How do you explain to a sales person that programming is really difficult and takes time [closed]

Asked 15 years, 11 months ago Modified 12 years, 7 months ago

Viewed 1k times

8

I often work with sales and marketing types that cannot figure out how to use Excel, let alone understand the scope of their requests from a technical perspective. Of course, it would not be fair to expect them to, but that still leaves me with a problem.

What is the best way to show marketing and sales types that they have asked for something that requires a lot of complex programming and some patience?

Could you please share examples of problems and solutions?

Could you please recommend books on this subject?

Thanks!

estimation

Share

Improve this question

Follow

edited May 4, 2012 at 20:43

MPelletier

16.6k ● 18 ● 89 ● 140

asked Jan 20, 2009 at 4:47

J3r3myK

1,447 ● 2 ● 13 ● 21

3 Tell them that if they can double the profit of the business in 24 hours then you'll have your product finished in 24 hours. – Mike B Jan 20, 2009 at 15:50

8 Answers

Sorted by: Highest score (default)

15

Break the problem up into as many sub-divided tasks as possible. Provide a per-item estimate in hours beside each one.

When they think of a project as a whole, it seems simple. However, when they see each individual thing that must be done and the number of hours each item will require, it is putting it into terms business people can understand. Suddenly the software solution they want isn't a "black box" to them anymore and they now have some insight into the process.

If you are looking for books I would suggest Software Estimation - Demystifying the Black Art.

Share Improve this answer Follow

edited Jan 20, 2009 at 7:34

answered Jan 20, 2009 at 5:06

mmcdole
92.7k ● 61 ● 188 ● 224

Dividing into numerous subtasks is a good suggestion. Often when non-technical people see a long list they assume that equals a lot of work. – J3r3myK Jan 20, 2009 at 5:31

9

The computer will do what you told it to do, not what you want it to do.

Any form of abstractions needed be translated to exact details.

source http://c2.com/cgi/wiki?TeachMeToSmoke

```
Teacher: "It's hard to express ourselves clearly.
You're a smoker, right?
Are you pretty good at it? [Student nods.]
Let's pretend I'm a man from Mars and you are
going to teach me to smoke.
Do you have a fresh pack? Let's start with that.
[Takes pack.] OK, now tell me what to do."


Student: "Tear open the pack."

T: [Tears pack to shreds. Cigarettes fly
everywhere.]
S: "No, no, tear off the top of the pack!"
T: "OK, sorry, do you have another pack? No? OK,
let's just start with this cigarette. [Picks one
up.]
S: "Put it in your mouth."
T: [Puts whole cigarette in mouth.]
S: "No, no, just put the end in your mouth!"
T: "Sorry." [Tears filter off, puts whole filter
in mouth.]
S: "No, no, don't tear the cigarette, just hold it
between your lips!"
T: "Oh, sorry, give me another one." [Places new
cig sideways between lips.]
```

... and so on. You can play the game for a long time. It's hard to give clear instructions, even when you know the domain. Programming will endure for a long long time. --RonJeffries

Share Improve this answer   answered Jan 20, 2009 at 6:02

Follow

Michael Buen
39.3k ● 10 ● 97 ● 120

I've read a variation of this with how to make a peanut butter and jelly sandwhich to teach children about programming.
– mmcdole Jan 20, 2009 at 6:57

5

I had a friend who could do the Rubik's Cube in seconds.

That made me think of this way of explaining to my manager why did our latest project FAIL!

Olivier takes an average of 10 seconds to completely sort all colors of a 3x3 Rubik's Cube after looking at it for approximately 5 seconds.

If you ask him to make an estimate of how long it will take to sort it, you give him the cube, start the clock and after 5 seconds he will say:

"OK, as soon as I start I will be done in 10 seconds"

you smile and say: "Start!" After 3 seconds you ask him to stop.. give him another Rubik's cube and say.. sort this one instead...

4 seconds after he starts the second Rubik's cube, how long do you think he will take to sort the first one again?

If you answered 7 seconds approximately, congratulations: You're upper management material!

(and Olivier would be rightly entitled to force you to eat the cubes)

Share Improve this answer Follow

answered Jan 20, 2009 at 7:51

Ric Tokyo
6,575 ● 3 ● 32 ● 48

Yip. Human task switching is not good. In my job, I have decided to do one thing at a time and do it properly and finish it before doing something else. – Geoffrey Jun 22, 2009 at 20:27

3

I agree with Simucal in the sense that managers tend to do better when you break a problem into hours, rather than into programming tasks. For example, saying to your boss, "That should take about two hours to complete, but I have a few other things that I have to complete first, so I should have it to you by tomorrow." is a lot more useful than saying, "Well, first I have to design an interface to communicate between objects, and then create the classes to implement the interface, and so on." Managers understand what they can see, so anytime you can explain your task in terms of end-user effects, you will likely have more success.

