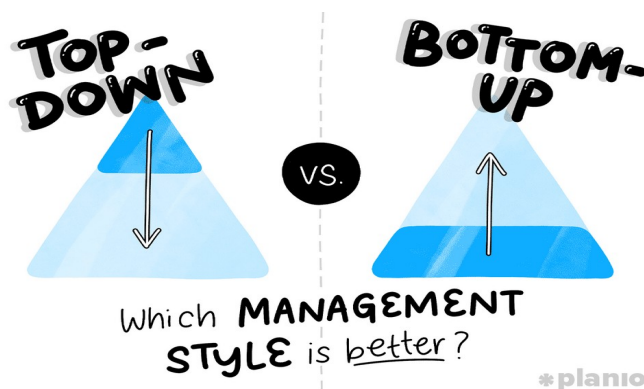


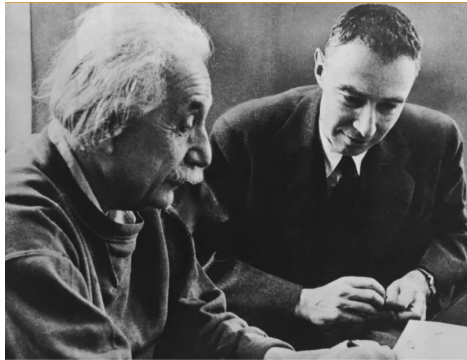
### 5.3 Hierarchical versus Horizontal Management



The most successful software development teams tend to be horizontally organized, with “bottom-up,” consensus-driven decision making. The US is generally thought of as a less hierarchical culture, but the software industry is *extremely* resistant to hierarchy, even by US standards. This is by design, and an outcome of how the industry was created. Building on this early history, a more bottom-up, horizontal organizational structure has been **found to produce better results**, so this tendency has only grown more pronounced over time.

In a PSHS context, this is relevant, even though some hierarchy is inevitable. Wherever possible, within the team a more horizontal organization should be encouraged. For example, I recommend avoiding any reporting relationships between the different team members; any determination of bonuses, salary or other typical management functions should be kept outside the development team. A developer should be empowered to challenge the project manager, a QA engineer should be encouraged to express opinions about the UI and about testability of the system, and so on.

It is perhaps a worthwhile digression to consider the history of software, as a means of understanding why the industry is so anti-hierarchy. I contend that this goes back to The Manhattan Project. The US Army did something absolutely brilliant: They recognized that a military organization steeped in hierarchy was incapable of the required innovation. Instead, they recruited Oppenheimer, who built a non-hierarchical oasis in the New Mexico desert. Interactions with their customer (the military) were managed by Oppenheimer, and he sheltered his team members, to the extent possible, from the hierarchical mandates that can kill creativity and innovation.



In the immediate aftermath of World War 2, the US military again showed brilliance and humility. They realized the strategic importance of computer technology (learning the lessons from e.g. the UK's Bletchley Park<sup>15</sup> codebreaking of the German Enigma encryption machine). They also recognized, again, that the military was incapable of the kind of technological innovation required. So, starting in the late 1940's, the US Department of Defense gave lavish funding to civilian universities, notably MIT and Stanford. They also funded the founding and operation of a civilian research and development company, called the RAND Corporation (Research ANd Development). RAND had the freedom to pursue wild ideas, even including the theoretical "world-circling spaceship" that was featured on the cover of their 1949 report.



15. <https://bletchleypark.org.uk/>

In 1955, RAND spun off the System Development Corporation, SDC. SDC is considered to be the first computer software company. Incidentally, I have a personal connection to SDC and RAND.<sup>16</sup>

Many of the people working at SDC were veterans of World War 2, and they did not universally hold military officers in particularly high regard.<sup>17</sup> This probably contributed to the anti-hierarchical bias in the software industry. Internally, SDC went to great lengths to keep the programmers away from their primary customers, the US Air Force and DARPA (the Defense Advanced Research Projects Agency). Another sign of the non-hierarchical nature of SDC as early as the 1950s was the dress code. At the time, men were generally expected to wear a suit and tie in an office environment. Particularly during hot summer days in the Los Angeles area, SDC programmers simply refused. There wasn't much management could do about that, since experienced programmers were extremely rare in an era where a computer cost millions of dollars – a very large sum indeed in 1960!

This culture of empowered computer programmers who are resistant to hierarchy (and who dress informally) has continued and grown with the rise of the software industry. Partly, it's because a more horizontal decision-making structure is so much more *effective*. Perhaps this is because software systems are so complex, and the only people who truly grasp the real complexity of a given system are those who work at a detailed level with the code, day-in and day-out.

Incidentally, DARPA also created the ARPAnet,<sup>18</sup> which grew into today's internet. Much of the design of the ARPAnet was motivated by networking challenges that SDC encountered developing the US defense against Soviet long-range bombers, SAGE.<sup>19</sup>

The preference for non-hierarchical, horizontal decision-making has made its way into formal software engineering methodologies. From Fred Brooks' observations in the seventies, through the object-oriented revolution of the eighties and nineties, and into Agile and related methodologies of the early aughts, there has been ever-increasing support for a "bottom-up," consensus-driven approach. The Project Manager isn't really "in charge," he or she acts more as a facilitator, and as a screen to protect the team's integrity and focus.

## 5.4 Other Documents

During my volunteer service at PSHS, I produced a number of other documents. They are in a companion file called `other_documents.zip`. They are:

- **final\_report\_preliminary\_draft.pdf**: An old draft of this document, included for completeness. It has some analysis of other PSHS systems.
- **campus\_visit\_cvc.pdf**: A report of my campus visit to CVC.

---

16. <https://jovial.com/>

17. I believe my father, a WW II veteran who (like my mother) was a programmer at SDC, was known to refer to them as "those stupid sons of bitches wearing uniforms."

18. <https://en.wikipedia.org/wiki/ARPANET>

19. [https://en.wikipedia.org/wiki/Semi-Automatic\\_Ground\\_Environment](https://en.wikipedia.org/wiki/Semi-Automatic_Ground_Environment),  
<https://www.youtube.com/watch?v=iCCL4INQcFo>