A Gonzo Engineer's Take On How To Manage An Engineering Project

While trying to write something at work which was beginning to look more and more like a grant proposal, I started running through my mind how I would run the project, if it came to that, and I drew up a list of things that define the Gonzo Engineer Program For Engineering Success. And here it is!

1) Get a clear statement of the problem. Use that to define the endpoint of the program and what the data charts will look like (IF POSSIBLE: draw the charts, scale and label the axies ahead of time, but leave them blank until later).

2) Start with one leader, instead of many. Goes without saying.

3) Start with an experienced leader. Want a disaster? Assign a newbie who is ignorant of the fundamental physics to lead the team. And don't think for one second that by doing so, you are thereby furnishing that person with a "career growth opportunity". You are setting them up for failure.

4) Start with a small team, instead of big… but not TOO small. The smaller the team is, the easier it will be to manage. However, if you never achieve "critical mass", you'll never achieve success within your schedule.

5) Hand-pick the team. Want to fail? Take the people you can get instead of the people that you need. If your manager says, "sorry, you'll just have to make do with what you can get", you reply, "so I guess this project really isn't that important, is it? Remind me again why I'm supposed to work on it."

6) Start with a short list of modest goals instead of trying to solve all the world's problems at once.

7) Start nimble (think: Millenium Falcon), instead of ponderous (think: DeathStar). Be fully prepared to turn on a dime, in light of new information/ data.

8) Start cheap instead of expensive. The cheaper you start, the easier it will be to get funded.

9) Start short instead of long. Even if an experiment performed in one day only yields single-digit accuracy or perhaps just a yes/no answer, you can still build upon that right away and progress faster than if you waited a week for two-digit accuracy. And to enlarge upon that topic…

10) A fast approximation is priceless, and tardy precision is worthless. One-place accuracy is infinitely more useful than no data, and confidence in the face of imprecise measurement can be achieved with a larger sample size.

11) Start now, rather than later, even if understanding of the problem is incomplete. Do not wait for a complete theoretical understanding of the problem to be in hand before you begin, because you will almost never be so lucky.

12) Start now, rather than later, even if not everyone has signed off on the program. Better to beg forgiveness with data in your hands than to waste time waiting for permission.

13) Leave elegance to tailors. Your job is to be right, and put food on the table. Shotguns, as inelegant as they may be in comparison to other weapons, nonetheless do this every day of the week. (So do hand grenades, for that matter, if you happen to be fishing instead of hunting…)

14) Start the effort with the absolute single biggest and simplest, yes/no, "is it 1, 10, or 100" class of question you can ask. The answer is not intended to solve the problem, but instead to rule in or rule out whole classes of root cause (or, alternatively, quadrants of the solution space) as quickly as possible so you can focus your attention on a smaller piece of the solution space and home in on the truth.

15) Upon receipt of the first yes/no answer, lick wounds and regroup if necessary, but reformulate the next question to be answered in accordance with what was just uncovered BEFORE proceeding to look for the answer to the next question on your list.

16) Build-test-fix-build again (iterate the overall process in small, frequent steps) to rapidly & incrementally converge on a satisfactory approximation of success. Rapid iteration-and-convergence in small steps (many build-test-fix cycles i.e., "design iterations") of this sort enables useful progress (convergence upon the truth) even in the absence of an all-encompassing, theoretical understanding of the problem from the outset, which you will rarely, if ever, be blessed with. Plan on building up understanding incrementally, and don't schedule breakthroughs.

17) One measurement is a clue. Ten measurements is data. A hundred measurements is information.

18) Just because you have no way to accurately measure something does not mean it is unimportant. And just because something is easy to accurately measure does not mean it is critical.

19) If you have time enough to throw rocks, you have time enough to help. Use this line to make people who question the worth of your project go away.

20) Do not expect to win the ball game with a grand slam. You increase your risk of striking out if all you do is swing for the fence. Load the bases and drive the runners home one at a time. Although you might be able to cut the schedule or budget in half with one home run, don't write it into your schedule.

21) Regularly compare project status to endpoint goal (is that chart being populated with data?) to measure progress.

22) At endpoint, hand off the completed product to the implementation team with sufficient documentation to allow them to crank out more product without your direct involvement.