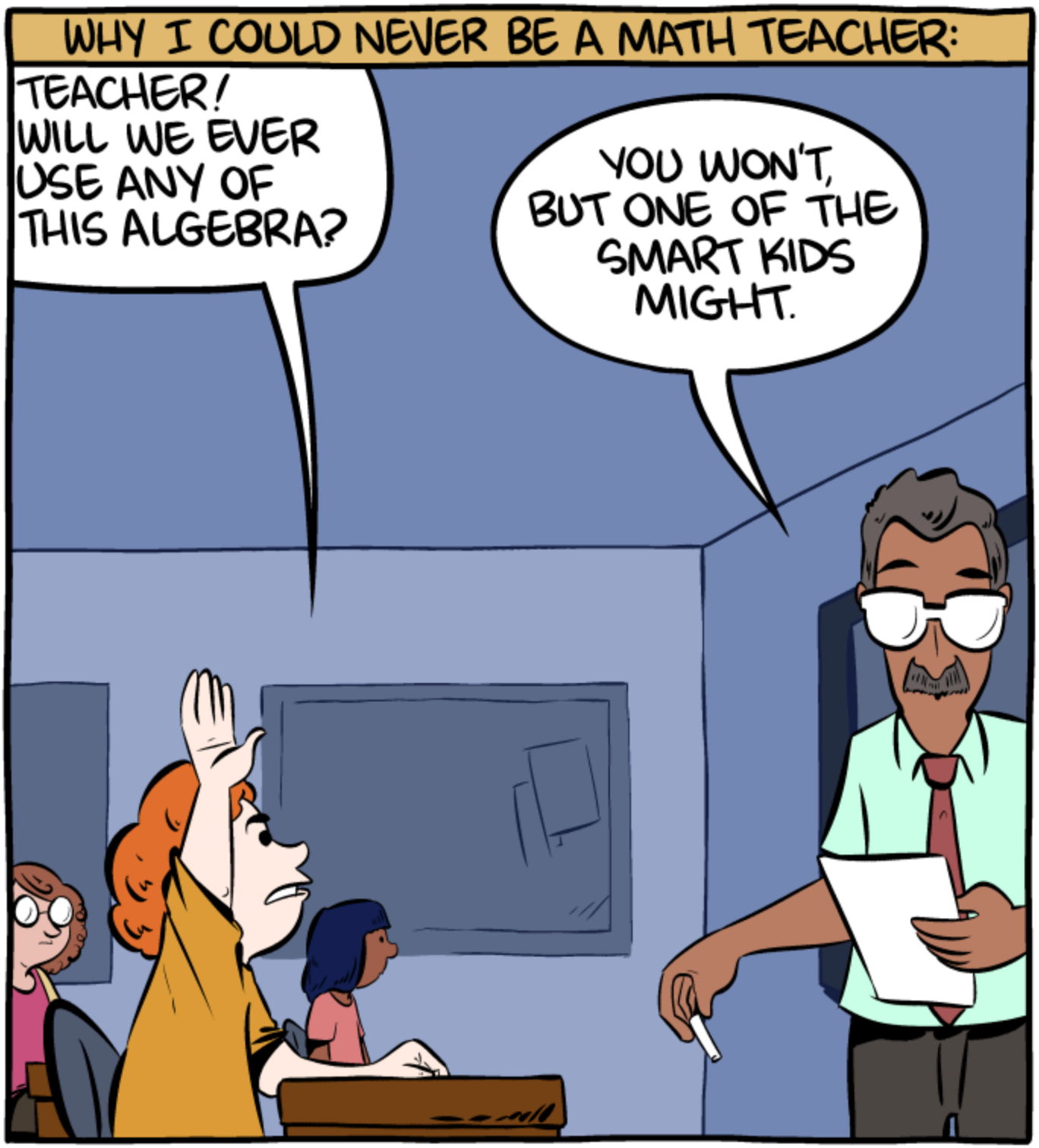
Optimize For Exceptionally Competent Small Teams

Oct 13, 2025 - 2182 words

Christopher Kalitin Blog (https://ckalitin.github.io/posts/managing-technical-teams/)



Several years ago I saw the meme above and realized I came to the exact opposite conclusion as most people discussing it. I've done my share of complaining about education (https://ckalitin.github.io/idea/2025/04/26/low-leverage-university.html), but the most useful insights from this meme is not to feel sympathy for the child.

The very small number of "smart kids" in that classroom have the potential to go on to do extremely useful things for humanity that will improve all of our lives and allow for prosperity long into the future. Catering to such students that show promise at doing useful things is far more important than catering to the median student. Having been one of them, I feel far more sympathy for the smart kids being forced to sit in that slow, boring classroom than the students slowing them down.

Catering to the smart kids does not end with education, but it extends to using their abilities to maximize leverage in humanity's most difficult technical projects.

I've spent the past couple of months reading books on some of the early monumental technical projects in human history. There are many common themes among them, especially in how the teams of exceptionally competent engineers were managed. Many of my insights are distilled from these books, hence this post will be heavy on quotes.

