SysAid

TIPS

To Help You Improve Problem Management

by Stuart Rance

Many IT organizations are very good at managing incidents. The trouble is people are so busy dealing with incidents that they don't make the time they need to stand back and work out how to prevent them happening in the first place.

Every organization I have worked with does attempt some problem management. But this is often limited to work carried out to help find the root cause of major incidents. It tends to take place during the incident itself and has limited impact once the incident has been resolved. This approach to problem management doesn't do much to deliver long term value to the organization.

A good problem management process can provide enormous value, by reducing how often your customers experience IT service failure and by limiting the impact of failures when they do occur.

Why Do You Need Problem Management?

Just imagine how your customer would react to the following conversation.

	IT: “You logged 500 incidents this week, and we resolved every one of them within the agreed timescales.”
	Customer: “That’s great, I’m glad you’re meeting your commitments in that area.”
	IT: “We’re planning to have even more failures next week. Hopefully you’ll log 1,000 incidents and we’ll really be able to show you how good we are at incident resolution.”
	Customer: !!&***!!$?

IT organizations invest a lot of resources in incident management. But however good at incident management you are, successful incident management on its own can never be enough for your customers. Every IT incident has a negative business impact, however quickly you resolve it. And even if you resolve incidents very quickly indeed, your customers would still prefer it if they hadn’t happened in the first place.

You need to understand that what your customers value isn’t incident resolution. What they value is the ability to conduct their business without being affected by incidents. And that is where problem management comes in.

Successful incident management on its own can never be enough for your customers.

What Is the Purpose of Problem Management?

Problem management has two main purposes:

- To eliminate the causes of incidents, so that similar incidents don't happen in the future
- To reduce the impact of future incidents that can't be prevented

Between them, these can lead to a major reduction in the impact of incidents on the business, resulting in increased service availability, improved customer experience and customer satisfaction, and reduced cost to the customer's business. The reduction in the number of incidents, and improvement in the ability to handle future incidents, can also result in the service provider needing to put a significantly reduced effort into incident management.

This can result in a reduction in the cost of IT, and can also free up technical people who can now focus on proactive activity such as problem management – a genuine win-win situation.

Problem management will never be able to eliminate all of your incidents, but it can make a very big difference if you invest the time and effort needed. This is one area of IT service management where an initial up-front investment can really pay off in the long term.

For some reason organizations find it hard to get started with problem management, so here are some tips to help you on your problem management journey.

TIP#1
Log Your Problems

The first step towards problem management is straightforward. Keep good records. If you don't log problems then you will never be able to manage them.

You should log a problem for every major incident that occurs. Use this to investigate what happened and to make sure that there won't be a repetition.

You should also log problems for recurring incidents, clusters of incidents or, indeed any incident that's a pain, even if it's apparently trivial. Logging problems and then analysing the problems you have logged will help you spot trends and deal with potential issues before they become serious.

You don't have to investigate and resolve every problem just because you logged it. The amount of effort you allocate to a problem will depend on how it is prioritised. When customers log incidents you do have to investigate and resolve every last one, but when you log problems yourself, for your own purposes, it's okay to simply log them without having to investigate and understand them all.

When you log your problems you can match new incidents to problems that have been logged before. This allows you to track how many incidents occur for each problem you identify, and what impact that problem has. This data will help you to prioritise your problems correctly, so that you can focus your problem management work on investigating and resolving the problems that are causing the most business impact.

The priority rating for a problem should depend on:

- How many related incidents there have been
- What business impact these incidents have had
- How recently the incidents occurred (so that a problem that "fixed itself" will eventually drop in priority)
- How much it would cost to investigate and resolve the problem

Many organizations produce a "Top 5" or "Top 10" problem report each month, identifying the problems that they are going to focus on.

One customer that I worked with had a policy that every incident must be associated with a problem or a known error. Service desk staff were trained to match new incidents to existing problems and known errors, and, whenever necessary, to log a new problem. This ensured that the impact of every problem was well understood, and facilitated a very high level of data quality. An organization does need to have a high level of process maturity to be able to adopt such an approach and it is not suitable for every IT organization.

TIP#2

Focus on Workarounds, Not Root Causes

Many people think that the purpose of problem management is to find the root cause of problems. This is a very inward-focused technologist's view of problem management. Finding the root cause of a problem will not deliver any value to you or your customers. It is simply something that we do to help us get to our real goal, eliminating the problem or reducing its impact.

I have been involved in helping to resolve a number of major escalations where technical people had spent weeks trying to understand the root cause of a complex problem, leaving the beleaguered service desk to deal with increasingly frustrated users who were unable to work every time a related incident occurred.

