White Paper

ITIL Report Suite: Health Check Part 2

WP0207 | August 2015

Jason Dove

Jason Dove is an ISEB accredited Business Analyst, Developer and Professional Writer.

He consults for multiple leading businesses across various industries – from marketing to counter-terrorism.

Jason specialises in Business Intelligence related disciplines, with a strong emphasis on ITIL systems - a commonly overlooked opportunity for organizations to get the most from their IT investment.

With over 15 years of experience in the industry, Jason has leveraged his knowledge into that of author, blogger and is a contributor to print and online publications.

In this second and final instalment, we are going to continue looking at the different aspects of the ITIL Reporting Suite and how to identify any discrepancies within those areas. This will be followed by a discussion on how to make the most of the results.

Software Flaws

This Health Check is one of the trickier ones to quantify, despite seeming definitive at face value. ITIL software should lock its users into the agreed process and kerb any attempts at deviation or incorrect data input.

There will always be human error, but a good ITIL Management System enforces correct behaviour as much as possible.

Unfortunately, not all ITIL software is equal and there is a good chance that it is limiting in some ways or, which is arguably worse, is too free in others. What this means for the ITIL Reporting Suite is that each report is likely to have been designed to a specification that was based on a logic process that has since become outdated due to the software forcing/enabling another series of behaviours.

The Check

Actively seeking out software flaws as a discrete task would be time consuming and unlikely to reap encompassing results.

These software flaws will appear organically during the other Health Checks, so no specific investigation is required.

All (that I've seen) ITIL management software includes Stop Clock functionality in some form and generally have organizational level logic attached to when it is acceptable to use it.

Stopping the clock because the Stakeholder needs to provide further information = good.

Stopping the clock to allow a long lunch without breaking any SLAs = not so good!

The responsibility for the above is squarely on the shoulders of the person updating the record and falls under the previously mentioned 'Reports and Processes' section.

But when the ITIL software allows further updates and work to be done and recorded while the clock is still stopped…that is what this section is about.

A Ticket can bounce from Resolver Group to Resolver Group and expect each group to check whether a Stop Clock is currently applied is unlikely to be part of any process mapping.

This can lead to Tickets going all the way to Closed while still being subject to a stopped clock. As all reporting will work on the premise of a Stop Clock being followed by a Start Clock before work commences any measurement of Stop Clock time and the true time taken on the SLA is going to be flawed.

The Solution

Unlike the rest of these Health Checks, it is not a matter or fixing a Metric calculation or tweaking a Report, even though it may be tempting to as often software flaws can be worked around.

These sorts of issues normally fall into data quality issues and should be handled via Exception Reporting with the onus on the person who owns entry with the bad data to correct it.

Sometimes the problem is something bigger. Software Flaws such as, "The elapsed time doesn't work" will take more than some Exception Reporting or a Report tweak! The idea of ITIL software not calculating elapsed time may seem an unlikely example, it often occurs as a result of bad configuration or simply a cause of needing to record several types of elapsed times against one event (often SLAs and OLAs follow different Support Hours and require different measurements).

In these extreme cases there is no option other than work around the problem as gracefully as possible. This is when Data Warehouses become very handy in holding any 'handmade' logical workarounds in one central place and storing the results.

If reporting directly from the source Database without a Data Warehouse, keeping bespoke intermediary logic up to date across all the reports is a much bigger challenge.

Note: ITIL software often requires a lot of set-up work to function correctly and to its full potential. A lot of “software faults” often turn out to be incorrect or missing configuration. If that happens to be the case, correct the configuration rather than work around it or raise additional Exception Reports.

Cross Column Compliance

OLAs have more holes than a sieve for the data to escape through! When measuring an OLA the whole point is to cover all aspects of the work done by a Team or individual during a certain time period. Unfortunately, by their nature, OLAs can be passed to multiple Resolver Groups as each OLA that makes up the Process is addressed (or miss-assigned!).

The same is true for SLAs to a lesser degree, and a similar approach should be taken to validate them. OLAs are the more complex of the two, so I’ve chosen OLAs to focus upon for this whitepaper.

It is easy for things to get lost.

Consider this checklist:

- Process steps opened before the time period and closed within the period.
- Process steps opened during the period but not closed until after.
- Process steps worked on during the period, but passed to different team for completion.
- Process steps worked on during the period which were opened before the period and not closed until after.
- Process steps started and completed within the period.
- All different types of job undertaken by the team in the period (e.g.: replying to phone enquiries as well as doing the actual work this generates).
- Process steps started (or closed) outside of agreed working hours.
- Cancelled processes.

To miss any of these is to do the Resolver Group or Team in question a disservice and discount the hard work they do.

The Check

Good OLA reporting will highlight any Tickets getting lost in the Process steps. It should be simply a matter of comparing the columns and ensuring they are consistent within themselves.