Small Strong Teams

"If the team is small and strong, then engineers and the code are the source of truth. So, they [engineers] have the source of truth, not some manager."
- Andrej Karpathy (https://x.com/rohanpaul_ai/status/1977266545333297525)

"The Skunk Works manager must be delegated practically complete control of his program in all aspects."
- Kelly Johnson's Skunk Works Operating Rule #1

Most importantly, a small number of exceptionally competent engineers can extremely rapidly make decisions that advance progress to their shared goal. If all engineers have ownership over their individual systems, they have the authority to make decisions they find useful quickly. This decreases the timeline to completion of a project. However, this only works if the decisions made rapidly are the right decisions, hence the prefix "exceptionally competent."

"There must be a minimum number of reports required, but important work must be recorded thoroughly."
- Kelly Johnson's Skunk Works Operating Rule #5

Furthermore, a small team means all engineers can easily interface with each other without excessive documentation or meetings. All engineers must understand how changing their system affects other systems and must understand all relevant interfaces. A subset of being an exceptionally competent engineer is being interdisciplinary, and this helps when interfacing with other engineers and systems.

“The number of people having any connection to the project must be restricted in an almost vicious manner. Use a small number of good people.”
- Kelly Johnson’s Skunk Works Operating Rule #3

The marginal addition of an engineer to a project makes interfacing more difficult, as there is another person making decisions to interface with. A greater number of engineers increases overhead, which is the fundamental reason why large teams move so slowly. Rapid decision making also becomes harder as more people must be consulted / interfaced with.

When engineers are let loose in a competent organization with a clear goal, they can independently attack critical path problems and find new solutions and innovations to problems that other people may not have even noticed. Instant decision making and minimal interfaces allow for individual engineers to move quickly to attack such problems. If this is not a clear point to you, read the examples below.

Examples

“In the seconds-critical world of Mission Control, a single individual must assume responsibility to take any actions needed for crew safety and mission success.”
- Failure is Not an Option, Chapter 22, Gene Kranz

In some cases, success is completely impossible with a large team. Apollo Mission Control is an example of this. During the Apollo 12 mission a lightning strike on ascent caused a fuel cell failure in the Command Module. If power was lost for >120 seconds, an abort would be required. John Arron was on console monitoring cabin pressure, saw scrambled data, and within seconds made a connection between this faulty data and similar data seen in ground testing a year prior when a technician inadvertently turned off a power supply. He traced the issue to the Signal Conditioning Equipment (SCE) power supply, and the crew switched to the Auxiliary (AUX) power supply, saving the mission.

If one exceptionally competent engineer was not on console, the Apollo 12 mission would have been a failure. Note that the term [“steely-eyed missile man”](https://en.m.wikipedia.org/wiki/John_Aaron) originated from John Aaron’s performance during this incident.

The reason Skunk Works was able to develop stealth technology was because a mathematician and radar specialist, Denys Overholser, came across a translated Russian paper by Pyotr Ufimtsev that used Maxwell's and Sommerfeld's work to predict what geometries would reflect electromagnetic radiation and how to calculate radar cross section. This work was mostly overlooked in the Soviet Union, but Overholser created computational methods for calculating radar cross section and convinced Ben R. Rich, who ran Skunk Works at the time, to pursue it. This spawned the F-117 Nighthawk program and improved US stealth capability by orders of magnitude. Overholser independently found a solution to a critical path problem for the US military.

To use extremely competent engineers to maximum leverage you must let them make decisions quickly, or get their managers to make decisions quickly. This excerpt (https://x.com/yrouel86/status/1967887843113799969) from Christian Davenport's new book Rocket Dreams tells the story of how two engineers designed a simplified docking collar for SpaceX's Dragon capsule, showed it to Mark Juncosa -"one of Musk's most trusted engineers" - who showed it to Musk, who within minutes approved the design.

"There were no deliberations. Not consultations with other engineers. No memos or meetings. Musk liked what he saw and simply made the decision to go." This example should illustrate why having a good technical intuition is extremely important. If Elon made the wrong decision quickly, it would have been suboptimal, but he made the right decision. Extreme competence is just as important as extreme speed.

What Must You Do In The Absence Of A Small Strong Team?

"Our people knew what they were doing, worked skilfully under intense pressure, and skirted hazards mostly by sheer expertise and experience. But as we grew the skill level decreased and sloppiness suddenly became a serious problem."
- Skunk Works, Chapter 3, Ben R. Rich

The UBC Solar student design team I'm a part of is a fun, fairly benign example of a slightly-too-large team with varying levels of competence.

Because we constantly take in new first-year engineering students who don't particularly understand how to do useful engineering work in a technical organization, we cannot run like the small strong teams I have been describing. Instead, more systems and processes are required to ensure everyone is making progress towards our common goal of winning competition.