One very effective way of helping these organizations to improve was to get the best technical people to focus on devising and documenting a good workaround and to return to their analysis only after this was in place. On one occasion we managed to reduce the downtime caused by a frequently recurring problem by over 90%, just by devising an effective workaround. Not only did this reduce the cost to the business, but it also reduced the pressure on IT, enabling them to take a more relaxed approach to problem analysis because they were no longer having to deal with a stream of complaints.

To put in place an effective workaround you need to know both what to do and, crucially, when to do it. So a workaround should have two parts.

- **A trigger:** This is a test or set of tests that you can apply to a new incident to see if it matches a problem for which you already have a workaround. This is essential to allow you to identify whether or not the workaround will actually be helpful in this particular case. If you can automate the trigger then that's great, otherwise you need to ensure that your service desk agents understand exactly what to look for when they are logging incidents.

- **Recovery action:** This is a clear description of the steps required to recover from the incident when it occurs. Again, it's really great if the recovery action can be automated. If it requires manual steps then it is essential to make sure that the service desk agents understand the steps and are able to implement them.

Each time the workaround is used, the service desk should be given an opportunity to offer feedback on how effective the trigger was, and how well the recovery action worked. If either of these is not working well, then you should get your technical people to stop investigating the problem for long enough to improve the workaround so that it meets the needs of the users, and of the service desk.

TIP#3

Focus on Improvement Opportunities, Not Allocation of Blame

When you analyse a problem it's really important that you have access to all available information about any incidents related to the problem. If the problem relates to a major incident that had a significant impact on the organization then there may only be a single incident for you to investigate, and it is even more important to understand the exact sequence of events and actions that led to the incident.

If people feel that they may be blamed for an incident then it can be very difficult to find out exactly what happened, as they may be reluctant to disclose things that show them in a negative way. This is why it is really important to hold "blameless post mortems" after major incidents, so that people can openly share what happened. In a blameless post mortem people are encouraged to share information about the timeline of activities that led to an incident. The underlying approach is that individual people are NEVER the root cause of an incident, nobody is blamed or punished. The post mortem may identify a need for training, mentoring or coaching, but this is always done in a positive way.

It can be much easier to hold "blameless post mortems" once an organization buys into the concept that problems don't have a single "root cause". There are always multiple contributory causes. Some things caused the incident to occur, others caused it to last longer than it should have done, or have a larger impact than it should have done. You need to identify all these contributory causes, and then decide which of them you want to deal with. For example, a new release of software may have had a bug that caused some critical data to be corrupted. Analysis of the incident may show the following contributory causes.

- The new code had a timing error that caused the data corruption
- The code used an inappropriate synchronisation technique, but this had been in use for many years and had never caused a problem before
- Testing did not discover the software error
- There was no integrity checking in place to rapidly detect the corrupt data and raise an alert before it caused a significant business impact
- When the incident was reported it was not escalated quickly, as the service desk did not realize how critical this data was to the business
- Technical support people analysing the incident did not have access to the right tools to help them understand the nature of the corruption or fix it

It would be easy to blame the programmer, and say that the fix is to modify the code so that it synchronizes correctly. While this would certainly prevent a repetition of exactly what went wrong in this specific instance, there are many other opportunities to improve here, and these should all be identified and acted on.

TIP#4

Train Staff in Analysis Techniques

The technical staff who analyse your problems probably have high levels of expertise in the appropriate technology, but they also need to know how to analyse problems. This isn't a skill that people just develop all by themselves, it requires some effort to acquire.

There are training courses available in some analysis techniques, but probably the most important thing you can do is provide opportunities for people to be mentored. Get your senior staff to involve junior staff with aspects of their work. This should provide senior staff with some assistance, and also offer junior staff the opportunity to develop their own expertise by observing first-hand how someone with greater experience sets about analysing complex problems. Provide people with opportunities to reach out when they need help with problem analysis, maybe review some problem analysis work after it has been completed to see if there are learning opportunities for the people involved. If you don't currently do much problem management due to shortage of skills or resources, then it may be a good idea to get in a consultant to do some initial problem management work and to mentor your people so that they can take over as problem management starts to be effective.

There are lots of different techniques that you can use to help with problem analysis. My favourite problem analysis techniques are *Timeline Analysis* and *Kepner-Tregoe Problem Solving* and I talk about these below, but you may use other techniques such as Brainstorming, Fault Tree Analysis, or Ishikawa Diagrams.

