

# 18 start-up mistakes from Y Combinator founder (part 2)

Choose the wrong platform, launch the product at the wrong time and the irrational spending is the startup killing cause given in this section.

Following part 1, the reasons for failure startups are given in this part 2 which focuses on the following topics.

7. Choosing the wrong platform
8. Launch the product (Launch) too slowly
9. Launch too soon
10. There is no specific user object
11. Calling too little capital
12. Spending too much

## 7. Choosing the wrong platform

A fairly relevant issue (because it often happens that you choose to be a bad programmer - the 6th cause) is a wrong choice. I think a lot of start-ups during the startup bubble killed themselves by choosing the application built on the Windows server. Hotmail still runs on FreeBSD for years before being bought by Microsoft. If the Hotmail founder chose to use Windows, they probably fell into a quagmire.

PayPal chose to avoid this "bullet". After entering X.com, the new CEO wanted to switch to Windows even if PayPal co-founder Max Levchin pointed out that their software increased by only 1% on Windows and Unix. Fortunately, PayPal later changed the CEO.



The foundation is a vague word. **It can mean the operating system, programming language or "framework" framework built on programming languages. It shows support as well as limitations, just like when you build a foundation for a house.** One scary thing about choosing platforms is that there are always options

outside of safety, like Windows in the 90s, that will ruin you if you choose them. Java Applet may be a good example, supposedly a new way to launch applications. It is estimated that it killed about 100% of startups who believe it.

How do you choose the right platform? The most common way is to hire a good programmer and let him choose for you. If you are not a programmer, you can visit the offices of computer science companies and see how they use them in research projects.

## 8. Launch the product (Launch) too slowly

Whether big or small, the process of completing software is also a difficult period. **The intrinsic nature of the launch method is that software is always 85% complete** . Efforts should be made to overcome this and release it to users <sup>(3)</sup> . Startups always give many reasons to delay the launch of their products, mostly the causes that people often use to delay other things in everyday life. Is there something that needs to be done first? Maybe so. But when your software is 100% complete and ready to be released, can users wait?

One reason for quick launch is that it forces you to actually complete the inertia of the job. **No product is actually completed until it is released** , you can see the urgency when releasing any product, no matter how much you think you have completed. Another reason is that only when you give your idea to the user will you fully understand it.



Some internal problems itself appear when there is a delay in **launching the product: working too slowly, not really understanding the problem, afraid of dealing with too many users, afraid of being evaluated, working with too many products, excessive perfection** . Fortunately, you can beat them all in a simple way, forcing yourself to launch a product very quickly.

## 9. Launch too soon

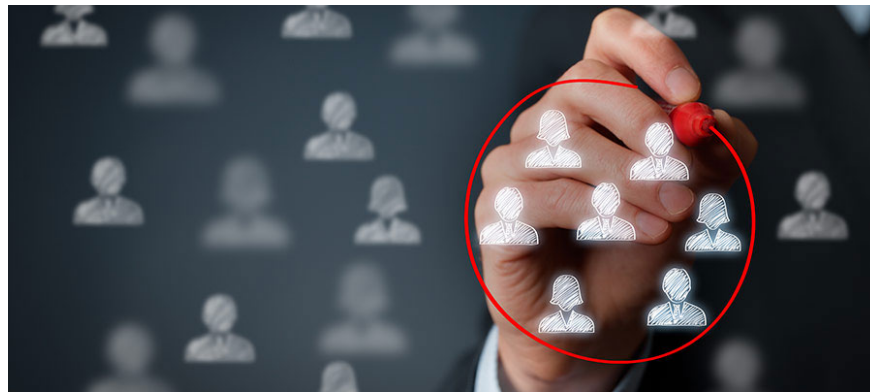
Launching too early probably killed hundreds of times more startup than launch too early but fast launch is not impossible. The danger is that you ruin your reputation when you launch the product, the trial of early adopter and if it is not good, they will never return. **So what is the minimum to start the product launch?** I suggest that startups should plan what they will do, identify the core that (a) is itself useful and (b) can gradually expand throughout the project and try to improve quickly. possible.

