Why do personal software projects fail? (i.e. projects with the goal of leading to fulltime income) [closed]

Asked 16 years, 2 months ago Modified 14 years, 7 months ago

Viewed 2k times

16

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 5 years ago.

Improve this question

What are the main reasons personal projects (software apps etc) never get to the level of competing with your salary?

To me one big problem is "on-the-fly" feature expansion, with this problem, the end only gets further and further away!

project-management misv

Share

Improve this question

Follow

edited May 25, 2010 at 10:15

Georg Fritzsche

98.9k ● 26 ● 196 ● 236

asked Oct 17, 2008 at 19:20

Anonymous Cow

Didn't this post used to be community wiki? I just noticed I'm getting rep from my answer to it ... – John Rudy Nov 10, 2008 at 22:47

14 Answers

Sorted by: Highest score (default)

▲

37

▼

For me, it's simple: I work 8 hours a day already. I spend a few more hours a day keeping current. I have ~~a girlfriend,~~ some local family and a decent circle of friends. I have (gasp) non-computer-related interests and hobbies. In other words, I have a life.

So ... **Time.** Time is not on my side. Would that it was ... My blog might be a bit more current if there were just two more hours in every day. :)

(Originally posted by John Rudy.)

Share Improve this answer

Follow

edited May 23, 2017 at 12:07

community wiki

3 revs
John Rudy

21

If you want your hobby to become your job you have to acquire all the other skills you need to be in business. At the end of the day your pet project has to stand on its own two feet in the real world. At the same time you are enjoying the coding you need to get yourself a concrete plan to commercialise your activity.

Most hobby projects fail to make the big time for one of two reasons:

1. The idea is not commercially viable
2. The discipline necessary to commercialise the idea is missing

Just because you are a great technologist does not mean you'll be a great businessman. You may be, but the two are not necessarily linked. It is no weakness to consider partnering with someone who has no technical skills but a good network and some proven business acumen. Quite often people like that are looking for techies too so you might find a great partnership. That person can provide the structure and commercial discipline that you probably lack if feature creep is pushing your completion backwards.

Share Improve this answer

Follow

answered Oct 17, 2008 at 19:46

Simon

80.6k ● 26 ● 92 ● 119

True - most small businesses fail and software is no exception. Especially if it's not treated as a business. – pjc50 Jul 10, 2009 at 10:53

1 I think time is a red herring. The conclusion must be that all the people who are failing simply because of time could be internet millionaires if it were just for a few extra hours. That's almost certainly not true. What's more, if it is just a question of time because your idea is a dead cert, then you're a fool not to pursue it and probably lack the commercial drive necessary to be an entrepreneur, or you are concentrating on the wrong thing and should be raising money for someone else to spend the time. Either way I think time is often an excuse, rarely a reason for failure. – Simon Feb 8, 2010 at 8:55

▲

10

▼

I think the primary reason is the simple work overload that most developers experience. Most personal projects take place in the evening and weekends, and as excited as most of us get about our ideas for personal projects, after 40 hours (or more) of salaried programming, it's hard for "more work" to compete with watching a game while sipping a beer or spending quality time with the family.

Share Improve this answer

Follow

answered Oct 17, 2008 at 19:27

Raelshark

3,055 ● 3 ● 29 ● 36

7

Different skill sets are required to start and maintain a business than to develop software. Entrepreneurship skills can be learned, but not every has the skills to make it happen. A lot of times the skills it takes to get something started and off the ground are different than the skills it takes to finish it and polish it. For me, I know that I have the creativity to make software and find ways to solve problems, but I have little interest in finding funding for a business and marketing a product or service.

Share Improve this answer Follow

answered Oct 17, 2008 at 19:32

tvanfosson

532k ● 102 ● 699 ● 798

6

Assuming that you're a developer, it's most likely due to the fact that you do not know when, or are incapable of, stopping development and focusing on other things, like marketing and sales.

Share Improve this answer Follow

edited Oct 17, 2008 at 20:38

Josh

8,016 ● 5 ● 43 ● 63

answered Oct 17, 2008 at 19:59

Greg Dean

30k ● 15 ● 69 ● 79

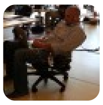
This has been and continues to be a big problem for me. Fortunately, I have someone (my wife) who can say: "It's done already! Just ship it!" when I'm constantly tweaking things. – Mark Bessey Oct 17, 2008 at 21:10

5

Time and Losing Interest, there is always a new tool or technology that can take your attention away from completing projects.

Share Improve this answer Follow

answered Oct 17, 2008 at 19:28



Anson Smith
6,204 ● 1 ● 25 ● 10

5

I'm not sure if I understand your question, but here are a few answers:

- Adding "on-the-fly" features isn't necessarily a bad thing. In fact, it's the expected model of Web 2.0 and Web 3.0 projects. The key is to keep them very simple, only roll them out once they've been tested, and listen to your users. If you try to dump the kitchen sink in on the first release, it will most likely be ugly, confusing, and buggy.
- Being a great programmer is only a part of it. You need business skills, marketing, knowledge of the user's needs and how to meet them, artistic/design skills, and a hell of a lot of luck.

