How can a software agency deliver quality software/win projects? [closed]

Asked 16 years, 2 months ago   Modified 8 years, 4 months ago

Viewed 2k times

26

I currently work for a bespoke software agency. Does anyone have any experience of how to win well priced work?

It seems there is so much competition from offshore/bedroom programming teams, that cost is extremely competetive these days. I feel that it is very different compared to a software product company or an internal IT department in terms of budget.

As someone else said before, we only ever really get to version 1.0 of a lot of our software, unless the client is big enough. In this case it doesn't make business sense to spend ages making the software the best we can. It's like we are doing the same quality of work as internal IT staff. Also a lot of our clients are not technically minded and so therefore will not pay for things they don't understand.

As our company does not have the money to turn down work it often goes that we take on complicated work for far too little money. I have gotten a lot better at managing change and keeping tight specs, etc. It is still hard.

Edit----------------------

Almost 3 years on from this post and I can list some important lessons that I have learnt since then.

Please see below for my answer

project-management   bids

Share

Improve this question

Follow

6   joel and jeff give their answer in their podcast
blog.stackoverflow.com/2009/05/podcast-54 (around min 35)
– dev.e.loper May 22, 2009 at 20:44

9 Answers

Sorted by: Highest score (default) ⇕

9

If you are concerned with doing too much work for too little money then work on an hourly basis. Yes, that is harder to sell in most situations.

Maybe you can try a two-phased approach instead. Have a very short initial engagement where the deliverables are very specific requirements documents that become the property of the client. You risk having to compete for the actual development but you take away the risk of pricing the project too low because you will already understand what the client is like to work with, as well as, the application requirements.

Once you win the work at a fair price then use the best practices suggested by mathieu to help ensure quality and productivity which both lower the cost you incur.

Share Improve this answer

Follow

answered Oct 15, 2008 at 13:35

Ed.T
1,695 ● 2 ● 15 ● 19

Excelent points, we do this when we can, and if the client dosent choose us for the build, then we charge more for the spec. – Chris Barry Oct 31, 2008 at 10:21

5

What you described in your post, (not your question), I think is a sales, management and marketing question first and foremost.

You say that your clients are not technically minded, this will require to have a cohesive sales, consulting and communications strategy, this isn't about programming skills.

Also, if your company constantly accepts projects that are too complex or expensive for your team, and you deliver low quality products you'll sooner or later be stuck in a hole. You will attract customers that you do not want, and existing clients will be turned off by your 'incompetence' and sooner or later find another company on which they'll try to play the same price game. Those clients are worth nothing in my opinion.

You ask 'how do you win well priced work'? People are social animals, they talk with each other. If there's a market perception that you are an unreliable company, people and future clients will sooner or later know. Customers don't care whether you offered them a product at a really low price, on a too tight schedule - It's not really their error, it's you who accepted it. So once again, I think whole ordeal is a bad business practice.

I found that you really have to define tight specs on jobs with low budgets, define what you will and can deliver, tell them the price, stop your boss from offering too many long term customer discount price tags because they are too afraid to lose the client. Communicate early and often when things start to get out of hand. Write precise offers for additional features. Write these precisely down, don't rely on phone conversations (you: "that's an additional 4

hours of work", client: "ok"... 4 months later, client "what was that again??? why am i supposed to pay for this").

Now of course, one way you keep prices down is by not hiring complete morons that might be initially cheaper than better qualified programmers. This is a shortsighted approach and will fail miserably.

Share Improve this answer

Follow

3

It is the relationship with your customer that will win you additional business. One developer actually stepped forward and halted a project because his sales consultant basically lied to us concerning a solution set. The developer then offered a straight forward, bare bones solution within the same budget and he delivered on time.
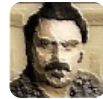
This guy and his team has been consulting at my company now for over 6 years. His integrity and earnest, hard working nature has been an immense advantage, and he has found quality people to work for him as his reputation has grown. His honesty is worth more than any savings I could get by shipping my company's intellectual assets overseas.

answered Oct 20, 2008 at 0:53

David Robbins
10k ● 7 ● 54 ● 83

▲

2

▼

🔖

🕐

"so much competition from offshore/bedroom program teams" - sounds like you guys need to put some time into networking. At the end of the day, people like to do business with people, not with businesses. If you're well known and liked in your client communities, you'll be the front-runners and you'll get a better price from the confidence you have built. And referrals will give you a powerful edge - ask for them.

"our company does not have the money to turn down work" - lot's of companies have this as a start point, but ultimately you have to get past this approach The time you spend on these types of jobs stands in the way of being successful. You need to make decisions about what type of work you want to do (and who the customers will be) and just as importantly what you don't do.

answered Oct 19, 2008 at 23:23

John Mac
2,550 ● 4 ● 24 ● 30

Agreed, particularly on the second point. I follow the online professional freelancing community (they're mostly writers/bloggers and graphic artists, but much still applies to

software development) and I can't count how many times I've seen people saying, "I started turning away low-budget stuff and practically overnight I have more higher-rate work than I can handle." – Dave Sherohman Jun 10, 2009 at 11:53

▲

2

▼

As a consultant, I have personally moved to an hourly-rate-only model for just this reason. I have been burned by too many contracts-gone-wild, and I feel your pain.

In the end, people who only ever go with the lowest priced proposal will be trouble for your company. While you can't be so choosy, such that you don't ever get projects, you definitely want to steer clear of contracts whose supporting management are so agnostic about the actual software development process that they only look at the initial price tag for something. Usually, it's the stuff not in the contract or specification that changes the profit margins and timelines.

