

# ITIL Dashboards by Example

Jason Dove



# Contents

2

Introduction

3

The Point of Dashboards:

Dashboard Scope  
Scope Breadth  
Scope Depth

4

Agile Stories

5

The SLA Metric Dashboard:

Target Date Range  
Scope Drilldown  
Using the Functionality

6-8

The First Drilldown for the Example Dashboard

Level One: Dynamic Contents Page  
RAG Abuse!  
A Few Final Design Options

9-12

Level Two: The Main Page

The Death of Pie  
Components for a Full  
PictureDrilldown Functionality

13

What isn't there!

14

Hierarchy Equals Volume

14

Types of Metric Dashboard

14

Getting the Data

15

Summary

# Introduction



In this guide I am going to look at designing and developing dashboards with a specific view to maximize their usefulness for ITIL based solutions.

The beauty of dashboard-based reporting is that virtually anything is possible. The downside to dashboard-based reporting is that virtually anything is possible! With all this freedom at our hands anything can be developed, whether it should be or not.

My aim with this guide is to work through a specific dashboard solution to provide a clear optimal means for monitoring ITIL services and their supporting SLAs.

By approaching this requirement with fresh eyes I hope to illustrate how a dashboard can be much more than a hodgepodge of charts.

---

# The Point of Dashboards



Before we plunge into looking at the example, I would like to lay out what I consider the primary reason for using a dashboard. Namely, to present a range of information that provides a comprehensive and meaningful view of a 'subject'.

I will make one promise before going any further, which is that no gauge/dial type charts will be used. Not that there is anything inherently wrong with gauges - they do have their place in the range of available charting options, but they are heavily overused in dashboards. I guess this is because there is a strong mental link between dashboards and gauges in the real world.

## Dashboard Scope

The first step is to know what should be included in a dashboard. A typical report focuses on one business area and tends to be in one physical location, in regard to data. Because it is expected for a dashboard to span functional and/or logical divisions, a greater range of choice is possible.

This widening scope for dashboards is made greater still when we include drilldown functionality, something which tends to be used sparingly in typical reporting.

Because drilldown functionality is expected in a dashboard, the greatest challenge can be keeping it within reasonable parameters so as not to impact performance and/or file size (if a dashboard is being distributed with embedded data this can make a huge difference on load times or whether or not it is email-able).

For this example dashboard, I have made the scoping decisions in line with what can reasonably be expected in a real business scenario where the audience is the Incident Manager and/or above.

## Scope Depth

This will be a weekly report that the Audience reviews on a Monday morning for the previous working week. This sets the volume of data to five days-worth of 'Depth'. This will also require a few older, historic data rows for previous weeks to add context. For the sake of this example, I am going to assume the audience is only interested in the Incidents that were raised during the last week in detail, but will include an addition four weeks. As we are including the Availability SLAs for the previous week, any Incidents from before that period will not be connected to any outages within the reporting period.

Later, I will look at how to handle the scope for drilldowns within a Dashboard.

## Scope Breadth

Our target area is Incident SLAs pertaining to Service Availability with supporting information such as Priority, Date(s), Resolver Groups.

# Agile Stories



In general, I am not really a fan of the Agile methodology being applied to reporting requirements. It tends to lead to requirements being unpinned by a user story.

Often with Business Intelligence and particularly Management Information reporting is that the client's 'story' is "It may be useful", "I just like to know" and so on, which leaves the client without a means to request the report.

---

These may not be the highest priority jobs, but if the client thinks they will add value and give them a better view of the operations, these reports should be developed.

# The SLA Metric Dashboard



The SLA Metric Dashboard, as the name suggests, presents a set of metrics and their level of success in a clear and easy to read manner. This is supported by one level of drilldown to get to the detail of any issues impacting the metric.

## Using the Functionality

A key consideration for dashboard development is to know what the reporting software can do. This guide is deliberately software agnostic, and may include functionality which some software cannot facilitate - and, importantly, vice versa.

Knowing what the dashboard/reporting software can do both functionally and in regard to the range of charts available can have a dramatic effect on what can and cannot be produced.

A good example here is the Pareto Chart.

Some reporting software has built in Pareto Charts, some do not and some reporting software is built in such a way that building a bespoke Pareto Chart is almost impossible.

In my opinion, Pareto Charts are overused, often in a way that is detrimental (eighty per cent of the confusion is caused by twenty per cent of Pareto Charts!), but in this particular case, a Pareto Chart is just the thing to identify which Services are generating the most Incidents. So, in this case, if the dashboard software does have a built in Pareto, the prospective audience should be made aware of it at the design stage.

## Target Date Range

The date range is not really that important for this example, as it does not fundamentally impact the appearance or functionality of the dashboard. I have chosen the weekly range, with a daily summary chart available on main page. There could just as easily be any other compatible date ranges. But please avoid trying to force weekly summaries into a monthly top view, whole weeks do not fit into months and can produce spurious results.

