that much from a scenario where people work 8 normal hours. As a matter of fact, even some places showed an improvement in worker performance, signifying that such changes can be good.

However, the performance improvements where not the same in all work areas. There were studies where countries that reduced the working time of their employees experienced negative economical outputs, since they compete internationally with other countries [4]. Nonetheless, it should be noticed that the industries where economical contrary effects where experienced were mostly physical-labor oriented, where workers had to perform physical work for and achieve objectives in a determined period of time.

A. *Orienting efforts towards the right goal*

Another factor to consider is the implementation of the right objectives for a company. Some companies might require workers to attend work for a concrete period of time, while other sectors in the industry might as well be interested in a more goal-oriented work time approach. The key issue is, in fact, to analyze what is the real objective for a company before planning the path to accomplish it [5]. Leslie Pack says that once a primary goal is established, companies should learn the right ways to attain the goal, without precisely caring in the amount time it should be taking to get the job done. And referring to computer science, goals are oriented towards delivering a good product for the client [6].

This software product comply with quality standars that are widely known in the field of computer science. Therefore, it could be inferred that developing a software product should be entirely focused on delivering it with these standards for the client to be satisfied with it.

Hence, a given worker could develop his or her work with decent quality in a given period of time. Now, considering the case where an experienced worker in software engineering can achieve his or her objectives in less time than his or her colleagues, there could be the chance that this professional have to help his or her colleagues. E. Erickson describes that the obligation of a worker to help with colleagues' tasks that were not initially assigned to them might bring frustration and burnout [7]. This worker would have to then do only his or her work and even if the worker finished a task before time, then they would not virtually have to do anything else, in order for the company to take care of its employee. Nevertheless, this is not the case in real life, and the question arises *"Does a company loses money when its employee does not work even though the employee achieved the goal?"*.

It could probably be that a given company does not have any issues with this scenario, but many other companies could have problems with their workers not collaborating with extra tasks that were not previously assigned to them. T. Gong describes in an interesting article how the hard work could be compensated by clients in a worker-client situation [8]. In the article, Gong explains that customers could probably reward the hard work of the employees by being loyal to the workers and always helping the workers by helping them improve their products with new ideas and customer feedback. Likewise, measures to support pro-activity and superior productivity could be taken by the companies towards their employees. In such case, working under a schedule could be very beneficial for an experienced software developer, because this developer could be working on extra tasks the remaining work time in a shift, becoming deserving of this developer's company's rewards and bonuses.

III. Ideas implementation and Discussion

In this section some ideas are discussed. These ideas could be implemented to measure effectiveness of software engineers at work and see if working under a given schedule brings more productivity or if working under objectives and skill is better.

A. *Company-goal-oriented plan*

It has been mentioned already that in the software engineering industry, a product with quality for the customer is the ultimate goal that must be achieved. The professionals involved in all the aspects concerning the product development and release could work under a *goal oriented* or *time oriented* metric. The following case is considered for the evaluation:

Let us consider the development department of the team developing a software product. Inside this development team, there exist an experienced software developer who finishes his or her daily tasks always in a lesser period of time than eight hours, say, five hours. What should he do the other three hours? The company has a few options:

- Option 1: Send this developer home since he or she finished his or her work.

- Option 2: Have the developer help his colleagues with other tasks, indicating that the developer's salary is given on the amount of time they work and not the amount of work they do.
- Option 3: Also have the developer help with other tasks but with the promise of a bonus.

It is clear that the company might experience some advantages or disadvantages. Analyzing each of the options, the company could experience:

a) Option 1: The developer is not working the three remaining hours of the day; therefore, the company would be wasting the developer's skills greatly, when other tasks could be assigned to the developer at that moment. On the other hand, the company would not be risking losing such an excellent employee by forcing them to do other tasks, in which situation, the developer may experience negative emotions towards the company. The company could just "wait" for the rest of the development team members to finish their tasks before moving on to the next phase of the project.

b) Option 2: On the one hand, in this situation the company would be maximizing the developer's skill power to carry out with tasks faster, having a more efficient development team. Consequently, the experienced developer might resent to be given so many extra tasks. This latter factor would cause of risk for the company to lose the employee. On the other hand, the company could remind the employee that the salary is being paid for the worked time and not the tasks done, but this could be a really risk approach, because it could lead to either of the following:

- the developer might ask for a salary increase: the company would lose income in the developer's salary rise.
- the developer might start to slow down their productivity. In this case the developer's skill power would not be seized to the fullest.
- the developer could think about quitting.