My first project on UBC Solar was characterizing our main pack current sensor. This took about 5 months from start to finish. For context, when our previous electrical lead Mischa Johal suggested this project he believed he could do it in a weekend. I took a few wrong turns on this project and spent the majority of it learning the systems I was working with. Because I didn't have the requisite

technical understanding at the beginning, I had to spend a lot of time learning and going down rabbit holes which were not on the critical path of the project. This was very useful for my learning, but it was not the shortest path to completing the project. In contrast, last Saturday I selected parts and designed the circuit at a high-level for our next generation vehicle's shunt current sensor in a few hours.

UBC Solar has a few systems in place to mitigate the risk of novice engineers (as I was!) going down suboptimal paths:

1. Design Review meetings and documents
2. Integration meetings with other subteams
3. Detailed Documentation

Throughout a project's lifecycle UBC Solar has a few primary design review documents and associated meetings. Design Review 0 (DR0) documents set functional requirements and constraints for a project. DR1 documents explore solutions that meet the given requirements. DR2 documents are detailed designs ready for implementation. DR3 documents are mostly reflection, summary, and lessons learned.

This process makes the current state and direction of a project clear at any given instant, but requires a lot of overhead. These processes were designed for novice engineers to avoid common pitfalls, but new members are often the ones who appreciate this structure the least (it is mostly writing and meetings). You need to get a few scrapes and bruises before you appreciate the value of such processes (as I did).

"In fact, my sole interest lies in improving the ability of the car to score points and what helps me do that is my experience across the disciplines."
- Adrian Newey, How to Build a Car

Integration failures occur when members of a team do not have a full understanding of how their system affects other systems in the car. UBC Solar's current electrical lead Aarjav Jain has elected to solve this problem through numerous integration meetings between subteams. I'm particularly happy about this new process, for Conway's Law (https://en.wikipedia.org/wiki/Conway%27s_law) has shown up at UBC Solar far too many times. However, an excess of integration meetings slows progress without significantly decreasing the risk of Conway's Law.

"Engineers needed to become operators. There was a hell of a difference. An engineer can explain how a system should work in theory, but an operator has to know what the engineer knows and how the systems tie together to get the mission accomplished."
- Gene Kranz, Failure is Not an Option, Chapter 3

Strive to be the kind of operator Gene Kranz describes. You can't just know how individual systems work in theory; you must understand how they work together in practice.

"What are they doing? Compiling one million sheets of paper every day. Reports and data that no one in the bureaucracy has either the time or the interest to read."
- Ben R. Rich in Skunk Works on the cost overruns due to countless USAF auditors on the B2 bomber program.

Each additional process imposes a cost. We must be careful not to overburden our team with excessive documentation and meetings; it's a fairly thin line to walk. The most elegant and beautiful solution (https://ckalitin.github.io/ideas/2025/09/29/must-be-beautiful.html) to this is having a competent enough team that these processes are not required.

Team Progress Asymptotically Approaches Zero Without Clear Functional Goals

"We must revitalize NASA. Lacking a clear goal, the team that placed an American on the Moon, NASA, has become just another federal bureaucracy beset by competing agendas and unable to establish discipline within its structure."
- Gene Kranz, Failure is Not an Option, Epilogue

During the summer I visited San Francisco (https://ckalitin.github.io/space/2025/06/02/impulse-master-plan.html) and had the chance to talk to Kevin Huang who works on the Tesla bot. He described how at any given moment the Tesla bot team is working towards a given functional goal, and course-correcting based on whether they are coming closer to or further from that functional goal. A constant clear functional goal means everyone is on the same page about what work must be done, and what work is not useful.

My current sensor characterization project on UBC Solar is an example of what happens when you don't spend every working minute optimizing for a functional goal. You end up spending time on side tangents that don't bring you closer to success.

"The outcome of any given company is the vector sum of the people within it."
- Elon Musk (https://x.com/elonmusk/status/1871997501970235656?referrer=grok-com)

As organizations increase in headcount, the risk of an individual engineer not working in the direction of a common functional goal increases. This becomes particularly problematic when an organization does not have a clear functional goal. Hence, the progress towards a supposed goal asymptotically approaches zero as the goal becomes less clear and team size increases. Casey Handmer has written about how progress may even become negative, not approach zero - Dittemore's Law (https://caseyhandmer.wordpress.com/2025/01/17/dittemores-law/).

Finally, How Do You Find Exceptionally Competent Engineers

“You find your key men by piling work on them. They say, I can’t do anymore. And you say, sure, you can. So you pile it on, and they’re doing more and more. Pretty soon, you have men you can rely on absolutely. You have an organization that can really get things done.”
- Founders Podcast: #66 Henry Kaiser

Finding exceptionally competent engineers is a function of both recruiting engineers that show promise and developing / discovering their competence in organizations that bring their abilities to the forefront. Some time in the last couple of months Casey Handmer went on a podcast and described how the Musk companies are able to sift through a large number of engineers to find those that are maximally competent (https://x.com/CJHandmer/status/1964453403809239115). It’s not purely a function of recruiting, but as Henry Kaiser showed, you must sift through a large number of engineers by piling work on them and finding those that can handle and excel at it.