THOMAS SLUYTER AND
ROLAND VAN MAARSCHALKERWEERD

when disaster strikes

CAILIN AND ROLAND DISCUSS CRISIS MANAGEMENT

In daily life, Thomas (a.k.a. Cailin) is part of a small, yet highly flexible, UNIX support department at ING Bank in the Netherlands. He took his first steps as a junior UNIX sysadmin in the year 2000. Thomas part-times as an Apple Macintosh evangelist and as board member of the J-Pop Foundation.

■ *tsluyter@xs4all.nl*

As a senior UNIX sysadmin, Roland van Maarschalkerweerd delivered input for this article, having over 20 years of experience dealing with all kinds of OSes, but over the last decade specializing in (Sun) UNIX. Besides working as a colleague of Thomas, Roland mainly enjoys bringing up four kids, providing an extra dimension in crisis-management experience.

■ *joostb8@planet.nl*

WE'VE ALL EXPERIENCED THAT SINK-ing feeling: blurry-eyed and not halfway through your first cup of coffee, you're startled by the phone. Something's gone horribly wrong and your customers demand your immediate attention!

From then on things usually only get worse. Everybody's working on the same problem. Nobody keeps track of who's doing what. The problem has more depth to it than you ever imagined, and your customers keep on calling back for updates. It doesn't matter whether the company is small or large: we've all been there sometime.

The last time we encountered such an incident at our company wasn't too long ago; it wasn't a pretty sight and actually went pretty much as described above. During the final analysis, our manager requested that we produce a small checklist to prevent us from making the same mistakes again. The small checklist finally grew into this article, which we thought might be useful for other system administrators.

Before we begin, we'd like to mention that this article was written with our current employer in mind: large support departments, multiple tiers of management, a few hundred servers, and an organization styled after ITIL (the IT Infrastructure Library). But most of the principles described here also apply to smaller departments and companies, albeit in a more streamlined form. Meetings will not be as formal, troubleshooting will be more supple, and communication lines between you and the customer will be shorter.

We have been told that ITIL is mostly a European phenomenon and that it is still relatively unknown in the US and Asia. The Web site of the British Office of Government Commerce (http://www.itil.co.uk) describes ITIL as follows:

> ITIL . . . is the most widely accepted approach to IT Service Management in the world. ITIL provides a cohesive set of best practice, drawn from the public and private sectors internationally.
>
> ITIL is . . . supported by publications, qualifications and an international user group. ITIL is intended to assist organizations to develop a framework for IT Service Management.

Some readers may find our recommendations to be strict, while others might find them completely over the top. It is, of course, up to your discretion how you deal with crises.

A Method to the Madness

The following paragraphs outline the phases one should go through when managing a crisis. The way we see things, phases 1 through 3 and phase 11 are all parts of normal day-to-day operations. All steps in between—4 through 10—are to be taken by the specially formed crisis team.

1. A fault is detected
2. First analysis
3. First crisis meeting
4. Deciding on a course of action
5. Assigning tasks
6. Troubleshooting
7. Second crisis meeting
8. Fixing the problems
9. Verification of functionality
10. Final analysis
11. Aftercare

1. A FAULT IS DETECTED

"Oh, the humanity!"

—Reporter at the crash of the Hindenburg

It really doesn't matter how this happens, but this is naturally the beginning. Either you notice something while v-grepping through a log file, a customer calls you, or some alarm bell starts going off in your monitoring software. The end result will be the same: something has gone wrong and people complain about it.

In most cases, the occurrence will simply continue through the normal incident process, since the situation is not on a grand scale. But every so often something very important breaks, and that's when this procedure kicks in.

2. FIRST ANALYSIS

"Elementary, my dear Watson."

—The famous (yet imaginary) detective Sherlock Holmes

To be sure of the scale of the situation, you'll have to make a quick inventory:

- Gather all incident cases, phone calls, and other reports related to this particular problem.
- Make a tally of the number of servers, applications, and customers affected by the problem.
- Assess the impact on each individual customer and on the company as a whole.
- Make a quick list of colleagues who are knowledgeable on the subject at hand.

Once you have collected all of this information, you will be able to provide your management with a clear picture of the current situation. It will also form the basis for the crisis meeting, which we will discuss next.

This phase underlines the absolute need for detailed and exhaustive documentation of your systems and applications. Things will go so much smoother if you have all of the required details available. If you already have things like Disaster Recovery Plans lying around, gather them now. If you don't have any centralized documentation yet, we'd recommend that you start right now to build a CMDB, lists of contacts, and so-called build documents describing each server.

3. FIRST CRISIS MEETING

"Emergency family meeting!"

—Cheaper by the Dozen

Now the time has come to determine how to tackle the problem at hand. In order to do this in an orderly fashion you will need to have a small crisis meeting.

Make sure that you have a whiteboard handy, so you can make a list of all of the detected defects. Later on this will make it easier to keep track of progress, with the added benefit that the rest of your department won't have to disturb you for updates.

Gather the following people:

- The operational supervisor or, your organization has no ops supervisor, the department head
- The resident ITIL problem manager
- The current on-call team member, meaning the one who took all the calls and who gathered the information in phase 2
- One or two people who are especially knowledgeable on the resources involved in the problem at hand (you'll select them from the short list you made)

During this meeting the on-call team member brings everybody up to speed. The supervisor is present in order to prepare for any escalation from above, while the problem manager needs to be able to inform the rest of your company through the ITIL problem process. Of course, it is clear why all of the other people are invited.

