White Paper

ITIL Report Suite: Health Check Part 1

WP0205 | July 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 organisations 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.

Recently I completed a short series of white papers featuring the most common and damaging gotchas/pitfalls that undermine the quality of reporting for the ITIL framework.

After all that scaremongering, I think it is prudent to provide a method to check the quality of an organization's existing report suite, or as a sanity check during the development of an ITIL Reporting Suite on a greenfield site.

None of these checks are technology or platform specific, and most can be carried out with a minimum of technical knowhow.

The remainder of this white paper consists of various Health Check activities, with each one uncovering a different aspect of erroneously implemented reporting.

I think everyone knows the story about the business which went bankrupt due to an end user's mistake on an accounting spreadsheet....I don't even know if the story is true, but think the warning is a valid one regardless.

So can an ITIL Performance Report create such destruction?

Probably no.

But it depends on how pivotal the report is to business decision making. Many ITIL reports measure services by third party suppliers, and erroneous measurement can cost.

Another risk is that if a team (3rd party or otherwise) realizes certain work is not measured it can lead to a lackadaisical attitude in this area. A prime candidate for this scenario is when a team is not measured on how quickly they respond to Tickets assigned incorrectly to them.

That is a localized example, but incorrect measuring of work volumes can lead to unneeded redundancies followed by overworked remaining staff.

At the very least, the organizations who have little or no faith in the information provided, will not take it into account in their decision making....which is pretty much the whole point of ITIL Reporting Suites.

The following Health Checks can be carried out in any order, but they are presented in a sequence of 'most likely to have the greatest impact' and some may be more relevant than others depending on the organization and their current ITIL Reporting Suite.

1. Report Distribution

This first Health Check is the most general and easiest to evaluate to ease us into the review process.

The ITIL Reporting Suite has one function and that is ensuring the entirety of the Service Catalogue and its supporting KPIs are clearly reported.

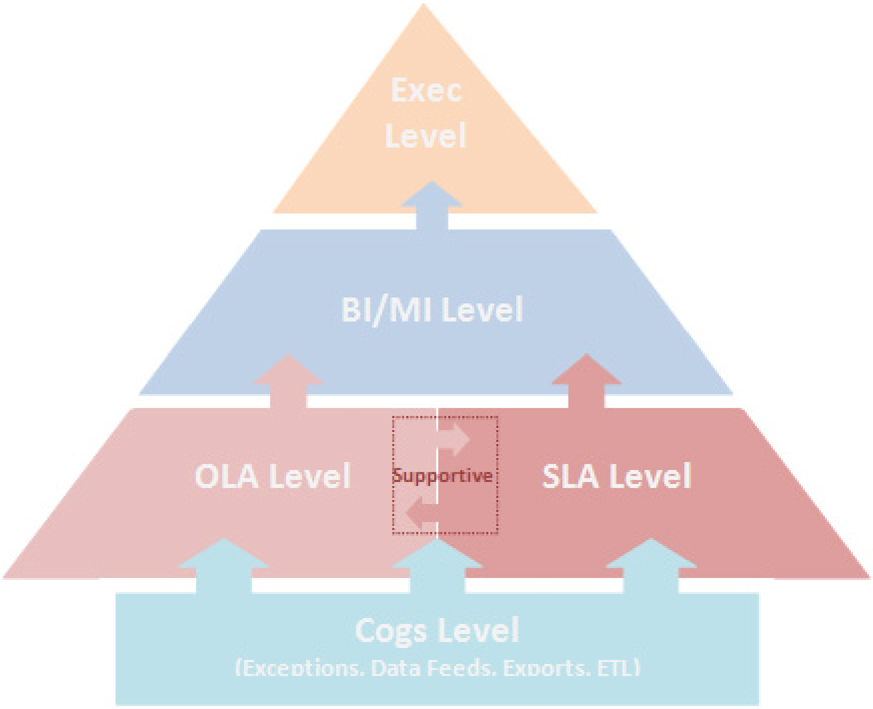

Figure 1: ITIL Metric Distribution and Flow

To be honest, it is very rare that ITIL reporting is this encompassing, but it really should be. Any gaps between metrics and the audience that needs to see them are a blind spot that leaves the organization at risk.

Even comprehensive ITIL Reporting Suites can fall out of date over time and develop cracks in their coverage.

The Check

Easily enough, a quick check to ensure each metric is reaching all levels of its audience. Who the audience is will vary on the Metric and the Service, and will also vary from organization to organization.

A general audience rule that can be applied across ITIL is that each Metric should have:

- Worker
- Manager
- Owner
- Stakeholder

This can be applied to any Metric, whether it is for an Incident or Service Availability.

