**Business Integrated Development**

Joshua Arbon

Neumont College of Computer Science

ENG 110: Intro to English Composition

Sarah Carter

November 21, 2024

Business Integrated Development

There are many aspects in making a successful business, I would like to explore one of these aspects in depth. Throughout this paper, I will discuss the software development perspective of business management. More specifically, I will be talking about how micro-management is detrimental to the workflow of software developers. I will first talk about businesses that do not rely heavily on management within their company. Do developers in these companies perform better or worse? Then I would like to talk about the necessity of management in a successful business regardless of software developers. Does the necessity of management negatively affect developers? Next, I will talk about the necessity of separating management and developers. Finally, I will speak on actual methods implemented in industry to bridge that gap between development and business management. Without further ado I would like to answer; Does pressure from business-minded individuals negatively affect the productivity of software developers?

First and foremost, I would like to talk about a few companies I have found in my research that do not use traditional management systems. The first of which I have found is Valve Corporation or Valve for short. Everyone in the tech industry knows Valve. Valve and its CEO, Gabe Newell, are well known for providing amazing products such as Steam that are revolutionary in the tech field for their incredible consumer-first business practices. Since Valve’s founding in 1996, they have set the standard for online video game marketing services. It should also be noted that according to glassdoor.com, 90% of Valve employees would recommend working there to a friend. Not only that but a whopping 97% of employees approve of Game Newell, the CEO of Valve.

But what about Valve is so interesting regarding its management? Simply put, since the founding of the company Valve has been manager-free (Stewart, 2019). This is insane considering Valve in 2022 was worth $7.7 billion, and they generated around $13 billion of revenue in the same year (David, 2024). This goes to show how businesses can be successful without these deep management systems that are found in nearly every company. It is good to see a successful company that does not have a management structure that also treats its employees well. However, this sort of success is not common for a company like this. Valve is a statistical outlier in their managerless approach to development.

Other companies such as GitHub have tried this style before. In GitHub, this style of management was used up until 2014 when GitHub decided to introduce more formal styles of management into their development process (holman, 2016). One potential reason for this could be that its co-founder Tom Preston-Werner resigned from the company in 2014 (Miller, 2014). This is however only speculation as to why they switched management systems. But the fact still remains that to this day GitHub is a successful company whose product is used by developers worldwide.

As mentioned before, Valve is an outlier in its managerless business practices. Other companies have managers for their employees. Not only do some of these companies have managers for employees, but they also have managers for their managers, and managers for those managers. The thought of having managers for managers is insane to me, but it’s quite common in the business world. It is so common in fact that they have different terms for these varying levels of management. They are as follows: top management, middle management, and lower management. Each level of management reports to the level above them, with the top-level management reporting to themselves as the CEO, the president, and the board of directors (Hanif, 2023).

In practice, the typical management chain works, but this is assuming no one goes on a power trip with their role in the company. On occasion, you will find managers who abuse their power and do something called micromanagement. Micromanagement is when a manager tries to control everything about their team, a situation, a project, or a place (dev, 2023). Employees hate micromanagement, especially software developers. Software development is already stressful enough with deadlines to meet. Adding a manager breathing down your neck and watching your every move to the mix only adds to that pile of stress. There are more negative effects of micromanagement, not just stress, all of these are direct effects of micromanagement: poor job performance, decreased morale, high turnover, decreased teamwork, less innovation, faster burnout, and as we already discussed, more stress (dev, 2023).

Regardless of the chance of micromanagement, management is still necessary in most businesses to run the company. It should go without saying but, most software developers are not businesspeople so to speak. They are developers and they should focus on developing software. Businesspeople are needed to run the company, but software developers do not want management breathing down their necks. So now there is a problem. How do you keep pesky micromanagers away from developers so they can continue to work efficiently? As it turns out, others have had this question and have come up with a solution. One proposed solution is to introduce a middleman.

Better said, there needs to be a role in the software development process that communicates between managers and developers and vice versa. This role exists in modern-day development environments, and we call people in this role, Scrum masters. Scrum is a development process implemented in most software teams these days, and the Scrum master is just one role in that process. So, let’s take a step back for a minute and explore what Scrum is and why it is used so widely in development environments.

