Jonathan Lam
Prof. Barrett
EID370
Engineering Management
5 / 10 / 22

Lessons on engineering management and applications to software engineering

At some point, I looked down on managers and people studying management. To me, managing is purely a soft skill that could be easily picked up, and I thought that only the brilliant mathematicians and scientists and engineers who were working at the pinnacle of their respective fields were the smart ones. Through this class, though, I learned that management is a skill that can be taught and learned, and is not intuitive or trivial at all.

It seems to me that engineering management is a fairly intuitive process, but that it may not be easy to tell from a glance what methods work from what methods don't work. As I had thought, management is a role for people with elevated soft skills, but this is far from easy due to the inconstant and sensitive balance of every individual. It also seems that many other engineers have fallen into a similar trap of looking down on management: we learned that many engineers (or people in other technical roles) who have risen to the role of engineering management initially end up stronghanding their position (e.g., via micromanagement), since they have a preconception that management is solely about leading and owning the group of subordinates.

No, we learned that management is not about leading or owning or controlling people; management is about *organizing a group of people towards achieving the company goals*. Anything else is secondary and an intermediate goal towards the company goals. One may consider additional characteristics such as respecting employees or promoting efficiency as part of the definition for management, but really these are still subordinate to the more general definition provided above.

I found the evolution of engineering management thought to be extremely formative. There was not really a strong need for management before the Industrial Revolution, because there are not many large organizations of people working together for extended periods of time, except perhaps outside of the military or government. Thus military management was some of the oldest forms of management. We did not study ancient military management because it may be too different and outdated for modern engineering management, even if there are probably many overlapping traits.

In the Industrial age, we see a major factors contributing to the need for management. Firstly, mass production and efficiency were the new forces behind innovation and human progress. Secondly, people were moving away from traditional tradesman jobs, which work at a much smaller scale than industrial factories and with much fewer people in tandem. Lastly, there is a change in the nature of many of the jobs, especially with the use of machines – many jobs have become less "human" and more repetitive. The question of how to keep these large groups of people working together efficiently and effectively, while not becoming demotivated, becomes a very important logistical problem.

Enter some of our classical schools of thought. We have Frederick Taylor, who was one of the pioneers in scientific management in the late 19th century. Taylor's methods were solely aimed at improving efficiency in the steel-making processes at his firms. He proposed many logical improvements to the current processes, such as incentivizing above-standard work with extra pay, training new workers, rest breaks, and division of work between workers and management. While these ideas may be common sense for us nowadays and more or less still valid today, they were not in widespread practice before then. The Gilbreths also sought to improve efficiency through the use of motion studies. These methods, which systematically assess actions to improve efficiency to the utmost, are similar to data-driven methods today.

Taylor, however, was often hated by his workers because he made them perform more work. Around the same time, the idea of behavioral management (or "human relations management") arose, most notably with the Hawthorne Experiments by Elton Mayo at the Western Electric Company. The experiments arose out of a desire to improve efficiency, but their findings were unexpected. The first experiment, with the six workers and changes to the work environment, found that workers tend to do better if consulted by management about changes to their environment, irrelevant of the actual change.

The second (and less famous) experiment showed that certain changes to improve efficiency may always be rejected by an existing group, and thus that existing group norms must be taken into consideration when implementing any change. Together, these experiments show the importance of understanding that the needs of the group and the individual need to be considered, and that the human psyche is not straightforward, predictable, or intuitive. While Taylor thought that everything could be solved with plain “common sense,” Mayo showed that this is not the case. Undoubtedly, the Hawthorne effect is widely accepted today, and feedback for management changes are common implemented today.

At some point, with the introduction of many different management styles, some of the ideas began to consolidate into a set of general principles. Henry Fayol’s Fourteen Principles are probably the most eminent. These fourteen rules give a general framework for thinking about organizing a large group of people into a hierarchical structure. Briefly summarized, these rules are:

1. *Division of work*: Workers should specialize to a small number of tasks.
2. *Authority and responsibility*: Authority and responsibility should be appropriately delegated so that people can carry out their tasks effectively.
3. *Discipline*: Appropriate levels of discipline are necessary to ensure order in the workplace.
4. *Unity of command*: Each worker should have a single manager.
5. *Unity of direction*: There should be a single overall plan that everyone is working towards.
6. *Subordination of individual interests to the general interests*: The company’s goals are foremost, even before employees’ own goals.
7. *Remuneration*: Employees should be appropriately paid.
8. *Centralization*: An appropriate level of (de)centralization dependent on the circumstances may increase overall organizational efficiency.
9. *Scalar chain*: Chains of authority should be vertical and not horizontal, to avoid confusion.
10. *Order*: Order should be obtained by establishing well-defined roles for everyone.

