

Measuring the Quality of Problem Management in an ITIL Environment

BY BERRIE SCHUURHUIS
CONSULTANT – SERVICE EXCELLENCE

In an attempt to find the customer value created in an Information Technology Infrastructure Library (ITIL) environment, there's a strong need for performance indicators. Given the structured framework of ITIL in a mostly well-automated part of the organisation, measuring the quality of work is often related to measuring time-related parameters. However, does focusing on "time" and not the "quality" of outcomes truly measure the value of "Problem Management"?

A CLOSER LOOK AT HOW PROBLEM MANAGEMENT IS DONE

Understanding how analysts and engineers work on problems, find root causes and follow up with appropriate actions sounds like an easy task. Once one has access to the application used for documenting ITIL Problem Management, the case content can be read. All it seems to require is access to the case management tool and some skills to use that tool.

However, asking Problem Managers how they handle problems will typically uncover the true procedures that describe what steps they take when finding and working on problems. These documented processes and procedures are very helpful: expectations are very clearly set on the steps to be taken to progress problems that require attention.

Reading problem tickets, or asking Problem Managers on how they fill in the steps in the procedures, seems a logical next step to find out more about how value is created in Problem Management, as this is where information is really gathered, data is analysed and conclusions are drawn. So, how does the performance of Problem Management get measured? Many organisations seem to measure timing-related parameters around the problems, or counting the number of problem tickets in a given state. Examples include:

- Number of open problem tickets (backlog) per group of applications, considered over time
- Average age of open problem tickets, often considered over time
- Average time to find root cause in problem tickets
- Number of recurring problems

Considering the goals of Problem Management: to find causes of problems and proactively take actions to avoid future incidents and problems, how well do the examples above tell how successful a team is towards these goals? Are we asking for one thing, and measuring something completely different?

A REAL LIFE EXPERIENCE

About two days into an assessment of how Problem Management was being handled in a global IT department of a worldwide company, we decided to take a break to compare our findings amongst the participants in the assessment. Fields that were being considered for inspection included the ticket summary and problem description, as well as individual progress updates and the resolution descriptions.

The pattern seen was that in most Problem tickets, the summary was clearly indicating the affected application or hardware and what was wrong with it, followed by some underlying data in the detailed problem description. Further updates would typically indicate how the problem was traveling through the procedural steps of Problem Management as time was progressing, reaching a conclusion in the resolution description.

Although this seems like an individual case, it represents the pattern that was seen amongst the team doing the assessment. Talking through other experiences the following picture was made, representing the observations seen:

An example of a problem ticket:

Summary: login to application server ABC takes multiple minutes

Detailed description: Starting this morning users in the APAC region raised incident tickets for slow response during login to the ABC application server. As a result customers were kept waiting on the phone as order entry by the new shift was slowed down. The incident was cured by restarting the application front end. The application log (see attachments) shows some time outs on DB049 access during the login procedure.

Update 1: Tuesday 04 January 20xx 10:03:29

Involved Database team for checking state of DB

Update 2: Tuesday 04 January 20xx 10:07:12

Involved Network team through a call out to check network latency between Singapore and Japan

Update 3: Tuesday 04 January 20xx 14:37:49

Mail from Fred (attached), stating that according to the database team DB049 is running fine, no special observations.

Update 4: Tuesday 04 January 20xx 17:23:02

Steve reports latency between Singapore and Japan is back to normal since 9:30 this morning. Full details in email, attached to this ticket.

Resolution description: Kuala Lumpur switch to be upgraded to latest patch level in next maintenance window, next Thursday night.



This raises a set of questions on how conclusions were drawn and actions were taken or planned:

- What data needs to be gathered to find a cause effectively?
- How do experts make sure they have gathered the appropriate data at the appropriate time?
- What does the magic look like? What undocumented steps were taken? What undocumented thinking was done?
- What other causes were considered?
- What level of confidence did the resolving team have that the found cause really was the “true cause”?
- What side effects may actions taken to fix the problem have caused?

Answers to these questions may give a good insight in how value was created in Problem Management for any given ticket. The answers to these questions are typically not related to timing or numeric parameters around the Problem Management procedures. They are about the quality of data gathering and the quality of the thought processes by the individuals involved.

Get Control Over Recurring Problems — Get Stability.

Some may state that when “magic” is done well, the business will see a low number of recurring problems, which was indicated as a performance indicator for Problem Management above. This is true!

Unfortunately.

What a company gets through recurring problems is a message that the Problem Management process didn't do a good (or good enough) job in finding the root cause when the problem occurred for the first time. Since reoccurrence can take weeks or months to happen, this is a lagging and imprecise indicator for Problem Management performance. What is really needed is a way to measure the performance (and therefore the value) of Problem Management such that a company will be able to foretell that the number of recurring problems will go down. In other words: what are the leading performance indicators for Problem Management?