Timeline Analysis

The most important, and simplest, technique to use is Timeline Analysis, or chronological analysis. It is particularly effective when you are analysing a problem that was caused by a single major incident.

Timeline Analysis is as simple as the name suggests. You find out what happened and put it all into a single timeline, regardless of the source of your information. I find it helpful to use a spreadsheet where I put the date and time in one or two columns, and then use subsequent columns for different sources of data, so that it looks like this:

Date	Time	Interview with A	Event log from system X	Incident record	Log from building management system
20 OCT	10:42				Sudden temperature increase
20 OCT	11:04		Disk error "xxxxx"		
20 OCT	11:22			Incident 912432 from user X, "Error saving document"	
20 OCT	11:25	Noticed red light on air handling unit in computer room			

The great advantage of using a spreadsheet like this one is that you can enter the data as you collect it, and then sort the spreadsheet to show a single timeline. With many problems a simple timeline view of what happened is sufficient to make the causes of the problem obvious. But even if you need more analysis, a Timeline Analysis makes a great starting point to ensure that the required data is available.

Kepner-Tregoe Problem Solving

Kepner-Tregoe Problem Solving and Decision Making is a proprietary process for solving problems, making decisions, prioritizing issues, and analysing potential risks and opportunities. It is particularly effective when you are analysing a problem that has many related incidents. I really like the training courses that are available from Kepner-Tregoe, but I am probably prejudiced as I used to teach them!

The problem solving process starts by describing the problem from a number of different perspectives, often summarised as What, Where, When, and Extent, as follows:

- What component/service/system/activity is failing?
- Where is it failing?
- When is it failing?
- How much, or to what Extent, is it failing?

For each of these you should also ask the negative question:

- What component/service/system/activity IS NOT failing that you might have expected to fail as well?
- Where is it not failing?
- When is it not failing?
- How much, or to what Extent is it not failing?

This combination of perspectives allows you to hone in on the exact symptoms of the problem. By asking the IS NOT questions as well as the IS questions you define the boundaries, which can often lead to new insights.

After describing the problem like this, you then go through a number of other steps:

- **Establish possible causes.** Look at all of the differences between the IS failing and the IS NOT failing to see if this might indicate a possible cause. Consider any changes that have happened and think about how these could explain the exact symptoms. Also use your experience of other similar things you have seen in the past.

- **Determine the most probable cause.** Consider each of the possible causes and ask how it can explain the exact symptoms you have. Can it account for the exact IS and IS NOT that you have described?

- **Verify the true cause.** Once you have identified a probable cause for your problem, don't just rush in and fix the problem, think about what test you could carry out to verify that this really is the cause. This will help to ensure that you apply the correct fix, reducing the risk of introducing new errors while attempting to fix the wrong cause.

- **Think beyond the fix.** This step involves thinking about the possible consequences of the cause you identified. Could there be other similar things impacted? Could there be other consequent damage that you haven't noticed yet? Could the fix lead to other problems? Etc.

TIP#5

Define Business-Focussed Problem Management Metrics

Like many other areas of IT service management, effective problem management is critically dependent on the attitudes, behaviour, and culture (ABC) of your people. If you have great people with good skills who really care about problem management, then you will probably do a good job. Implementing a formal process can certainly help, but it is much more important to get the ABC right.

Metrics have a number of different purposes:

- They can influence the behaviour of your staff, and thus indirectly influence attitudes and culture
- They can help you to understand whether you are meeting your goals and objectives
- They can show you trends so you can see whether things are improving
- They can provide the data needed for reporting to your customers

It is important to think about all of these when you are defining metrics. I have seen some terrible problem management metrics that succeed in influencing staff behaviour in exactly the wrong way, without helping to achieve any of the other purposes identified here.

One metric that is very commonly used focuses on the length of time taken to identify the root cause of problems. This metric is particularly bad because it:

- Drives inappropriate behaviours. People focus on finding a single root cause as quickly as they can, rather than on taking the time needed to understand all the contributory causes. This metric also leads to people not logging problems for things that look difficult to analyse.
- Has only limited relevance to the goals of reducing the frequency of incidents and reducing the impact of incidents that can't be prevented. Both of these can be achieved without ever understanding root causes, and the length of time taken to reach root cause has very little relevance to the length of time to eliminate the problem or reduce its business impact.
- Provides very little information about whether problem management is improving. The thing it measures is not closely enough related to the outcomes you want.
- Provides internal data that is not relevant to real customer concerns, and encourages customers to become involved with the internal working of the service provider.