11. *Equity*: Each employee should be treated equally and fairly.
12. *Stability of tenure of personnel*: This may help improve performance and lower costs and time for training employees.
13. *Initiative*: Employees should be encouraged to make plans for themselves.
14. *Esprit de corps*: The workplace should be a friendly place that encourages working together.

To me, these set of rules again feel very intuitive, but that may only be because these ideas are so deeply intertwined with modern corporate culture.

With the rise of a new type of machines (the digital computer), there are inevitably another wave of huge changes to the nature of work. Computers became deeply intertwined with work with a new "data-driven management," in which the scientific simulations or machine-learning predictions generated by computers are incorporated into the decision-making process. Data-driven methods are the driving force of many successful technology companies today, so the application of data-driven computer methods to management is no surprise.

As the number of management styles grew, people inevitably came to realize that the circumstances dictate the style of management. This itself may be called another style of management, called the "contingency approach." This is represented by the black-box in Barrett's model, and shows that we must be agile to the scenario. On a similar vein, we also realize that all styles of management require an understanding of the entire organization as a complex system of many entities and interactions, and thus we may picture management abstractly using the "systems approach."

In sum, the whole of management history comes down to "it depends." This shows how difficult management is, and how far off I originally was when looking down on management. It seems that we can only build up a repertoire of general management ideas, and apply the most relevant ones.

In my field of work, which is software engineering, the most common form of management is a fairly standard functional organization. There are "levels" of managers (called simply "Level 1," "Level 2," and on, which higher levels corresponding to higher levels of management). Perhaps the most widespread management theory in software engineering is something called the Agile methodology, which encourages rapid feedback through iterative development[1]. The smallest unit of management in software engineering is usually called the Scrum team, where Scrum is a specific set of actions that coexists with the Agile methodology. We also clearly see many of the other methodologies that were studied throughout the book, from Taylor's idea of training workers (which corresponds to an "onboarding" or "bootcamp" process), to Gilbreth's data-driven efficiency, to Mayo's feedback-driven management, to Fayol's "esprit de corps" with comfy workplaces, to the Japanese management's Kanban method of just-in-time development (Kanban boards are common in software development, e.g., through the Jira system).

I would best describe the software engineering approach as being a highly data-driven contingency approach. As the name suggests, the Agile technology, with its short development cycles and rapid feedback, allows for software companies to be extremely fast-moving and worker-friendly. Since it is data-driven, the things that work best (through past historical data) may be used to influence future decisions, even if they are perhaps not conventional wisdom, and rules can be broken if it improves efficiency and does not hurt overall management. For example, I believe that there is a lot of horizontal communication between employees for various reasons, which encourages overall knowledge share and camaraderie; this is in contrast to scalar chain, which specifies a more rigid structure in which only vertical communication is recommended.

The debates over management styles are becoming ever more relevant nowadays as the COVID-19 pandemic and work-at-home policies prompt another great impetus for changing work and management styles[2]. There was the so-called "Great Resignation" movement during the pandemic due to loss of motivation and disillusionment with work. While we must put some blame on the pandemic

[1] Honestly, I'm surprised this didn't come up in class, given how pervasive the idea is in software engineering. Perhaps it is because it is more recent than many of the topics in this class – many of the recommended books on software giants such as Apple, Google, and Amazon date back two decades, but the Agile methodology has only really become dominant in the last decade or so. It would be good to have some modern books on management in software engineering for this course.

[2] I believe that management since COVID-19 would be another great book topic for future classes.

and the inevitable stay-at-home and remote work policies, this is also an important challenge for managers to be able to adapt to such situations, continuing the respect the worker, and finding ways to motivate them in the face of greater troubles than ever before.

Luckily, software development companies have weathered the Great Resignation fairly well, due to a number of factors (e.g., software booming during stay-at-home policies, remote work usually works fine for software, Agile methodologies quickly adapted to new policies, etc.) but many other industries (e.g., restaurant workers or medical workers) didn’t fare as well. Being able to apply the lessons learned from (engineering) management will be important for any corporation in this volatile time.