Finding measures that indicate how well problems have been solved may only have a mild effect for simple (low impact) problems, where recurrence wouldn't be appreciated but it wouldn't be catastrophic either. Some companies occasionally have critical incidents and problems where they balance on the edge of a catastrophic business event tied to one or more IT-related events, and they resolve to never ever go through that experience again! Measuring recurring problems and trends are not likely to be a good enough metric.

A Best Practice for Doing Magic?

Asking engineers and analysts for their internal thinking processes when they're handling problem tickets gets many different answers. This is completely different from when the same audience is asked how to configure a specific application or some hardware. It is quite obvious nowadays that a common approach for configuring an application or a piece of hardware has many advantages:

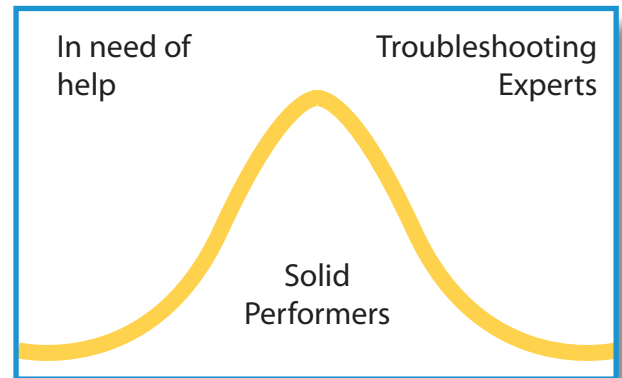
- A “best configuration” for the asset being used reduces variation
- A common understanding how asset adds value to the entire infrastructure helps with capacity management
- It simplifies communication on how assets are configured or changed
- It allows for seamless and high quality hand over and maintenance

Given these factors, it is remarkable that there is often no common approach for problem handling. As a result this remains as magic.

When a best practice for finding root causes for problems is established, it gives very similar advantages as a best practice for configuring an asset. Besides, it would enable a new language for troubleshooting with terminology that allows the documentation of what the “magic” looks like, and how conclusions are reached.

What Does the “Magic” Look Like?

There are many ways for finding the root cause of a problem. Some are more successful than others, and different people (without a standard framework) naturally have different approaches. The effectiveness of any group of troubleshooters falls somewhere along a bell-curve. Troubleshooting experts have a good reputation and can be given anything to work on with confidence. Solid performers are good for most tasks and have room to improve, and those with a poor troubleshooting reputation probably need help.



The Kepner-Tregoe (KT) method for Problem Analysis was researched and defined in the 1950s, and has continued to be refined and tested ever since. It is easy to recognise that this was many years before the acronym ITIL was invented.

It has been argued that a method which has been around so long can't be appropriate for the IT industry as neither IT nor ITIL existed at the time the method was first researched. It takes a closer look at the KT method for Problem Analysis to make a more appropriate judgment. The major steps in Problem Analysis consist of:

- Describing the Problem
- Listing Possible Causes
- Evaluating Possible Causes
- Proving the True Cause
- Thinking Beyond the Fix

For each of these steps, there are clear intentions and some sub-steps – which are put to work through the phrasing of questions and the documentation of the answers to get the right data feeding into the Problem Analysis thought process. This is all done without any specific product or issue in mind, and it's very similar to ITIL, which is working for all kinds of IT organisations. Problem Analysis is an approach for finding root cause for many different problems irrespective of the industry or technology.

Any kind of problem?

Well... yes! But there is a very specific KT definition of the term “problem” which is different, but matches quite nicely with ITIL. According to KT, three criteria must be true, before we trigger the Problem Analysis process:

1. There should be a gap between actual performance and desired performance. This is what we call a deviation (e.g. machine is not working, versus it should be working);
2. The cause for the deviation is unknown (e.g. not a Known Error);
3. There must be a need to know the deviation (e.g. enables to take action).

The result of going through a well-defined set of steps to find root cause is that troubleshooters can start communicating and documenting what has already been done and what is to-be-done in the process. An example of how gathered data is textually modeled to describe the symptoms of a problem is given in the picture below.

Known Magic

When the steps for a consistent and replicable approach to Problem Analysis are well understood, measuring the quality of a found root cause becomes much easier. If the magic in finding root cause is understood, it can be documented, reproduced, handed over smoothly and timed efficiently; which are all characteristics of a Best Practice.

Once an IT support organisation starts using a unified approach to Problem Analysis, the immediate quality or value of individuals and teams can be measured. This is exactly what KT

consultants do when assessing the quality of existing troubleshooting processes being done in an IT support environment. By reading through existing incident and problem tickets, and by estimating how much to structure the approach against a known standard, we can help generate a baseline leading indicator for the quality of troubleshooting.

As an example: IT staff who consistently document their Synopsis (or equivalent field in their case management tool) in terms of an Object with a Deviation (answering the question: "What is wrong with what?") appear to spend just over 10 percent less time on average before a root cause is found.