So what kind of metrics are better than the length of time to root cause? Firstly you need to understand your goals and objectives. Typical goals for problem management are:

- Eliminate problems that cause repeated incidents or major incidents
- Prevent the recurrence of major incidents
- Reduce the impact on the business of incidents that cannot be prevented

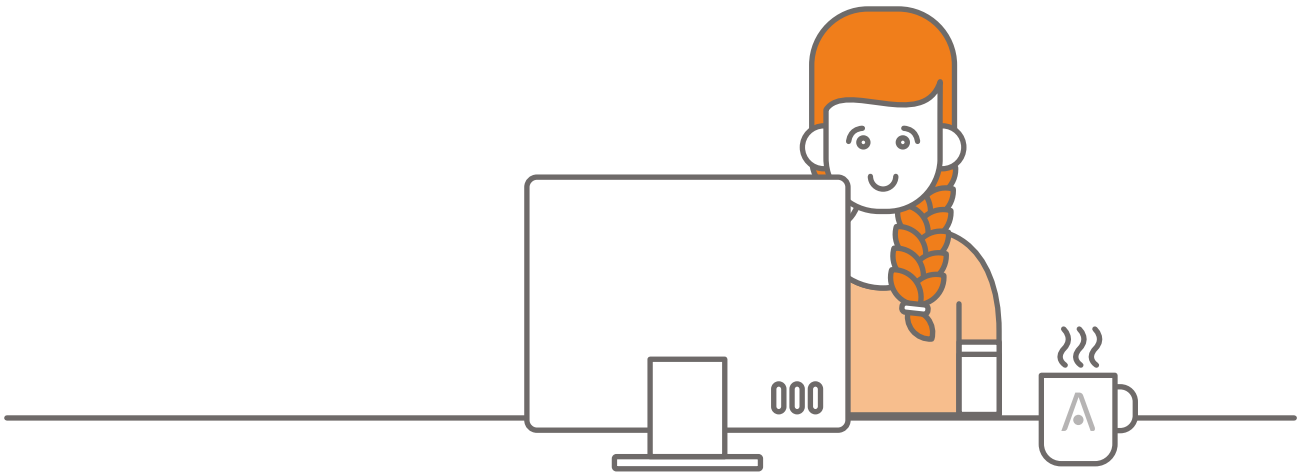
One of my customers had a very sophisticated way of calculating problem priority. This took into account how frequently the problem occurred, the business impact of the problem, and the effectiveness of the workaround. Their key metric for problem management was:

- Length of time required to reduce problem to priority 3 or lower

This reduction in priority could be achieved by eliminating the cause of the problem, or by implementing an effective workaround.

The great thing about this metric is that it measured something that really mattered to the customer, i.e. has the IT department done something to prevent this problem from impacting my business? Most IT organizations don't have the maturity needed to use a metric like this, so here are some simpler metrics that you could use to help you understand whether you are meeting your goals:

- Percentage reduction in the number of incidents caused by previous month's "Top 5 problems"
- Percentage reduction in the impact of incidents caused by previous month's "Top 5 problems"
- Average length of time required to create a problem workaround
- Effectiveness of problem workarounds (as judged by end users, or by the service desk)



Summary

5 Tips To Help You Improve Problem Management

by Stuart Rance

As you read through the ideas in this document I hope that you have seen how easy it can be to create value with problem management. The key things to remember are:

Tip 1 - Log your problems

Make sure you log problems for frequently recurring incidents and for major incidents. Create “Top 5” or “Top 10” problem reports to help you focus on the highest priority problems.

Tip 2 - Focus on workarounds, not root causes

Don’t spend weeks analysing root causes while the business is suffering; focus initial efforts on defining a trigger (to identify incidents related to the problem) and recovery actions (to get the business working as quickly as possible).

Tip 3 - Focus on improvement opportunities, not allocation of blame

Every problem has multiple contributory causes, not a single root cause. Each of these contributory causes may give you an opportunity to improve. People are NEVER the root cause of an incident.

Tip 4 - Train staff in analysis techniques

Technical people need to learn how to analyse problems; it doesn’t come automatically with their technical knowledge. Find opportunities for mentoring as well as formal training to help them develop the skills they need.

Tip 5 - Define business-focussed problem management metrics

Metrics should drive the behaviour you need from your people, as well as helping you to understand how well your process is working and how well you are meeting business needs. Time to root cause is a very bad metric for problem management. Measure things that matter to your customers.

If you follow these tips, then you should be able to get problem management working for your business, and after a while you’ll be amazed at what a difference it makes.