The Solution

Even easier, develop reports to fill the gaps! Which may not be bad news depending on the size of the problem. Sliding another page into a Reporting Pack is one thing, but developing new Dashboards is quite another.

From this high level check, let's dive into the data.

2. Rubbish in…

…rubbish out. We all know variations of this phrase and have heard it enough that it seems to have lost its meaning. Bad data is the nemesis of any reporting suite and ITIL is no exception.

Speaking from bitter experience, there is nothing worse than having a report criticised for quality of content. Especially when the report itself was correctly developed but is being sabotaged by poor data.

In the same way as testing electrical equipment begins with the fuse, ensuring that the underlying data is correct is the first step in reconciling a reporting suite.

Unfortunately, a questionable report can quickly lead to a lack of faith in the whole reporting suite that undermines every output. In order to be valuable to the organization the reporting suite needs a level of authority: enough authority to be heeded as an expert in metric production and presentation in its own right.

"One version of the truth" is often lauded as a goal in reporting suite design, and it is important that reports do not conflict each other (more of that later!), but providing a consistent inaccurate picture in order to avoid criticism is the tail wagging the dog.

Hopefully my mini rant has emphasized how important this point is!

The Checks

If an ITIL Reporting Suite has been live for a period of time, at least some of the data issues will have surfaced. Sometimes it is anomalies in charts which everyone learns to ignore or the recurring commentary that highlights why a bad result is not really bad.

Report Commentary

Any report/dashboard/reporting packs that are focused on the performance of a Resolver Group or individual Service, should provide a mechanism for the Service Manager to either sing the praises of their subordinates or provide mitigation when KPIs are missed.

This is usually presented as a narrative alongside the data/chart that shows the relevant KPIs.

On a mature ITIL implementation, these commentaries can often provide an insight to any known issues within the Reporting Suite. I am introducing this for checking bad data, but it is a great starting point for most of these health checks.

Reading through several months of review commentaries will often show recurring patterns in issues.

From example, a single instance of a commentary blaming bad triaging for a failed Service is no cause for alarm, but if that claim is being repeated at each review it is a different matter. And do not underestimate the gratitude when long standing issues are addressed.

The checks are as wide ranging as the data and will vary from organization to organization. So the following bullet points are more a guide of how to think around possible data issues (although each is a real world example):

- **Data Entry Fields:** anything where data is entered manually, rather than being selected from a pick-list. Ironically, it is more likely to be those '*Required Fields' that are so important the software would sooner have a dash, period or 'aaaaaaaaaa'.

- **Logical or Lazy Errors:** these are the mistakes like a Close Date being before the Start Date, that particular one being a favourite in early January when the habit is still to write the previous year.

  But this can also mean picking the 'General' option from a pick-list rather than searching for the correct item.

- **Test Data:** this shouldn't need saying, but unfortunately this is all too common. Test data and/or early implementation data (which often is as bad as test data) can often be found in live

ITIL systems causing all sorts of anomalies and gremlins in the Reporting Suite.

- **Names and Process Changes:** Over time, Organizations change and working practises evolve. It is not uncommon for the reported processes to fall behind what is actually happening in the real service world. This essentially makes the reports incorrect, which then require realigning to the current business as usual model.

  An ITIL Reporting Suite has to run to stand still and just stay aligned to an evolving system and the first hints that it is falling behind will be in the data quality.

The Solution

The solution is the same for all of the above, process tightening quickly followed by a data cleanse. Removing any trace of the previous bad practices aids in the adopting of the new improved process and stops any historical blips in data quality showing in trend reports.

3. The Bigger They Are The Harder They Fail

The bigger the organization, the bigger the scope of the ITIL Reporting Suite and the more likely information is going to be lost or be skewed.

If a single Network Support person wanted analysis on everything they had ever done,one single report would be all encompassing. Once the sole support acquires a colleague and the work is divided between them, a crack appears. Some jobs may belong to one of them, some to the other, and some to both. Without proper care the shared owner jobs can vanish from reporting...or be counted twice, which equally skews the results.

Now the two Network Supports want to know how much work each have completed this month. So now jobs which were started before the month and closed in it can vanish, alongside the jobs which both worked on.

Extrapolate this up to a multi-national company with thousands of Stakeholders across hundreds of Resolver Groups and the potential for missed information becomes exponential.

The Checks

- **Complex Triages:** Once an organization grows beyond the scope of a single Service Desk for Incident Management and starts using Triage, Teams Tickets can travel through numerous owners before arriving at a Resolver Group who may or may not have the skills to address the issue.

