How to react when the client's response is negative on delivery? [closed]

Asked 16 years ago Modified 7 years, 8 months ago Viewed 2k times

15

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 10 years ago.

Improve this question

I am a junior programmer. Since my supervisor told me to sit in with the client, I joined. I saw the unsatisfied face of the client despite the successful (from my programmer's perspective) delivery of the project!

Client: You could have included this!
Us: Was not in the specification!
Client: Common Sense!

As a programmer, how do you respond in this situation?

client project-management feedback

Share

Improve this question

Follow

edited Apr 6, 2017 at 9:45

Brian Tompsett - 汤莱恩

5,875 ● 72 ● 61 ● 133

asked Nov 24, 2008 at 19:06

Saj

18.7k ● 12 ● 53 ● 77

This completely depends on the feature. Is it really common sense? How are we to judge? – Jonta Nov 24, 2008 at 19:09

Was there a question here? Is the question how to react when a client does that? – Paul Sonier Nov 24, 2008 at 19:12

@Jonta : Exactly! A lot of things you might wanna add to make it better. It's a software, you can always make it better in every aspect. Well.. That's what I think.. @McWafflestix: Yes, how do we react? What can you tell them? How would you tell them? – Saj Nov 24, 2008 at 19:15

My reaction is simply to state that this may be included in a future release now that we are aware of it, someone will be in touch over when it will be delivered. After all, it isn't your job to determine what work gets done next, is it? – JB King Nov 24, 2008 at 19:35

Excellent question. Strikes at the heart of the industry. Upvoted. I gave an answer below. – Mike Dunlavey Nov 24, 2008 at 20:19

21 Answers

Sorted by: Highest score (default)

20

What you should do to avoid this situation:

Explicitly spec out what will be included and what will not be included.

The problem probably comes down to the unspecified parts of the spec:

- The client thinks that unspecified stuff should be in, i.e. it was implied.
- The developer thinks that unspecified stuff should not be in.

For future specs that you have, you should have a catch all statement, that explicitly states that if something is not specified in this document, it can be done after the original specification is done at an additional cost.

What you should do in the current situation:

Other than learning from your experiences, you should come to some compromise with the client.

Example: I will do this feature that you feel is common sense, but for all future additions/changes it will have to be spec'ed out explicitly.

I.e. you will have to do a little more work, but it is worth it in return for the catch all explicitly spec'ed agreement your client will enter into.

Bad spec?

Was it necessarily a bad spec? No.

It is impossible to mention everything your clients may expect, so it is critical to have this catch all statement mentioned above stated clearly and explicitly in your spec/contract.

Other ways to reduce the problem:

- Involve the client early, show them early prototypes. Even if they don't demand it.
- Try not to sell the client an end product, but more of a service for working on his product.
- Consider an agile development model or something similar so that tasks are well defined, small, paid for, and indisputable.

Share Improve this answer Follow

edited Nov 24, 2008 at 19:26

answered Nov 24, 2008 at 19:08

Brian R. Bondy

347k ● 126 ● 602 ● 640

But how would you react to that? What would you tell them? How would you tell them? – Saj Nov 24, 2008 at 19:10

3 Upvote for explaining that you do the work in exchange for getting a new signed-off spec with the desired catch-all. IE: the new work isn't free. The client pays you by giving up the ability to come back in 6 months with another "common sense" feature. – Joel Coehoorn Nov 24, 2008 at 19:17

@Brian: The project manager was saying something like that in a very polite way. But my eyes were fixed on client's face and I was thinking would he come back to us for future project.. – Saj Nov 24, 2008 at 19:18

1 Projects literally will go on forever, so it is important for this client to learn that everything must be spec'ed, or else it's time to find a different client. – Brian R. Bondy Nov 24, 2008 at 19:19

14

This would be one of many reasons why I switched to an Agile development philosophy. The only way, in my opinion, to successfully avoid this scenario is to either be omniscient or involve the customer heavily and release early/release often to get feedback as soon as possible. That way you can develop the software the customer really wants, not the software the customer tells you they want.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:12

tvanfosson

532k ● 102 ● 699 ● 798

Agreed. I'm absolutely amazed that the client saw the product for the very first time at delivery! – James Eichele Nov 24, 2008 at 19:18

1 You still can have this speech with Agile... you get it just more rapidly ;) – Patrick Desjardins Nov 24, 2008 at 19:51

Approached agilely, it's more likely to be "Oh. I guess we forgot ..., let's do ... in the next iteration." – tvanfosson Nov 24, 2008 at 21:49

You know those "special" type of clients, those who'd go like- "How could you forget? Didn't you take it seriously?" – Saj Nov 25, 2008 at 12:29

14

Client: You could have included this!

