What you do as developer with non-bugs client requests, f.e small fix in UI? [closed]

Asked 16 years, 1 month ago Modified 15 years, 7 months ago

Viewed 223 times

▲

2

▼

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 4 years ago.

Improve this question

I am only one person working on project - so I am developer without PM above me. I finished portal, hovewer client from time to time attacks me with request such as "make font bigger" or change margin in css or make button which makes "xxx and yyy". There are simple task, sometimes only for few clicks, but it takes my time and I hate doing such tasks. On the other hand I understand those people, while sometimes small fix helps them a lot in work. What say them on communicators - it's hard to ignore them. Is disabling communicators best solution - but I need it to communicate with my co-workers. What you do in such situations?

client time-management

Share

Improve this question

Follow

asked Oct 31, 2008 at 14:03

pbrodka

1,137 ● 1 ● 12 ● 23

"What say them on communicators" What does this mean? What is a communicator? The last couple of sentences on this question are confusing. – epochwolf Oct 31, 2008 at 14:09

I think it means "What do we think about these communication tools?" (hopefully only email, clients contacting you via IM is not a good thing) – Dave DuPlantis Oct 31, 2008 at 14:19

6 Answers

Sorted by: Highest score (default)

▲

4

▼

✓

Create an established queue where your users can submit requests, in a manner that doesn't disrupt your day-to-day workflow.

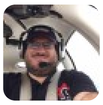
From the sounds of this you are getting requests via a communication channel that you check regularly, you might try to move it off to the side.

Cutting off communication is NEVER a good solution. Also, I would formalize a process and time schedule for when you get to those types of requests. I've found great success with this simple approach.

Share Improve this answer

Follow

answered Oct 31, 2008 at 14:06



Mitchel Sellers

63.1k ● 15 ● 114 ● 174

▲

2

▼

If you're working for yourself, you clients are your single most important reason you're there. They are your business! Thus, it's always good practice to keep them happy.

That being said...

You should always always always have a clearly defined contract when working on any sort of software project for a client. You need to ensure that your deliverables are clearly expressed and defined both to you and to your customer. Once you've got that taken care of you need to also ensure that there is a section that covers "future maintenance requests" and you can then work with your client to ensure expectations are acceptable on both ends of the spectrum and your time spent on them is both accounted for and part of the original plan moving forward.

The fewer open ends, the better.

Afterwards, implementing a system to manage/handle customer requests for each of the projects/websites you've implemented can also be a great help. Tools like FogBugz from one of this sites founders do a great job in handling customer interaction and bug/feature requests. Check it out.

Share Improve this answer Follow

answered Oct 31, 2008 at 14:08

Mat Nadrofsky

8,274 ● 8 ● 51 ● 73

0

Although not a technical "bug", usability by the client is the most important bug to the user. If you want to continue business with the client, the small things need to be worked.

fixing small bugs == client happiness == more work == more $$

Share Improve this answer Follow

answered Oct 31, 2008 at 14:07

Inisheer

20.8k ● 9 ● 54 ● 83

0

Deploy a system for tracking bugs and tracking change requests (at my office we use MKS, which is also used for source integrity). Then when a user has a request, they go into the tracking system and enter the request as the appropriate type. Ideally they should also be able to attach a severity/priority indicator to it so that the outstanding requests can be ranked. You can then go in and see all outstanding requests, and prioritize them. Since they are not being directly sent to you, you won't feel inundated with requests, and the users will find that they can track the status of their requests more easily than by calling you and asking "when will my fix be done?"

For yourself, you can check the list a few times a day and see if there are any high priority issues to work on. Then schedule some time on a regular basis (one day a week, or an hour day, whatever feels reasonable) to work on the lower priority issues.

Share Improve this answer Follow

edited Oct 31, 2008 at 15:01

answered Oct 31, 2008 at 14:13

Elie
13.8k ● 25 ● 78 ● 128

I'm not sure to what extent you meant "create", but for solo developers or those in small groups, I'd definitely recommend using existing packages rather than developing your own. A decent bug/change-tracking system takes a lot of work to maintain. – Dave DuPlantis Oct 31, 2008 at 14:21

I did not mean to say develop your own package, just to incorporate such a package and create a protocol for using it. You're absolutely right - it's a lot of work to maintain, not to mention build. But you do need a custom protocol for use that reflects your work environment. – Elie Oct 31, 2008 at 14:24

@Elie, other words like "deploy" or "implement" convey that meaning better than create. – Esteban Brenes Oct 31, 2008 at 14:59

0

I think you have to consider your ongoing relationship with your customer. If a customer spends a few minutes of your time occasionally you may consider that the cost to you is minimal and the benefits of the contact may outweigh the cost anyway.

If the requests are coming in thick and fast, you maybe need to talk to your customer about an hourly rate for changes or cover them in a chargeable support contract.

Share Improve this answer

Follow

answered Oct 31, 2008 at 18:12

BlackWasp

4,961 ● 2 ● 33 ● 42

0

Do not change your path on each feature request that you get. Collect feature requests for a while, then prioritize the requests, then select the ones that make sense, and then work on the next release.

In my opinion it is good to follow some fixed release schedule: it makes the development process more controllable, improves software quality, and your customers know what to expect.

Share Improve this answer

Follow

answered May 3, 2009 at 19:07

Roy

338 ● 3 ● 8