- Lot's of people have great ideas. Often different people have the same ideas. Most never get implemented. Of those that do, very few of them succeed. In some cases, revolutionary products took years to convince the buyers and users that they even wanted the product. Often the people or companies behind the first few iterations failed miserably and then a third or fourth person or company finally hit the market at the right time with a right product. Apple is great at both ends of this by the way - they not only innovate (first Mac, the Newton, etc.), but they also wait until the market need grows and they sense a place to pounce in and take advantage of it (the iPod, the Mac vs. Windows issues, etc.)

Most of these bullets apply as much to software as they do to widgets and services. The big advantage that software has is lower startup costs. Just like the saying "On the Internet, no one knows you're a dog" - "When looking at a web app, the user doesn't know if you are a multi-billion dollar company or a single guy sitting in your underwear in your parent's basement." If your software is *good*, that is...

Share Improve this answer

Follow

answered Oct 17, 2008 at 19:35

CMPalmer

8,623 ● 5 ● 42 ● 50

3

I'd say one of the big reasons is that by nature, personal projects don't get as much attention as your job will.

I have a slew of personal/side projects I'm working on, but they get far less of my attention that my 'real' work does because, right now, that's what's paying the bills.

If I were to take a month off and work *only* on my personal stuff, it'd probably be pretty cool / worth money.

Share Improve this answer Follow

answered Oct 17, 2008 at 19:26

warren

33.4k ● 23 ● 89 ● 128

3

1. developers often design for themselves instead of for their customers
2. developers tend to put off releasing products until things are 'perfect' - and they never will be

Share Improve this answer Follow

answered Nov 10, 2008 at 19:55

Steven A. Lowe

61.1k ● 19 ● 135 ● 204

0

- Weakness of mind and spirit. Build a team around your product early.
- Scope creep. Concentrate on selling what you have already got: *"The customer can have any color he wants so long as it's black". Henry Ford*

- Small feature set. Leverage features of your product by what is already available on the market.
- Not enough hours spent daily. Often achieving something might depend just on simple routine, putting your time in.

Share Improve this answer Follow

edited Oct 18, 2008 at 14:41

answered Oct 18, 2008 at 14:24

Vlad Gudim
23.5k ● 16 ● 71 ● 92

▲ 0 ▼

Deep down I think its a lack of belief in the project. If I believed in what I was doing I would not stop in completing the project.

Share Improve this answer Follow

answered Oct 18, 2008 at 14:54

MikeJ
14.6k ● 22 ● 74 ● 89

▲ 0 ▼

- Desire to build an ideal product
  - For example: There are various ways (algorithms) to get a particular task done. But, people wait to discover that one ideal solution. Even if there are multiple solutions for the same

problem already available. That ideal solution is never found.

- Procrastination

Share Improve this answer Follow

edited Oct 18, 2008 at 14:55

answered Oct 18, 2008 at 14:48

Vivek

16.5k ● 5 ● 32 ● 37

0

Your personal software projects don't compete with your salary for one reason.

What do you **do** for your salary? Whatever that is -- however much you may like or dislike it -- it *more* valuable than your software product.

"But my day job involves a lot of stupid time-wasting meetings." So? Clearly, someone will pay you more for wasting your time in meetings than for your software products.

"But my day job forces me to waste months in useless analysis and design documents and test plans that never even get used." So? Clearly, someone thinks this activity is more important than writing software.

"How can meetings or useless documents be more valuable than software?" I don't know, but look at your experience. Companies love to pay programmers relatively large amounts of money to hang around and waste time.

Companies don't love to pay for software.

Your personal projects don't compete with your salary because your time is more valuable than your products.

Share Improve this answer Follow

answered Oct 19, 2008 at 1:07

S.Lott
391k ● 82 ● 517 ● 788

if this were true, then no one's hobby projects would ever compete with their salary. UltraEdit comes to mind... – Steven A. Lowe Nov 10, 2008 at 19:54

@Steven A. Lowe: Disagree. I think you have to step outside the conventional 9-to-5 salary paradigm, and take a big risk to see a hobby turn into a product. – S.Lott Nov 10, 2008 at 20:01

why do you think that? the hard part is selling it, not writing it ;-) – Steven A. Lowe Nov 10, 2008 at 20:58

@Steven A. Lowe: Selling is only part of the issue. The issue is that employers pay for a lot of dumb things that don't seem to create a lot of value. But that's what employers like to pay for. That's why hobbies rarely pay. – S.Lott Nov 10, 2008 at 21:08

@Steven A. Lowe: While perhaps not always true, it's generally true. There are far more salaried programmers going to meetings than there are commercially viable pet projects. To be a success you need good software that people want, marketing and luck. – Robert Paulson Nov 11, 2008 at 0:35

-1

The biggest reason? Because if you can write it yourself and people like it, someone else can make an open source version with much better support than you can provide alone. Why not skip the middle man and release it as open source yourself? Sure, you miss out on the direct profit, but that looks very good come hiring time.

Share Improve this answer Follow

answered Oct 17, 2008 at 20:16

Kirk Strauser

30.9k ● 5 ● 52 ● 69

That is all well and good unless you have no interest in working for someone else. – TWA Mar 20, 2009 at 0:08