Us: Was not in the specification!

Client: Common Sense!!

Us: We do not attempt to go beyond what the client has specified - we follow the specification. It's as important to NOT implement features not specified as it is to implement features specified. We will never second guess our customers, who value the fact that they can completely depend on us to correctly and completely implement the specification on time and under budget.

As others very rightly point out, the situation is almost always more complex than the simple exchange I've described above.

However, the above is valid if the implementer has a specification with the customer's signature on it which essentially implements an agreement that says "once the software provably implements all the features in the spec then it is considered complete", and anything additional is outside the specification and therefore outside the contract.

The contract itself may have some input here as well - if you don't have a signed contract than it doesn't matter what's in the spec - everything so far has been done on a handshake, and the entire deal (including payment) can go down the toilet based on any dissatisfaction on either side.

But if you have a contract and a specification, and the customer has seen and signed both, then they have no wriggle room to ask you to go further.

Now, as to the question of whether you should implement it:

AWESOME! You delivered a product and they only had *one* complaint. Implement the feature, call it a 'freebie (make sure they understand you're working outside the spec and contract and explicitly send them a bill for the work with the discount shown in dollars) and have them sign off on the project as a whole.

It will explicitly demonstrate that the project is ended, that you went above and beyond the call of duty, and that any further 'surprises' are outside the contract/spec, which gives you a nice layer of protection beyond what you already (ostensibly) have.

If it's a UI issue, then you're in murkier water.

Does the spec adequately describe the UI? Does it have mockups? I wouldn't fault a customer for this complaint

about the UI if the spec did not very closely describe the layout, usage, and include mockups.

Either way, I think you can understand the customer's position - if they haven't played with UI mockups, then they're going to be disappointed with the result regardless - there's no way, psychologically speaking, that you and your customer could have possibly had the same idea in mind (nevermind the fact that common sense isn't!).

Quite frankly if this is the first time the customer has thought about checking out the UI before the work is finished, then it's at least partially your fault for not explaining good UI design processes to them. This is a key feature for their app, and it's very tightly coupled to what they've imagined - no one can be satisfied in such a situation unless they've 'grown' their internal representation over time to match what the reality is.

This disconnect is solved only through frequent user and customer testing, which is obviously missing. This is a problem regarding client education and communication, not whether the specification was met or not.

-Adam

edited Nov 24, 2008 at 20:22

answered Nov 24, 2008 at 19:24

Adam Davis
93.5k • 60 • 271 • 333

Problem with is that 90% of the time the spec was produced by the company doing the programming after picking the brains of the client. So it's their fault the spec wasn't good, not the clients. At least the client will see it that way.
– Paul Tomblin Nov 24, 2008 at 19:37

That may be the case, but if the client signed off on the spec, then they agreed that the final product would be considered complete once it provably met the written, signed off, specification. So even though the non-inclusion of the feature was not their fault, they can't blame the implementer
– Adam Davis Nov 24, 2008 at 19:40

1 But, if the customer never signed off on the spec, then all bets are off - the customer does have a valid complaint.
– Adam Davis Nov 24, 2008 at 19:41

Client just realized that their design (UI) was not good enough when it was brought into reality. They reacted like: "well, we didn't know it would look like this in reality! You were doing it..you could have changed the look and feel! We should have sent someone down here time to time".
EXACTLY. – Saj Nov 24, 2008 at 20:03

@ZiG - That is a hard one to solve. Non-functional UI mockups can help a lot with UI design, you can iterate each day with the client for a week and get the UI right before you start. – Tom Leys Nov 24, 2008 at 20:12

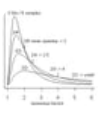
10

- Expect last minute changes of scope - they always happen, so be ready.
- Review progress frequently with client - to minimize surprises.
- Contract: Functional Spec, plus Time & Materials with initial cap (so client feels control). Then when changes come along, re-negotiate the cap if necessary.
- **Never** say they can't have what they want. They can get that answer for free!
- **Always** give them a little more than they asked for, so they know you've got a positive attitude.
- Relate to the client as being on the same team with them. Don't accept being legalistically painted as an adversary.
- They may think of contractors as not loyal, compared to employees. Show them you're as dedicated to their success as their employees are, and you'll go the extra mile.