It's often in you best interest to be picky up front rather than lowering prices in order to receive a wave of what end up to be trouble-clients anyway. In your case, while there's always the contention between an RFP that is not as concise as a technical requirements specification, initial quotes should be understood as a general estimate based on the level of clarity in the RFP.

And I definitely agree kitsune, that if your company is consistently accepting contracts that don't fit your company's development expertise or bandwidth, all that will result is overhead and bad reputation.

2

This could be seen as quick guide to the above problem.

Proposal Document

1. Start with a proposal document that explains your understanding of the clients needs as best as possible. This can be done in 1-2 pages of writing as a minimum. It can start heading towards requirements, but it should be more casual than that.

Budget Document

2. Now go to Excel and list all the tasks in the project you think you will have to do. Put down the times in days, none bigger than 2 (0.25, 0.5, etc.).

3. Add a column for testing, and make it a percentage of the development time (20-30% is normal)

4. Now add a column for management (project + account) and add a % of time for that (over the previous two columns). 20-40% is normal. (70-30 split pm/am)

5. Set a day rate for your company. You can get more complex and have different rates for different user functions, but as a minimum set a rate that will mean that you will have a good margin whatever the work is being performed.

6. Work out what the value is for the total days that has been recorded so far. Then add a contingency amount on top of this (for fixed price work) 10-20% is normal here, but can change based on experience with the client and the amount of change you are used to with them.

7. At this point you can discount the total amount, which is better than lowering any other part of this document, as it will show the client that you are not magically making jobs go faster, rather you are removing some of your margin. They should therefore not expect you to reduce the timeframe of the project.

**Important - Analyze your client's budget and business when developing a costing. There is no point in you delivering a costing designed for a huge enterprise to your mate who wants an application done cost effectively. Likewise, make sure that you charge correctly to an organisation that will be used to high end freelancer rates.

Think like a business analyst. Not only will it help you make the client happy when they see your costs, but it will probably give you a greater insight into their business.

If they are going to make money by using you, then you are probably onto a winner. If you can't ask directly how much money they have to spend with you, work it out by asking how many customers they have, what they charge, how many employees they have, etc. You should then be able to work out if what you are proposing is going to be profitable for them.**

8. Go back to your proposal document, and add a table with sections for design, development, management, etc... You could show the client your costing sheet in certain circumstances, but it is better to avoid complexity in most cases. It's there as a backup though, and you didn't just cough up a number.

Timeline Document

9. Take the list of design and development tasks from the budget, and put it into a new sheet (or Project if you're posh like me). Put in the start and end dates of each section adding roughly 30-50% extra on top.

10. Add some graphical representation of the days as blocks in Excel or use a [Gantt template like this one](#).

11. Go back to your proposal document and add key milestones from the timing document.

Proposal stage completed

12. Send or present your proposal to your prospective client. Hopefully they are excited with what you have proposed and happy to proceed to the next stage,

full requirements gathering. In subsequent projects with the same client you may be able to go straight to requirements.

Requirements stage

13. List each requirement against a separate id, either 1,2,4,5 or 1.1,1.2,1.3. It doesn't really matter, but the second one can help with large lists.

14. There are some tests for requirements and you can try to follow these, but sometimes they don't apply (some requirements might be design led for example). Some of these are: Is the requirement testable, is it singular, is it clear? I'll try to find a link to this somewhere.

Share Improve this answer

Follow

edited Aug 1, 2016 at 22:20

This is my developer standpoint:

1

- Version control best practices: Keep the trunk clean, don't commit code that doesn't compile, and commit frequently

- Continuous integration

- Unit testing (with code coverage)

- Automatic deployment on test servers

- Automatic packaging of the application

- Automate as much as you can :)

Also, hire good developers, and treat them well :)

Share   Improve this answer

Follow

edited Aug 1, 2016 at 22:12

Peter Mortensen
31.6k ● 22 ● 109 ● 133

answered Oct 15, 2008 at 12:59

mathieu
31.2k ● 4 ● 65 ● 91

Good list, but they really have nothing to do with agile (or at least aren't agile specific). Everything you list is just good practice no matter what your actual development lifecycle methodology is (agile, srcum, xp, waterfall, etc.) – Scott Dorman Oct 15, 2008 at 13:38

woops, my bad. deleted this part :) – mathieu Oct 15, 2008 at 17:46

Sell fixed price, fixed scope work, not hourly work. That mitigates your customer's risk of an over-run (you absorb all the risk, but you're doing that anyway), and frames the project in terms of the value of the software, not the quality of the effort going into it.

0

Share   Improve this answer

Follow

answered Oct 20, 2008 at 0:56

Tim Howland
7,970 ● 4 ● 29 ● 46

We are trying to build a product and reuse existing experience. Comparing to UK in Ukraine (where I work) salaries are lower, but still 4-5 times higher than in India.

So far, the best result is to get two new clients, which need a similar solution, so we can offer better pricing and we are more confident in our estimate.

BTW, I checked out whywaitdigital.com, and it seems that we have a product which your clients might need. We do portals - editorial, B2C, geo-enabled product catalogs and we use ASP.NET MVC also. You can find contact information on our website, www.socialtalents.com.

Share   Improve this answer

Follow

edited Aug 1, 2016 at 22:16

Peter Mortensen
31.6k ● 22 ● 109 ● 133

answered Jan 21, 2010 at 2:07

st78
8,296 ● 12 ● 55 ● 68

Thanks for your comments, but this was for a previous company, where I was not in charge of as much. At the current company all pricing and technical specification contracts go through me, and we haven't had any problems since then. – Chris Barry Jan 21, 2010 at 17:57