If the featured columns in the Report do not paint a complete picture (e.g.: ten tasks Brought Forward, five new ones Opened, nine Closed and two remain Outstanding…then four are unaccounted for) investigate a trouble causer and add more columns until the Report makes sense.

If that does not fix the discrepancy in the first attempt, repeat the investigation for the next most likely record.

And so the analysis circles around until the Report shows all pertinent variations supporting the OLA set, or the broken processes are fixed.

The Solution

This fine tuning of the OLA reports may not even impact the overall KPIs but including them paints a fuller and more honest picture and stops the report being challenged or questioned over accuracy.

In a Metric heavy environment like ITIL it is easy to forget that Reports should tell a story, or at the least, content a complete set of facts in order to be of full value and provide context to the OLAs. This context can often highlight issues in process bottlenecks with greater clarity than OLAs alone.

Note: I love these throughput data grids in their own right as a personal choice. But even if the Stakeholders are not fans, producing this style of Report has this additional benefit of easily highlighting any issues for remedial work in the background. For that reason alone it is worth the development effort.

Outstanding Versus Carried Forward

This also applies primarily to throughput style reports and is similar in concept to the cross column compliance but is harder to catch as it spans across different instances of the same Report.

Unfortunately, when this issue is discovered it is usually on a live system by a Stakeholder comparing two instances of the same Report for different reporting periods.

If the reporting period is monthly, there is an additional check that is worth the effort to catch another, very specific error.

The Check

The Health Check is the same, a simple enough comparison between two Reports. Any Business Intelligence / Reporting encourages a blinkered view where the focus is on each Report instance as a discrete item without looking at a wider picture (i.e.: consistency across different reporting periods).

Week 27

Task Priority	Brought Forward	Open	Closed	Closed in OLA	Closed in OLA (%)	Outstanding
1 - Priority One	2	2	3	3	100.00	1
2 - High	3	4	5	5	100.00	2
3 - Medium	18	8	25	16	64.00	1
4 - Low	12	5	16	15	93.75	1
Total:	35	19	49	39	79.59	5

Week 28

Task Priority	Brought Forward	Open	Closed	Closed in OLA	Closed in OLA (%)	Outstanding
1 - Priority One	3	5	6	5	83.33	2
2 - High	5	8	5	5	100.00	8
3 - Medium	15	2	17	16	94.12	0
4 - Low	10	8	15	15	100.00	3
Total:	33	23	43	41	95.35	13

The problem with monthly reports is that most (probably all) reporting software has built-in functions for handling months. Which may sound like a good thing, but it encourages logic based on month comparisons like this:

MONTH(This_Month_Field) < MONTH(Next_Month_Field)

At the time of writing, it is July (the seventh month), with August (the eighths month) next month. Seven is indeed less than eighth and the above formula will accurately return 'TRUE'.

Which is fine for January through to November, but when it is December (the twelfth month) and the next month is January (the first month), this logic breaks.

Another quick Health Check in this arena is if the Metric calculations are based on Ticket Statuses. These can/will change over time and skew metric results when comparing the same calculation over time.

The Solution

The two example Health Checks are by no means exhaustive and a myriad of subtle errors in logic can cause this difference between reports. It's also easy to miss for an analyst, developer or even a tester: as consecutive report comparison are seldom done.

So checking the Outstanding Total or this month's report against the Carried Forward of the previous month is good, but to be completely sure of the report's accuracy, it is worth checking the year end as a separate piece of logic.

Whether the error is as heavy handed as that example or not, once identified it should be simple enough to fix being a change of existing ETL logic.

Using the Health Check Results

The point of this Whitepaper is to provide means of validating the ITIL Reporting Suite. What to actually do next if the health check fails is another matter.

Depending on the structure of an organization, the real world implications may be nominal, so while any errors obviously require fixing, it may not be the imminent disaster it first appears. On the other hand, of course, it can go the other way and a small error can cause major issues.

With the best will in the world, every implementation of ITIL is so different that it is impossible to pre-empt the issues that may be encountered and each must be judged on its own merits.

Keeping that in mind, what follows is a collection of general tips that should be kept in mind when remediating an ITIL Reporting Suite.

Do: Hire an expert

ITIL reporting is not easy, and speaking as a consultant who has worked across a myriad of business sectors and reporting arena, I would personally class it as one of the most challenging areas to report on.

To carry out the kind of work being discussed in these whitepapers, an 'expert' can be defined as someone who has worked on multiple ITIL Reporting Suites using the reporting software and approach (rational reporting or data warehousing).

Of those three, working on multiple ITIL reporting should be single most important aspect, while the other two can be easily picked up by an experienced practitioner.

Do: Start at the lowest level

A good ITIL Reporting Suite builds in complexity from the ground up, so it follows that any remedial work should follow the same approach.

Here is the distribution diagram from the part one of this series for reference.