Share Improve this answer Follow

edited Nov 24, 2008 at 20:16

answered Nov 24, 2008 at 19:58



Mike Dunlavey

40.6k ● 15 ● 94 ● 138

4

Classic case...

There's not definite answer to this one, but it all turns around communication. There should have been preventive measures put in place (like weekly reviews or something like that).

For sure, you can't redo the whole thing for free.

Two ways: Or to tell them to ** off or you deal with it.

If you choose to deal:

- First, empathize, respect the client.
- Have a look at what can easily be changed.
- Have a look at the contracts.
- Maybe create a new agreement.
- Don't do too much.
- Make them see the progress and the work it takes.
- Find workarounds for the missing features (maybe using other great features, or available tools.)

Use your common sense, it is so common, its not even funny.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:16

Loki

30.8k ● 9 ● 51 ● 62

This is one of the many drawbacks of a fixed bid arrangement. Any time business needs or priorities change, or there is even a simple misunderstanding, it results in anything from an awkward situation like this to calling lawyers in. If you have an arrangement where you get paid for development time, you can always react to any change and get paid for whatever time it takes to make that change. Also, having a by-the-hour arrangement does not preclude having a plan or making an estimate.

Once you are in a fixed bid pickle, though, your options are: 1) Do it at an additional cost. 2) Do it free. 3) Don't do it.

Option 3 is the worst, and Option 1 is the best. If you have a good trusting relationship and decent communication with the client, it's usually easy to arrive at Option 1. If the relationship is bad, then you've got bigger problems. At that point, just try to avoid laywers.

A final point - any project that has something known as "The Delivery Date" inevitably runs into the problem described. Projects with said date usually involve retreating to a cave for several months to develop in hiding followed by an unleashing of the product all at once in front of the stakeholders. This is abrupt and leaves plenty of time for client expectations and the actual product to drift apart. If, instead, you show intermediate versions of the product and gather feedback every few weeks, two things happen. First, you get better

feedback, minimize misunderstandings, and make a better product. Second, there is no single point in time on which a massive amount of expectation is laid. The potential difference between what the client is imagining and what actually exists is much smaller. No surprises.

Good luck.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:26

Greg Smalter

6,689 ● 9 ● 46 ● 63

2

"how do you react?"

Question 1 - do you want to continue this relationship with this customer? Seriously. If they are going to claim that unspecified features are "common sense," this may not be a good relationship to maintain or enhance.

If you want to disengage, then that's easy. Ask for them to highlight each part of the specification that you failed to comply with and play that game. Get specific test criteria for each missing feature. Pull Teeth. Be confrontational in determining what's missing. Don't ask why. Just ask for all the details up front. It's slow and unpleasant. But you don't want them anyway.

If you want to engage, well, you're going to have to change the relationship. Currently, you have a Passive Aggressive Customer. They won't say what they want, but they will say what they don't want.

This may be a habit with them; this may be how they win concessions. Or this just may be sloppy specification on their part.

If you want the relationship, your reaction has two parts.

1. Short-term. Get something they're happy with. They have to identify specific changes. You have to score each change with a "cost to do" and "fit with specification".
    - Some things are cheap and a good fit. Do those.
    - Some things are cheap to do, but a bad fit with the specification. Think twice about enabling a bad specification to lead to rework. In a sense, you purchased the specification from them; you may need to raise your standards, also.
    - The expensive things which (sadly) fit the specification are a problem. You're in trouble with these, and pretty much have to do them.
    - The expensive things which don't fit well with the specification are lessons learned for everyone. Detail a plan for these, including specification rewrites and approvals.
2. Long-term. Make sure they you're not PA'd again. Review early and often, use Agile techniques. Communicate more, prototype more, release more.

answered Nov 24, 2008 at 19:40

S.Lott

391k ● 82 ● 517 ● 788

2

Well, it was not successfully delivered. Somewhere along the line there was miscommunication. Without knowing the specifics I would suggest this is not a developer injected problem and this is probably not to be blamed on the customer - the requirements gathering task was insufficient. This is a classic example of what happens when the software side does not have domain experts or the requirements discovery process doesn't do all that it could...

If it was me I would correct the problem and figure out how to avoid similar issue in the future.

How you handle this can very well determine the future of this contract/business with the client. Taking responsibility and correcting the issue is a huge opportunity for your company.

EDIT: This is a good time to evaluate how this happened to help correct it. Some companies choose to totally revamp everything they do which is a mistake I think. So is ignoring it. Blaming people for the problem is also a mistake.