Therefore, it does not seem to be a good idea to have a salary conversation with the highly qualified developer.

c) Option 3: This scenario seems to be ideal for the developer, but the question is whether it could be ideal for the company. At the beginning it could be that giving bonuses to the employee for extra performance could trigger the fact that the company experiences loss income in the employee's bonuses, but a great advantage seems to rise in this scenario. The fact that an experienced developer takes gladly on extra tasks for extra bonuses (money or any other thing else) might speed up the development process of a software product. This factor could be greatly beneficial for the company, since it would be able to deliver products faster. Clients would be glad not to wait that much for their products, and everybody would win. However this could be just an ideal scenario, since in reality software development goes through a large phase processes that must be met before releasing a product to the customer [9].

It is in fact difficult to choose among all these options, because all advantages and disadvantages for the company should be carefully considered.

B. *Worker-goal-oriented plan*

Now again, the question arises: *"Should the employee work under time or under goal accomplishment?"* This question in fact is also difficult to answer and it should bring an employee to determine what his primary objective is, whether to work for a determined time or to achieve a determine career goal. It could be the case that the employee only wants to receive a salary and then work their assigned time in a day to later go home or do other activities. It could also be the case that this employee wants to achieve some milestones in life and moves towards them. This is why it so important to understand the employee's mind before trying to even answer the formulated question in this section. As already mentioned, it should be determined what is the real employee's objective to then aim at an strategy [10].

Let us consider the same case study where a experienced senior software developer is working in a project for a company. Likewise, the same example as in the previous section is cited, where the developer reaches their eight-hour work goal in only five hours. What should this skilled professional do in the remaining three hours? A few of the possible options the developer could take are:

- Option 1: Go talk to the human resources department to see if he can go home early.
- Option 2: Stay in the office and hang around while the day work shift ends.
- Option 3: Willingly take on other tasks and possibly wait for the company's initiative to offer them a better position or a bonus.
- Option 4: actively state the developer wants to get some extra bonus in exchange for doing more work they are supposed to do in the company.

Again, as formerly analyzed with the company, the worker might experience some positive and negative outcomes from making any of these decisions. Let us assess each of these options to see possible consequences for the developer.

a) Option 1: The developer might actually create a bad self impression on the company. The company could think the developer is not committed enough to the success and progress of the company. This could potentially generate bad popularity for the developer. Now it could also be that the company does not mind the developer starting to go home at early hours, taking into account the good performance the developer has, but experience and studies indicate that companies do not have much of a sense of care for the worker but they rather focus on performance and productivity [11]. Therefore, this option might as well not be the best the developer could make. Let us explore the other options.

b) Option 2: Here the developer would experience one or more of some of these interesting consequences:

- If the developer only waits until the completion of their time, both they and the company would lose productivity.
- The company would soon realize the developer's slack-off in working hours. Therefore, the developer could start being badly seen by their the company. This would be a great minus.
- It could be that the developer finds this time valuable to review their work and do other tasks to support and check their previous work. This attitude would be bene-

ficial for the company and the employee. However, it is not clear if the developer would get something beneficial from such an action, since the company might not notice these extra in-office efforts.

c) Option 3: In this case the developer would be extremely kind, and this is rarely observed in research. Such mindset could exist in some workers but tt is well-known that humans act mostly on rewards of any kind and not only because they want to do extra work without any incentives [12]. However, if the company were willing to offer some kind of extra bonus for the developer, then this decision would seem quite feasible to be made. The developer and the company should come to terms to clearly establish the conditions under which a bonus could, if ever, be given for exceptional or extra work.