It all may sound so easy that this cannot be true – just documenting the object and defect that experts are planning to find the root cause for will save just over 10 percent on time to close. Well, you might be right: it may sound easy, but it is not. To get this thought process imprinted and reflexive requires a change in behaviour, and in the heat of the moment, under time and other pressures from the business this simple step can fall aside if not practiced and supported away from the high pressure issues. The steps for implementing a best practice for troubleshooting is well understood, but making the change will still take attention, focus, good planning and thinking. Fortunately enough, thinking is easy, but implementation teams may get distracted. The KT thinking processes, like Problem Analysis, are not a silver bullet which guarantee that root cause will be found. It is just a method that guides already knowledgeable experts towards the goal, and mileage in finding root cause may vary depending on the quality of data (and observability) that goes into the process.

Problem Analysis
Describe Problem

One Way traffic on < Truck Equipment >

What object?	< Truck Equipment >	< comparable equipment >
What deviation?	One way voice	Packets being lost
Where geographically?	< Named > Telecom < exact location >	In lab < test rig >
Where on the object?	< equipment name > < telecom card name >	< equipment name > < maintenance interface >
When first?	03 Jan 00:10	Before
When since?	Intermittently	Constant/Periodic
When in the life cycle?	After loading patch set 11	After previous reset
How many objects?	1	more or less
What is the size?	complete loss for 5 min.	more or less
How many deviations?	40 instance reported	more or less
What is the trend?	Stable	Getting Worse or Better

The latter is a key ingredient for success; just filling in the form, template or spreadsheet doesn't give a good root cause, because Problem Analysis is built on a firm foundation of hard logic that needs to be actively used. It still takes intensive data gathering, thinking and checking, which is no different from troubleshooting in an unstructured troubleshooting environment. The big change is that the steps in thinking become visible and they get a name, all based on a clear underlying plan for Problem Analysis. As a result of this we can measure and communicate about where we are and how we're doing in the process of finding root cause.

In this case measuring is not a database query showing how much time or how many tickets meet a given set of criteria. It gets to a rating that can be given by internal (troubleshooting) experts who judge the quality of gathered data in the distinctive steps of Problem Analysis. Such an assessment then becomes a leading performance indicator for the Quality of Problem Analysis.

WHERE DO WE GO FROM HERE?

Reading a book on playing a violin doesn't make the reader a great violin player. Similarly, just training an organisation on how to do better thinking in troubleshooting is not likely to turn the organisation into a world-class group of troubleshooters. It will take attention, exercise and dedication to embed the approach into the thinking approaches that individuals take, and the results will pay-off. Making an investment in how to find Root Causes in Problem Management is supporting investments in technical skills and experience leading to a workforce that is aware and well equipped as to what it takes to find good quality resolutions for complex problems. At the beginning of a Problem Management case, a manager will (still) never know how long it will take before the root cause is found, but there will be a clear and planned direction and the arrival time will be more predictable, which enables measurement of quality in Problem Management.





BERRIE SCHUURHUIS

Consultant – Tech Practice

Berrie Schuurhuis is a Consultant with Kepner-Tregoe Europe (KT) based in The Netherlands.

Before joining KT, he spent 10 years at Sun Microsystems; and since 2003, as a global implementation manager of KT ResolveSM in Sun and partner organisations around the world. These partners took care of troubleshooting for specific products, language support, or geographic regions. In this role, Berrie developed specific insights in cultural needs for implementations of Rational Thinking in different cultures and languages around the globe.

Berrie worked as a Support Engineer in local Dutch and European front-and back-line roles, giving direction to complex customer issues around hardware and software supported by Sun Microsystems.

Before that, Berrie worked as a consultant for the Computer Management Group (CMG) running technical projects in the telecommunications industry for public networks. And he started his career at AT&T Network Systems in 1989 in an R&D role in embedded software development in the early days of Object Orientation.

Today, Berrie is a strong advocate for Rational Thinking which he can combine with the technical skills that he developed in the IT and Telco industry, while keeping in mind that there's always a human at work. His career is built on a Bachelor degree in Electrical Engineering from the Windesheim Academy in Zwolle, the Netherlands.

For more information, visit us at www.kepner-tregoe.com
or e-mail us at european.enquiries@kepner-tregoe.com



® Kepner-Tregoe, Ltd.
Moorbridge Court
Moorbridge Road
Maidenhead, Berkshire SL6 8LT
England
+44 1628 778 776 Fax +44 1628 587 856
www.kepner-tregoe.com
e-mail: european.enquiries@kepner-tregoe.com

Kepner-Tregoe, Inc.
P.O. Box 704
Princeton, NJ 08542
609-921-2806 Fax 609-497-0130
www.kepner-tregoe.com
e-mail: european.enquiries@kepner-tregoe.com