It is a good time to walk through how this happened, what the process is, and maybe how it could have been caught. I would not make huge rule changes or process changes - but coming up with guidelines for future work is a great thing. Your company had a clear lesson about a shortcoming. Losing the opportunity to correct this

problem and to correct your process would be a waste of a good chance.

Share Improve this answer Follow

edited Nov 24, 2008 at 19:43

answered Nov 24, 2008 at 19:29

Tim

20.4k ● 24 ● 122 ● 219

Yes, there was a meeting in corporate level after that! :)
– Saj Nov 24, 2008 at 19:31

2

ZiG, I've had to deal with this problem on several occasions at my current place of Employment. My group (3 developers) tries to approach things in an Agile manner. We're used to getting mid-stream and even last-second requests (which we then treat on a case-by-case basis).

However, we make it clear that resources (particularly time) are limited and if it's not in the spec we can't make promises. If it's judged important and it can't fit into the current release, we generally plan a followup release. If it isn't important, it goes on a list.

One thing I've found is that you can get users to agree to Spec S at Time T. However at Time T + N, getting them to remember they agreed to Spec S, or getting them to acknowledge that they did so (with the documentation

you've been keeping, I hope!) can be trickier than it should be.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:57

peacedog

1,415 • 20 • 32

2

Speaking to the OP's subject and question:

If you are an employed programmer, then I would hope that other resources are in the meeting with you. Possibly "higher ups" in the organization.

If this is the case, then your job is to answer DIRECT questions, and to keep your emotions in check. Yes, you may feel injured because they don't love your code, but showing any emotion with bosses present is not a good thing. Rather, try and look neutral and let the others handle the session.

Now, if they "hang you out to dry", then I would recommend the following questions:

a) "OK. I see. Why exactly to you feel this is common sense to include this feature? I'd like to discover why we didn't include it." (force them to explain their thought process. Common sense to one person is rarely common sense to anyone else.)

b) "Well, I'm sure we could include that in the next release. I'll leave it up to XXX (the bosses) to come to a

mutually agreeable approach" (i.e. don't talk cost or freebies with bosses present. EVER.)

Again, this assumes you are a programmer WORKING for a company that delivered the product. Now, if are more than that - i.e. you ARE one of the higher ups, then many of the suggestions here are excellent.

However, if you are the higher-up or are a consultant programmer, then first and formost

a) Apologize for the process that did not catch this requirement. Promise to work with the client to prevent this from recurring.

Then on to the other strategies. It really doesn't matter if you charge for the fix or not - the apology is the most important action to the client. Again, it bears repeating - you are not apologizing for the missed feature. You are apologizing for the faulty design process that let it slip. Clients are usually pretty accommodating when you start this way and then seek a solution.

Cheers,

-Richard

answered Nov 24, 2008 at 21:57

Huntrods

2,561 ● 3 ● 22 ● 29

1

Use SCRUM like approaches to avoid this deathtrap: involve the client in the dev process early, frequently and in informal, restricted commitees -> risk reduction and improved agility.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:15

Yann Semet

623 ● 2 ● 5 ● 12

1

In terms of your literal question, how to react, the best way is to ignore your ego ("what?! After I worked so hard on this and met the spec?!") and instead focus on some active listening and working to consensus.

Client: You could have included this!

Us: Was not in the specification!

Client: Common Sense!!

Us: I understand that you're not happy that we didn't go beyond the bounds of the specification. Seeing how you feel about this, how can we make you happy? Let's see if there's a process we can create together that will help everyone.

Essentially, you don't want to turn this into a "you said/I said" death match. The only way to resolve those involves lawyers and then nobody wins. If you can agree

that the spec or the process was to fault, work together to fix those.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:42

plinth

49.1k ● 11 ● 83 ● 123

1

This approach actually just worked for me: **wait for the guy who doesn't like your software to leave and be replaced by the guy who does like it**.

Obviously you can't really rely on this, but if you're sure that you did a good job and that your software really will satisfy the business needs of the people who hired you, it does pay to wait it out. Sometimes the client's initial reaction will not be their final one, especially if you can quickly incorporate their concerns.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:43

MusiGenesis

75.2k ● 41 ● 197 ● 338

1

Don't try make the client feel like it is their fault. It might be their fault, but making them feel that way will not produce constructive results, and could just annoy them.