d) Option 4: In this context, the developer could be seen either as pretentious or as an innovator for productivity boost. In the first scenario, making such a move would reduce future possibilities for the developer to level up their career in the company. But in the second situation, the developer might arrange some good deals for rewards to extra productivity that would positively affect them and their colleagues. The relationship between the company and the employee should be analyzed in order to make a decision concerning this option. In many cases, where the worker has already been part of a company for a long time, they have importance in the company and the employee's opinion is considered

Although it is hard to make a choice and fully predict the consequences of each of them, it should be noticed that, up to this point, it has been clearly showed that, as everything in life, working under the clock or under goals would bring benefits and drawbacks for both, the employee and the company.

IV. Conclusion

To this point, several scenarios have been brought up for both, the worker and the company and possible consequences of these scenarios were also described.

It is worthwhile mentioning that the analysis was done in a hypothetical environment where the developer in question was an experienced worker who could finalize their work in less time than the assigned one. *Why was not the analysis done in the case of a developer who needs more time than the assigned to complete their work?* The answer seems to be obvious here: because simply, in such case, the company would eventually replace this worker for a more effective one. In fact, the analysis of this paper was done in a company-worker relationship and not in the software development itself, an area that requires other type of analysis briefly described in this document, when observing that software products must be done with quality first over doing them fast.

All in all, working under a fixed schedule seems to have issues, either for the developer or for the company, and it would be much better for the companies in the Information Technologies world to implement a system that encourages their workers to accomplish several tasks instead of having hem sitting idle just waiting for the clock to mark their time. However, this recommendation is not set in stone, and the analysis of this paper is only to bring a guideline for those software managers looking to understand the problems of having their employees working under the clock and the things they could do about them.

References

[1] Irene Hau-siu Chow and Irene Keng-Howe Chew. The effect of alternative work schedules on employee performance. *International Journal of Employment Studies*, 14(1):105–130, 2006.

[2] Akira Bannai and Akiko Tamakoshi. The association between long working hours and health: a systematic review of epidemiological evidence. *Scandinavian journal of work, environment & health*, pages 5–18, 2014.

[3] Torbjörn ÅKERSTEDT and Göran Kecklund. The future of work hours—the european view. *Industrial health*, 43(1):80–84, 2005.

[4] E Jeffrey Hill, Jenet Jacob Erickson, Erin K Holmes, and Maria Ferris. Workplace flexibility, work hours, and work-life conflict: finding an extra day or two. *Journal of Family Psychology*, 24(3):349, 2010.

[5] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer, 1993.

[6] John L Bennett. Observations on meeting usability goals for software products. *Behaviour & Information Technology*, 5(2):183–193, 1986.

[7] Eva Ericson-Lidman and Gunilla Strandberg. Burnout: co-workers' perceptions of signs preceding workmates' burnout. *Journal of Advanced Nursing*, 60(2):199–208, 2007.

[8] Youjae Yi and Taeshik Gong. If employees "go the extra mile," do customers reciprocate with similar behavior? *Psychology & Marketing*, 25(10):961–986, 2008.

[9] Leon Osterweil. Software processes are software too. In *Engineering of Software*, pages 323–344. Springer, 2011.

[10] Athanasios Hadjimanolis, Georgios Boustras, Aristodemos Economides, Anastasios Yiannaki, and Leandros Nicolaides. Work attitudes and safety performance in micro-firms–results from a nationwide survey:(the opinion of the employees). *Safety science*, 80:135–143, 2015.

[11] Pina Tarricone and Joe Luca. Employees, teamwork and social interdependence–a formula for successful business? *Team Performance Management: An International Journal*, 2002.

[12] Tomas Jungert, Kaspar Schattke, Félix Alexandre Proulx, Geneviève Taylor, and Richard Koestner. Whose autonomy support is more effective? managers' or co-workers'? an experimental comparison of source and occupational context on intrinsic motivation. *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, 38(2):209–223, 2021.