With that said, don't let your manager intimidate you into making promises that you can't keep. You may know that all they want to hear is "I'll have it by the end of the day.", but if you know it can't be done, don't say that it can, hoping that if you have it to them sometime in the next couple days, that it will be "close enough". If you start factoring in time for designing and testing and give them appropriate estimates, eventually they will start to understand how long it takes to accomplish certain types of tasks, and stop expecting everything to be done by yesterday.

I've also noticed that tangible results along the way tend to put their nerves at rest (temporarily, at least). My boss starts demanding finished results when he starts to panic as to whether or not a task will be completed on time. However, when he is able to "see" the step-by-step progression, then he is more likely to understand that we are, in fact, making progress, even though it isn't in the finished product yet.

Also, as you start this process, try to look at things from their point of view, and understand that until you get to a point where you can spend the amount of time you think is necessary, you may have to find a happy medium. There came a point in my experience where I needed to develop a Cache object, and while I would have loved to take several weeks to design and implement a configurable and extensible Cache that could be widely distributed across multiple applications, I had to limit myself to the task at hand. Just make sure that if you decide to scale back or follow through with a short-sighted design, be sure that it is well-documented so you can go back and fix it when you have time (or so another developer can pick up on the train of thought that you were unable to finish). Also, don't sacrifice good coding standards and style, as this will also make your code easier to maintain and update properly in the future.

Good luck!

Share Improve this answer Follow

answered Jan 20, 2009 at 7:01

jeremyalan
4,786 ● 2 ● 31 ● 39

Thanks for the suggestion. This is a great reply. – J3r3myK Jan 23, 2009 at 9:36

2

This one may be a good book for non-programmers to understand some of these issues and pitfalls of runaway requirements:

[Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software]()

Share Improve this answer Follow

answered Jan 20, 2009 at 5:10

Turnkey
9,406 ● 3 ● 29 ● 36

Ditto. This is a great book. Thanks for the suggestion.
– J3r3myK Jan 20, 2009 at 5:26

I'd be curious to see if it contained any advice that wasn't in the classic: en.wikipedia.org/wiki/The_Mythical_Man-Month including consideration of en.wikipedia.org/wiki/Brooks%27s_law – IRTFM Oct 6, 2015 at 0:25

1

In all seriousness, I think the best thing it to actually tell them that some things are complex and do require complex problem solving, analysis and design. There is a gap between what they do, and what the programmer does and its unfortunate that they will never understand the full implications. You sometimes just have to be firm and explain that it can take alot of time.

Perhaps a breakdown of the task into subtasks and giving them estimates may help.

Share Improve this answer Follow

answered Jan 20, 2009 at 5:14

Jobo
916 ● 6 ● 11

Generally one person does not understand how he (or any other person of either gender) actually thinks. The "thinking" done by sales people is often multi-dimensional and inherently fuzzy. The thinking needed to hammer together a software projects is quite different. – IRTFM Oct 6, 2015 at 0:31

1

Make sure you understand their issues too. People will often bring solutions to the table ("we need this feature") rather than start with root business needs. The more you understand the problem the more likely you are to be able to suggest a compromise.

On occassion I've been told a certain large feature is absolutely essential, but I've been able to deploy much simpler solutions that substantially addresses the problem. Sometimes these interim solutions have grown into vital features, just as often I've been able to remove them two releases later without anybody noticing.

Share Improve this answer Follow

answered Jan 20, 2009 at 5:26

codybartfast
7,543 ● 3 ● 23 ● 25

0

In my experience, whenever I started to explain to sales people in the past why a task takes a certain amount of time, they quickly admit that they do not really want to know the technical details, and I am fine with that. I usually do not want them to explain to me why they still have not nailed down that big sale after n days either. To do work effectively, everybody has his own area of responsibility. Just make sure that your relationship with the sales people that you provide estimates for is good and they trust in your ability to do proper and reasonable estimations and get the work done. IMHO there should be no need to explain and reason an estimation in every detail, but if there is, I would say the real problem lies elsewhere.

And I wholeheartedly agree with "It depends" above.

Share Improve this answer Follow

answered Jan 20, 2009 at 6:02

Timo
1,088 ● 1 ● 10 ● 19