Instead, you should realize that clients only complain about software they use, in most cases because they like it. Nobody complains about software nobody uses. It is inevitable that a client will complain about the software

you deliver, even if you deliver exactly what they ask for. So don't sweat it. Software is never done.

Share Improve this answer

Follow

answered Nov 24, 2008 at 20:01

John Dibling

101k ● 32 ● 191 ● 331

0

Total failure on the part of the person in charge of requirements collection, no doubt about it. Additional failure of the project management to not iterate the deliverable and have check-in meetings with the client.

However, you have a signed-off spec, and what you've delivered matches the spec. So, your company has two choices: write off the cost in the name of business development and make the change for free, or charge them for the change request.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:12

Greg Hurlman

17.8k ● 6 ● 55 ● 87

This is not a "total failure" - some things got into the delivery. But it is an indication of a problem in the requirements/specification gathering step(s). – Tim Nov 24, 2008 at 19:45

The level of failure is related to the missing feature - if something was left out that the client was "obvious", it's much more damaging than a feature that the client feels was just somehow left out. For anyone that's done any amount of consulting work, requirement gathering is the easy part.
– Greg Hurlman Nov 26, 2008 at 18:53

0

If it ain't in the spec, it ain't in the spec. As a developer with no specific domain knowledge, 'common sense' is an irrelevant concept. Different industries work in different ways and one approach might be quite appropriate for a particular domain but completely unacceptable in the other.

Writing good specs is an art-form. IMO, you can either take an agile 'analyst/programmer' approach where you make small iterations or write and maintain a detailed, unambiguous specification. Both are highly skilled tasks, and are still iterative. You still have to evolve the specification.

Either way is not as easy as it sounds and both require the ability to establish a good working relationship with the client.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:14

ConcernedOfTunbridge Wells

66.5k ● 15 ● 148 ● 198

0

You cannot know what your customer think in his head. This situation occur often with client that haven't got any experiences with programming project. What I suggest to you is to simply show him that "common sense" isn't very accurate as answer in engineering (or programming if you prefer).

Show him other example in life that will show him that you cannot build something that aren't written. Example: building a new house, the guy who build the house need a plan with all detail... he won't put optional electric plug because in the living room it's more "common sense" to have some extra...

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:16

Patrick Desjardins

141k ● 89 ● 294 ● 346

0

I had this once. And luckily it wasn't me that created the design because that proved to be the problem.

It is of vital importance that the communcation between your company and the client is as perfect as possible. Be sure you understand each other. Ask questions and let them ask questions. Do not let anything open in the design. This will be the problem point at delivery. And have regular meetings during the project (preferably with a prerelease).

Unfortunately a lot of developers are bad at communciation, and a lot of clients are not aware of their own needs. But if you can minimize the gap, you have found yourself a happy (and returning) customer.

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:18

Toon Krijthe

53.4k ● 38 ● 149 ● 202

0

This is why I/the teams I worked with always used a prototype-style approach, that means:

1. after collecting the requirements, you show the client an early and basic release of the software
2. the client says "**you could have included this**"/"**it's common sense**"
3. you change your design to reflect the client's desiderata
4. iterate from point 1 till the official release

Share Improve this answer

Follow

answered Nov 24, 2008 at 19:50

friol

7,086 ● 4 ● 49 ● 82

And in the final day, some clients (apparently assholes) say "you could have done this!/common sense!!" – Saj Nov 24, 2008 at 19:57

@Zig, lol I can relate to that.. – zamfir Nov 26, 2008 at 7:53

0

You have to start it early on; tell the customer, early and often, that the spec/use-cases/user-stories are a contract which define what will be delivered. in an agile environment there are plenty of chances for the customer to observe some "common sense" feature they want and ask for it, which is one of the advantages of an agile approach, but if you start accepting "common sense" additions at the end, you are preparing yourself for infinite extensions, probably at your expense.

Some customers *expect* this; the more and better you tell them they can't, the easier the eventual arguments will be.

As a junior guy, I realize you can't do this -- yet -- but one of the hard-but-necessary lessons is that sometimes you have to fire a customer.

Share Improve this answer

Follow

answered Nov 24, 2008 at 20:25

Charlie Martin

112k ● 26 ● 196 ● 264

0

You **learn** - everything is learning and nothing is personal. We are experts in our area we know better than customer what he need. And next time for next customer we will suggest all useful features in advance and make him happy and will make him pay more money because we are the experts and we know better.

Share Improve this answer

Follow

answered Nov 24, 2008 at 20:39

Ilya

3,138 ● 4 ● 24 ● 30