Scrum is an implementation of agile development, which is the counterpart to waterfall development. AWS puts it like this,

Scrum is a management framework that teams use to self-organize and work towards a common goal. It describes a set of meetings, tools, and roles for efficient project delivery. Much like a sports team practicing for a big match, Scrum practices allow teams to self-manage, learn from experience, and adapt to change (AWS, n.d.).

To sum that up, Scrum is a framework where teams self-manage and adapt to change through a variety of roles. There are a multitude of roles in Scrum which are: Scrum Master, Product Owner, and Development Team. Of these roles, the two important ones are the Scrum Master and the Product Owner. Both help bridge the gap between management and developers.

Put shortly, the Product Owner understands the customer and business requirements (West, n.d.). They are also responsible for the product backlog, which is a list of requirements or features for the product. The Scrum Master ensures that Scrum practices are being done regularly and correctly. They also help the Product Owner maintain the product backlog (West, n.d.). Aside from that, probably the most important thing they do is help resolve blockers that are outside of the development team’s responsibility. This is what makes them so valuable in bridging the gap between management and developers. Without a Scrum Master, those unnecessary tasks would be given to the development team. This would result in the product being delayed, and likely the developer being fired. However, the Scrum Master helps ensure the productivity of the development team by assisting management with assigning those tasks to the correct individuals.

The value in Scrum should be readily apparent given this information, but I would like to address one potential downside to this whole process. This downside is the possibility of miscommunication through the Product Owner or Scrum Master. One of the purposes of Scrum is to remove developers from the management process, but this leaves room for error in communicating between the roles put in place to achieve that. Information that needs to be conveyed to management from the developers now must be conveyed through a middleman that would not exist in a non-Scrum environment. This, however, is non-consequential because Scrum is a non-rigid process, and any mistakes can be corrected the following morning in the daily standup meeting. This is a meeting where the Product Owner, Scrum Master, and Development Team talk about all the things they have done the previous day, as well as their plans for the current day. This is also the time when the Scrum Master or Product Owner would ask clarifying questions about an individual’s progression.

So, in conclusion. It has been shown that businesses can be successful without any management systems in place. They are, however, very rare, and even more rarely are those businesses successful. More commonly there are management systems put in place to run the business side of things. But these management systems can on occasion be detrimental to the workflow of employees in cases where a manager ends up micromanaging. This is solved by adding a middleman who does all the communication between employees and managers. This does have a downside of potential miscommunication which is easily addressed in the Scrum development process. All these points lead me to believe that micromanagement negatively impacts developer workflows.

**References**

7pace. (2021, May 26).  *7 Negative Effects of Micromanagement on Software Teams.* Dev. <https://dev.to/7pace/7-negative-effects-of-micromanagement-on-software-teams-2chj>

Comparably. (n.d.). <https://www.comparably.com/companies/valve/reviews>

David, C. (2024). *Valve Revenue and Growth Statistics (2024).* SignHouse. <https://usesignhouse.com/blog/valve-stats/#:~:text=Valve%20is%20worth%20%247.7%20billion,was%20from%20the%20Steam%20store>

mercmobily. (2016, October 9). *How did managers get introduced in 2014, and did it really change everything?* GitHub. <https://github.com/holman/ama/issues/800>

Miller, C. (2014, April 21). *GitHub Founder Resigns After Investigation.* Bits. <https://archive.nytimes.com/bits.blogs.nytimes.com/2014/04/21/github-founder-resigns-after-investigation/?_php=true&_type=blogs&hp&_r=0>

Stewart, h. (2019, August 19). *Thirteen companies that don’t have managers.* LinkedIn. <https://www.linkedin.com/pulse/thirteen-companies-dont-have-managers-henry-stewart/>

Sivasubramaniam, J. (n.d.). *Five Companies with Almost No Hierarchies.* Berret-Koehler Publishers. <https://www.bkconnection.com/blog/posts/five-companies-with-almost-no-hierarchies>

West, D. (n.d.).  *Agile scrum roles and responsibilities.* Atlassian. <https://www.atlassian.com/agile/scrum/roles>

*Checked for grammar by Grammarly (2024)*