  Of course, this level of confusion can be easily replicated by outsourcing the Service Desk to a third party who does not know the Service Catalogue as well as home grown talent.

  The end result of either scenario is Tickets (usually Incidents) becoming lost in the system and being passed back and forth between Resolver Groups with no one taking ownership.

- **Resolved to Closed:** Getting a Stakeholder to find the time to confirm an Incident can be Closed is the bane of many a Service Desk. Once an Incident is Resolved, the inspiration for the Stakeholder to spend more time on it vanishes as they are able to return to doing their actual job.

  Some organizations may go the route of automating Incident closures after a set number of days, or ignoring Closed as the end Status and misuse Resolved to mean both. However, this undermines the role of Incidents in ITIL as a stopgap 'keep the cogs turning' solution, and treats them more like unofficial Changes.

- **Orphaned Tickets:** This one is more likely to manifest over time in more mature ITIL solutions. As the organization evolves and Services are retired or replaced, or Resolver Groups are merged or disbanded: Tickets become lost in forgotten queues, whether as an Incident or a Problem for a Service which no longer exists.

  Once again, the above are real world examples, but it is not an exhaustive list and not all may apply to every organization. Some organizations struggle with child Incidents, unimplemented Problem solutions (usually for more expensive solutions!), automated Incidents that are widely ignored, but occasionally shouldn't be and so on. Every organization has its own challenges and as much as ITIL standardises a generic approach, working practices and real world application make for a myriad of issues and approaches that is unique.

  With a comprehensive ITIL Reporting Suite, many of this type of error will manifest in one or more reports which is a good start, but it is worth investigating as a discrete piece of work.

  Hopefully the theme is clear - any situation in the ITIL processes that may lead to a Ticket becoming 'lost'.

The Solution

The first step is to look at the totality of the open Tickets in existence within the organization.

Start with the oldest Tickets, look for commonalities between them and place them in logical groups.

Identify why each group is still live. The reasons will vary greatly, but the solution is often the same: update the underlying Process.

How much effort is required to amend a Process will depend - informing a Triage Team to direct Incidents of a certain type to a different Resolver Group should be easy enough but finding skilled support for a Service that currently has no Resolver Group is a different matter entirely.

For any Tickets which are no longer relevant, do not Resolve or Close them as this can skew regular reporting.   Rather, set their Status to Cancelled and add Commentary as to why this metric has spiked (assuming Cancelled Tickets is reported against).

Once addressed, develop a report specifically to monitor new occurrences of the same issue to ensure the success of any amendments.

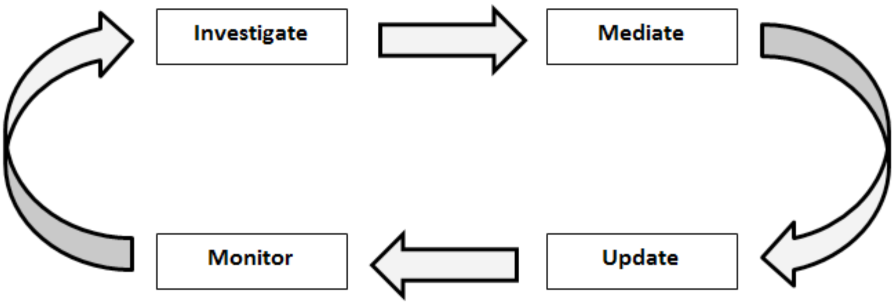

Figure 2: The Usual Continuous Improvement

Applying the above diagram on an on-going basis is strongly recommended, however, in the real world finding time for this sort of work will always be a struggle unless there is a suspicion of poor quality reporting.

However, with a bit of analysis and identifying the logical groups of errors, it is possible to attack data quality in a piecemeal manner and announce each success as it impacts reporting value (hopefully in a positive manner!).

4. Software Flaws

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

They 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 (arguably worse) 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 Checks

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

Stopping the clock because the Stakeholder needs to provide further information is 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.

The Solution

This is a very specific Health Check which differs from the rest in this paper in that the issue will often lie outside of the organization and be the responsibility of the software vender to address.

As this can be a protracted process, it is best to put official workarounds in place to either correct known failings or exclude bad records until manually mediated.

Data Warehouses

If the ITIL Reporting Suite is based on a Data Warehouse, these types of data tweaks would be carried out by the ETL.

If the Data Warehouse was built on a mature ITIL system, it will probably handle most of the issues this Health Check would identify but is still worth doing. Software issues are normally well known and do not take much time to investigate compared to some of the other Health Checks in this paper.

Applying additional data checks in existing ETL processes is a lot less labour intensive than correcting individual instances of reports.

This discussion will be continued in Part 2.

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