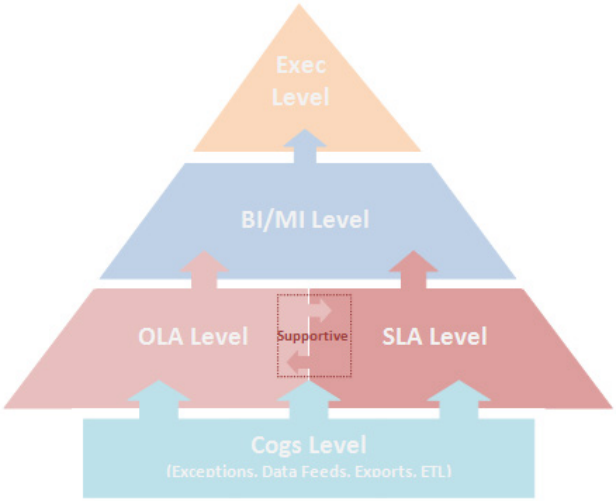

Figure 2: ITIL Metric Distribution and Flow

For many Organizations, just overhauling the coverage of metrics reporting to match above diagram will solve a lot of reporting issues.

Do: Engage with Stakeholders

For each Metric there are three groups of people who should be engaged during the Health Check and any subsequent remedial work:

1. Metric Stakeholder
2. Report Stakeholder(s)
3. Report Audience

All three have valuable insight that can make or break this type of work as well as provide the opportunities for some sneaky value "adds" to

the ITIL Reporting Suite. This is the chance to align the skewed column that has annoyed them for months or change the stacked Bar Chart for something that is actually readable…or any of a thousand little things that set a good implementation apart.

Do: Go after the big wins first

Occasionally a simple fix can reap massive rewards across the organization. These are usually the more generic issues, such as correcting elapsed time measurement.

Any fixes of this type can be presented early on to show the benefit of the wider piece of work.

Do: Focus on functional areas

When writing about ITIL reporting I endeavour to keep the content as generic as possible and not concentrate on Incident, Problem or Change Management unless a specific example is required (when I tend to use Incident Management).

However, when carrying out these health checks and even more so when implementing any remedial work concentrating within one area at a time is a great way to split the effort into discrete silos.

Don't: Be hamstrung by de facto Process avoidance

This is the trickiest item on this list because it is often wrapped up in wider organizational issues.

It occurs when one Team/Resolver Group is not following the same process as all the other Resolver Groups. This is fine if their unique Process is mapped, measured and reported, but the problems occur when it is not.

While this is covered earlier in this whitepaper as a Health Check subject, it is mentioned again here in this context to emphasize that it needs additional consideration and can be a large obstacle to a success Health Check and remediation.

Don't: Exclude in-house talent

No matter how experienced or gifted a consultant may be, they will not have the specific, in depth knowledge that a long term resource would have.

Losing this knowledge will hamper an expert and slow down initial progress, but looking at the bigger picture: this is a great opportunity to up-skill in house resources and lock in that all important continuous improvement.

Don't: Introduce anything new until underlying issues are resolved…

…except when you have to!

In the perfect world, the ITIL Reporting Suite should be locked down while remediation work is being carried out. Trying to fix an ever changing target is an additional challenge to an already complicated piece of work.

This is not a perfect world and there will always be that allegedly super important report with political backing that can't be ignored. But these exceptions should be just that, exceptions.

Don't: Fix Report logic piecemeal

The point of reviewing the entire suite of reports as a single entity is partially to identify reporting gaps, but primarily to allow a planned and concerted effort to resolve any reporting issues.

Unless considered urgent by the Report Stakeholder, a Report should only be amended when all the fixes for that Report can be applied.

Unless considered urgent by the Metric Stakeholder, a Metric should only be amended when all the fixes for that family of Metrics can be applied.

In all over cases, only present a Metric and/or Report for use when it is completely fixed and beyond reproach.

Summary

All those Do's and Don'ts are far from exhaustive, but they are my most encountered gotchas and should help navigate through some of the traps that await.

Depending on available resources, overhauling a flawed ITIL Reporting Suite does not have to be disruptive in all but the worst cases. And in those 'worst' cases, it can be a great justification to implement a new, planned and complete solution that will bring real value to the Services that are the back bone of any Organization.

The next series in this collection of whitepapers is focused on planning and implementing an ITIL Reporting Suite in line with the diagram featured in this paper. It will go into much greater detail on what is needed for a successful suite from a greenfield development but the approach is pretty much the same for repairing/fulfilling an existing suite.

Orbus Software UK
London

Orbus Software US
New York

Orbus Software AUS
Sydney

Orbus Software RSA
Johannesburg

enquiries@orbussoftware.com | **www.orbussoftware.com**

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG. T/A Orbus Software. Registered in England and Wales 5196435