How can I support the support department better?

Asked 16 years, 2 months ago Modified 16 years ago Viewed 316 times

6

With the best will in the world, whatever software you (and me) write will have some kind of defect in it.

What can I do, as a developer, to make things easier for the support department (first line, through to third line, and development) to diagnose, workaround and fix problems that the user encounters.

Notes

- I'm expecting answers which are predominantly technical in nature, but I expect other answers to exist.
- "Don't release bugs in your software" is a good answer, but I know that already.

language-agnostic

Share
Improve this question
Follow

edited Oct 1, 2008 at 13:59

asked Oct 1, 2008 at 11:45

jamesh
20.1k ● 14 ● 58 ● 96

9 Answers

Sorted by: Highest score (default)

5

- Log as much detail about the environment in which you're executing as possible (probably on startup).
- Give exceptions meaningful names and messages. They may only appear in a stack trace, but that's still incredibly helpful.
- Allocate some time to writing tools for the support team. They will almost certainly have needs beyond either your users or the developers.
- Sit with the support team for half a day to see what kind of thing they're having to do. Watch any repetitive tasks - they may not even consciously notice the repetition any more.
- Meet up with the support team regularly - make sure they never resent you.

Share Improve this answer Follow

answered Oct 1, 2008 at 11:49

Jon Skeet
1.5m ● 889 ● 9.3k ● 9.3k

4

If you have at least a part of your application running on your server, make sure you monitor logs for errors.

When we first implemented daily script which greps for ERROR/Exception/FATAL and sends results per email, I was surprised how many issues (mostly tiny) we haven't noticed before.

This will help in a way, that you notice some problems yourself before they are reported to support team.

Share Improve this answer Follow

answered Oct 1, 2008 at 13:00

Vilmantas Baranauskas
6,726 ● 3 ● 41 ● 52

4

Technical features:

- In the error dialogue for a desktop app, include a clickable button that opens up and email, and attaches the stacktrace, and log, including system properties.
- On an error screen in a webapp, report a timestamp including nano-seconds and error code, pid, etc so server logs can be searched.
- Allow log levels to be dynamically changed at runtime. Having to restart your server to do this is a pain.
- Log as much detail about the environment in which you're executing as possible (probably on startup).

Non-technical:

- Provide a known issues section in your documentation. If this is a web page, then this correspond to a triaged bug list from your bug tracker.
- Depending on your audience, expose some kind of interface to your issue tracking.
- Again, depending on audience, provide some forum for the users to help each other.
- Usability solves problems before they are a problem. Sensible, non-scary error messages often allow a user to find the solution to their own problem.

Process:

- watch your logs. For a server side product, regular reviews of logs will be a good early warning sign for impending trouble. Make sure support knows when you think there is trouble ahead.
- allow time to write tools for the support department. These may start off as debugging tools for devs, become a window onto the internal state of the app for support, and even become power tools for future releases.
- allow some time for devs to spend with the support team; listening to customers on a support call, go out on site, etc. Make sure that the devs are not allowed to promise anything. Debrief the dev after doing this - there maybe feature ideas there.

- where appropriate provide user training. An impedence mismatch can cause the user to perceive problems with the software, rather than the user's mental model of the software.

Share Improve this answer Follow

edited Dec 19, 2008 at 11:46

community wiki
3 revs
jamesh

2

Make sure your application can be deployed with automatic updates. One of the headaches of a support group is upgrading customers to the latest and greatest so that they can take advantage of bug fixes, new features, etc. If the upgrade process is seamless, stress can be relieved from the support group.

Share Improve this answer Follow

answered Oct 1, 2008 at 11:52
YeahStu
4,052 ● 5 ● 49 ● 69

This is not great practice in my experience. For corporate clients, they **hate** auto-update, as it messes with security practices, testing, certification, etc etc. – jamesh Oct 1, 2008 at 17:07

Similar to a combination of jamesh's answers, we do this for web apps

1

- Supply a "report a bug" link so that users can report bugs even when they don't generate error screens.
- That link opens up a small dialog which in turn submits via Ajax to a processor on the server.
- The processor associates the submission to the script being reported on and its PID, so that we can find the right log files (we organize ours by script/pid), and then sends e-mail to our bug tracking system.

Share Improve this answer Follow

edited Oct 1, 2008 at 13:15

answered Oct 1, 2008 at 11:50

Adam Bellaire
110k ● 19 ● 152 ● 165

1

Provide a know issues document Give training on the application so they know how it should work Provide simple concise log lines that they will understand or create error codes with a corresponding document that describes the error

Share Improve this answer Follow

answered Oct 1, 2008 at 14:05

WACM161
1,033 ● 2 ● 9 ● 20

Some thoughts:

1

- Do your best to validate user input immediately.
- Check for errors or exceptions as early and as often as possible. It's easier to trace and fix a problem just after it occurs, before it generates "ricochet" effects.
- Whenever possible, describe how to *correct* the problem in your error message. The user isn't interested in what went wrong, only how to continue working:

  > **BAD:** Floating-point exception in vogon.c, line 42
  > **BETTER:** Please enter a dollar amount greater than 0.

- If you can't suggest a correction for the problem, tell the user what to do (or *not* to do) before calling tech support, such as: "Click Help->About to find the version/license number," or "Please leave this error message on the screen."
- Talk to your support staff. Ask about common problems and pet peeves. Have *them* answer this question!
- If you have a web site with a support section, provide a hyperlink or URL in the error message.
- Indicate whether the error is due to a temporary or permanent condition, so the user will know whether to try again.
- Put your cell phone number in every error message, and identify yourself as the developer.

Ok, the last item probably isn't practical, but wouldn't it encourage better coding practices?

Share Improve this answer Follow

answered Oct 26, 2008 at 7:22

Adam Liss
48.3k ● 13 ● 113 ● 152

1

- Provide a mechanism for capturing what the user was doing when the problem happened, a logging or tracing capability that can help provide you and your colleagues with data (what exception was thrown, stack traces, program state, what the user had been doing, etc.) so that you can recreate the issue.
- If you don't already incorporate developer automated testing in your product development, consider doing so.

Share Improve this answer Follow

edited Dec 19, 2008 at 11:37

jamesh
20.1k ● 14 ● 58 ● 96

answered Oct 1, 2008 at 11:50

itsmatt
31.4k ● 11 ● 102 ● 165

1

Have a mindset for improving things. Whenever you fix something, ask:

- How can I avoid a similar problem in the future?

Then try to find a way of solving *that* problem.

Share Improve this answer

Follow

answered Dec 19, 2008 at 11:41

Ola Eldøy

5,938 ● 7 ● 52 ● 84

The five whys addresses a similar attitude. – jamesh Dec 19, 2008 at 13:59