That is the direction that I (and many other programmers) use when writing software, thinking about the overall goal and starting to write for the smallest branch in it but still worth using. It is a minor detail, but anyway you still have to write it and in the worst situation, it is not useless to do it. **You will realize that doing a small branch is good for the spirit of work and help you see clearly what the rest will have to do.** The first recipients you need to impress are also quite generous people. They do not expect a new product to be released to do everything, they just need it to do something.

## 10. There is no specific user object

You can't build what people like if you don't understand them. I mentioned earlier that most of the early startups tried to solve the problem encountered by the founder. Perhaps there is a rule here: **the value of what you create is proportional to how much you understand the problem you're trying to solve** . And the problem you understand best is your own problem. <sup>(4)</sup>

Surprisingly, there are still founders who assume that someone - someone they don't know for sure - will need the product they are building. The founder is not in the target market, then who is it? Could it be teenagers? Who do business?



Of course you can build a product for non-friends. We did that too. But also realize that you are entering a dangerous area. You are flying on a vehicle, so you need to (a) be aware of how to control it instead of assuming it is based on intuition as usual and (b) keep an eye on your vehicle.

In this case, the media is the user. **When designing products for others, you have to be very realistic** . You can't guess how effective it is to find users, measure their feedback. So if you want to create products for teenagers or business people or any group that doesn't include you, let some users try your product. Otherwise you will go the wrong way.

## 11. Calling too little capital

Most successful startups call capital at some point. When there is more than one founder, this seems to be certain, but how much should there be? Start capital is measured by time. **Each startup does not make a profit (meaning all startups in this group) have a period of time before they run out of money and are forced to stop working** . This is sometimes called the runway and used in the question "How long is the runway?". This is a good metaphor because it reminds you that when you run out of money, you "die".



Too little money will not be enough to take off. Taking off also means different depending on the situation. Often you will have to go to higher steps; If there is an idea then it is to give a sample, if there is already a prototype, then it is a product release, if the product has been launched then it is growth. This depends on the investor because you can only convince them to make a profit.

So if you call capital from investors, call enough for the next step no matter what the step is.<sup>(5)</sup> Fortunately, you can control your spending as well as what your next step is. We recommend that startups choose the two criteria at the lowest level. **From the beginning, spend very little and set a simple goal to build a solid prototype.** That is how to bring maximum flexibility.

## 12. Spending too much

It's hard to distinguish between spending too much and calling too little. When you run out of money, you can blame both. The only way to determine is compared to other startups. If you call 5 million and still run out of money, you probably spend too much. Over-spending is no longer popular today because the founders seem to recognize the lesson for themselves. In addition, starting a startup now costs less. By the time this article came out, there were quite a few startups spending too much, none of us had to spend this money to make this mistake (not because we invested little, many startups continued to call for more capital) .



The most classic cause is hiring too many people. This will be twice as harmful: not only will it increase costs, it will also slow down your business - the money must be long-lived and spend too fast. Most hackers understand this. Fred Brooks also explained clearly in *The Mythical Man-Month* book. We can offer **3 tips on hiring personnel** (a) if possible avoid doing (b) paying with dividends instead of not only saving money but because you want that person to commit to work more (c) hire only people who write code or can get customers, because those are the only ones you need at the beginning.

(3) Steve Job has taken the morale for everyone by saying "real artists ship" - Steve Jobs's idea is that many people can have idea but only true artist can convey that idea). This statement is good but unfortunately it is not really true. Many famous works of art were left unfinished. This sentence is true in areas with clear deadlines such as architecture and movies, but even in those areas, people tend to create variations before it really comes out.

(4) This is probably the second factor, startup founders tend to lead the technology so the problems they encounter are probably well worth it.

(5) You should get more than you think you need, perhaps 50% to 100% because the software often takes a lot of time to write and the time to reach an agreement will be longer than you think.

You finished reading the article "**18 start-up mistakes from Y Combinator founder (part 2)**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.