4. DECIDING ON A COURSE OF ACTION

One of the goals of the first crisis meeting is to determine a course of action. You will need to set out a clear list of things that will be checked and of actions that will need to be taken to prevent confusion along the way.

It is possible that your department already has documents such as a Disaster Recovery Plan or notes from a previous comparable crisis that describe how to treat your current situation. If you do, follow them to the letter. If you do not have these documents, you will need to continue with the rest of our procedure.

5. ASSIGNING TASKS

Once a clear list of actions and checks has been created, you will have to assign tasks to a number of people. We have determined a number of standard roles:

- One or more troubleshooters. These people perform the grunt work by going over each check or action on the list.
- One spokesperson who takes care of communications with your customers, management, and the ITIL coordinators. This person also keeps the problem record up-to-date. Basically, he's there to keep everybody out of the troubleshooters' hair, so they can do their work uninterrupted.

It is imperative that the spokesperson not be involved with any troubleshooting whatsoever. Should the need arise for the spokesperson to get involved, then somebody else should assume the role of spokesperson in his or her place. This will ensure that lines of communication don't get muddled and that the real work can continue.

6. TROUBLESHOOTING

In this phase the designated troubleshooters go over the list of possible checks determined in phase 4. The results for each check need to be recorded, of course.

It might be that they find some obvious mistakes that may have led to the situation at hand. We suggest that you refrain from fixing any of these, unless they are really minor. The point is that it would be wiser to save these errors for the second crisis meeting.

This might seem counterintuitive, but it could be that these errors aren't related to the fault or that fixing them might lead to other problems. This is why it's wiser to discuss these findings first.

7. SECOND CRISIS MEETING

Once the troubleshooters have gathered all of their data, the crisis team can enter a second meeting.

At this point it is not necessary to have either the supervisor or the problem manager present. The spokesperson and the troubleshooters (perhaps assisted by a specialist who's not on the crisis team) will decide on the new course of action.

Hopefully, you have found a number of bugs that are related to the fault. If you haven't, loop back to step 4 to decide on new things to check. If you did, now is the time to decide how to go about fixing things and in which order to tackle them.

Make a list of fixable errors and glance over possible corrections. Don't go into too much detail, since that will take up too much time. Leave the details to the person who's going to fix that particular item. Assign each item on the list to one of the troubleshooters, and decide in which order they should be fixed.

Then start thinking about plan B. Yes, it's true that you have already invested a lot of time in troubleshooting your problems, but it might be that you will not be able to fix the problems in time. So decide on a time limit, if one hasn't been determined for you, and start thinking worst-case scenario: "What if we don't make it? How are we going to make sure people can do their work anyway?"

8. FIXING THE PROBLEMS

Obviously, you'll now tackle each error, one by one. Make sure that you make note of all of the changes that are made. Once more (I'm starting to feel like the schoolteacher from *The Wall*), don't be tempted to do anything you shouldn't be doing, such as fixing other faults you've detected. And absolutely do not use the downtime as a convenient window for performing that upgrade you'd been planning on doing for a while.

9. VERIFICATION OF FUNCTIONALITY

Once you've gone over the list of errors and have fixed everything, verify that peace has been brought to the land, so to speak. Also, verify that your customers can work again and that they experience no more inconvenience. Strike every fixed item from the whiteboard, so your colleagues are in the know.

If you find that there are still some problems left, or that your fixes broke something else, add them to the board and loop back to phase 3.

10. FINAL ANALYSIS

> "Analysis not possible . . . We are lost in the universe of Olympus."
>
> —*Shirka, the board computer, from Ulysses31*

Naturally, your customers will want some explanation of all of the problems you caused them (so to speak). So gather all the people involved with the crisis team and hold one final meeting. Go over all the things you've discovered and make a neat list. Cover how

each error was created and its repercussions. You may also want to explain how you'll prevent these errors from happening again in the future.

What you do with this list depends entirely on the demands made by your organization. It could be that all your customers want is a simple email, while ITIL-reliant organizations may require a full-blown postmortem.

11. AFTERCARE

> "I don't think any problem is solved unless, at the end of the day, you've turned it into a non-issue. I would say you're not doing your job properly if it's possible to have the same crisis twice."
>
> —Salvaico, Sysadmintalk.com forum member

Even after the postmortem, you may need to take care of a few things. Maybe you've discovered that the server in question is underpowered or that the faults experienced were fixed in a newer version of the software involved. Discoveries like these warrant starting a new project. Or maybe you've found that your monitoring is lacking when it comes to the resource(s) that failed. This, of course, will lead to an internal project for your department.

All in all, aftercare covers all of the activities required to make sure that such a crisis never occurs again. If you cannot prevent such a crisis from happening again, you should document it painstakingly, so that it can be solved quickly in the future.

Final Thoughts

We sincerely hope that our article has provided you with some valuable tips and ideas. Managing crises is hard and confusing work, and it's always a good idea to take a structured approach. A clear and level head will be the biggest help you can have.