

I was recently invited to present my thoughts on the current state of software startups. Here are some of my remarks (Vince Sonson, Sullivan's Island, SC, July, 2017):

The key to a successful software startup, like any new business, is getting to product-market fit as efficiently as possible. What's enticing about software startups is that you can usually do so on a very limited budget, since there are few or no capital expenses. That's why so many people try it.

The failure rate is still very high, which is not a knock on those startups or their ideas per se. There are many reasons a software startup can fail, again like any other business.

Knowing the likelihood of failure is high, my opinion is, if my product is going to fail in the market, I would rather know as early as possible, to minimize the amount of time and money I've spent on it. Sounds logical, right? So why do so many software startups fail at "failing?" Why can they not get to a validated product-market fit - or not - efficiently? Here are three of the most common reasons I have found:

- They try to build the perfect product from behind their ivory towers before letting users get their hands on it
 - they fail to understand the importance of NIHITO
- They use the wrong metrics to evaluate how the market is receiving the product
 - for those that do get their product in the hands of users quickly, they often default to "vanity metrics" to evaluate their uptake, which may not be appropriate and can produce false positives
- they spend too much, too early, in the wrong areas
 - this is especially true for well-funded, venture-backed startups

These are just three reasons, there are many more of course.

So what is some advice I typically give early stage software startups in light of this:

- Run lean - bootstrap the operation if possible - and spend the majority of your time and resources in the field with your target market, really honing in on the personas and problems you are trying to solve.
- Use low-fidelity guerrilla techniques if necessary. My teams use hand sketches and iPhone video recording, for example.
- Once you hit a pain point with a product prototype that someone is willing to PAY for (that's a key - many users will say they like a software product and would use it, but they wouldn't necessarily pay to do so), then:
 - start testing business models and sales and marketing strategies to get after it
 - start entertaining offers for external investment (if you need it - otherwise keep bootstrapping it)
 - figure out what differentiates you from your competition and build mechanisms to defend it fiercely (could be IP, could be business model, etc. - it is rarely, if ever, the code)