If monthly level data summaries are required, the drilldown can really only go to daily and retain its accuracy. Personally, I strongly suggest moving to four weekly reviews rather than calendar month; this is the only method for true and accurate reporting across comparable time periods.

## Scope Drilldown

The drilldown functionality is not particularly new, but it still has sufficient novelty for it to be overused.

A good guideline for usage would be to check if a chart requires a drilldown to a second chart without adding information (i.e. it only reformats the target in the parent chart); in this case, the parent chart is not clear enough and the second chart is a waste of space.

# The First Drilldown for the Example Dashboard



The SLA Metric Dashboard, as the name suggests, presents a set of metrics and their level of success in a clear and easy to read manner. This is supported by one level of drilldown to get to the detail of any issues impacting the metric.

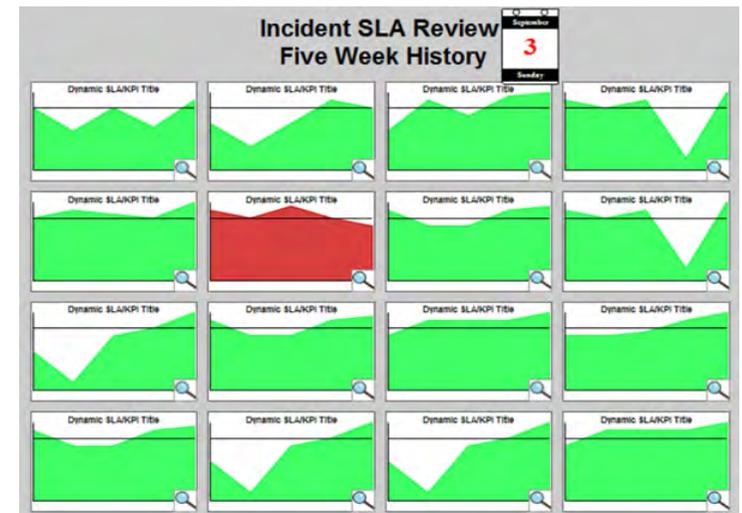
## Level One: Dynamic Contents Page

So, in a break with dashboard tradition, the object of this level is not to ram every possible chart and piece of data into the visible area. Rather, it is to be used as a dynamic contents page that clearly identifies any areas of concern.

For each SLA metric there is an Area Chart showing the previous five whole weeks alongside its KPI.

However, if the last (and therefore most recent) whole week fails to meet its SLA, the entire area of the chart is coloured red... and green if not. In our example dashboard I have not used amber (personally, I am not a fan) - an SLA has either been met or it hasn't.

Here is Level One:



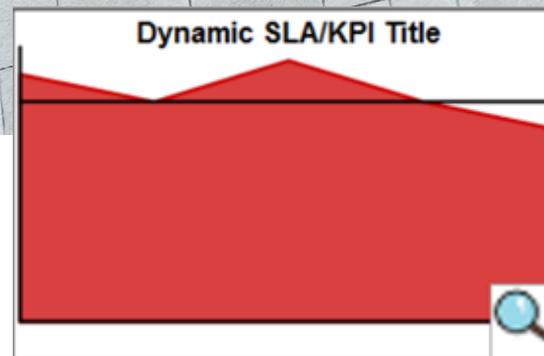
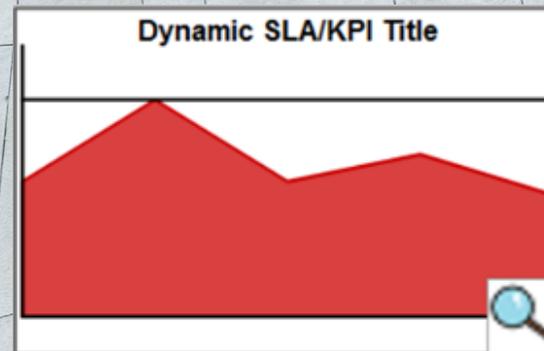
At a glance, it is easy to see where the current weaknesses are in Incident Management and the immediate history of each SLA to provide the context missing from the single bar chart solution mentioned in the previous section.

## RAG Abuse!

Colouring the entire chart red when only the latest week has failed may seem a bit unfair, or overly generous if coloured green when a metric has failed for the previous four weeks and has only passed on the latest one.

A case can be made for a simple indicator as to whether or not a metric has passed or failed, and this approach will work but I feel it misses an opportunity to pass contextual information to the audience.

Our Area Chart solution can tell the audience that while the metric has failed this week, it has improved greatly compared to previous weeks... or if a KPI is degrading over time and requires further investigation.



A simple red bar cannot convey the recent trend data. It is this same trend data that will be utilized in further levels and so has to be accessed regardless of design decisions for the 'content page' and it is just easier on the eyes when dealing with larger ITIL implementations.

## A Few Final Design Options

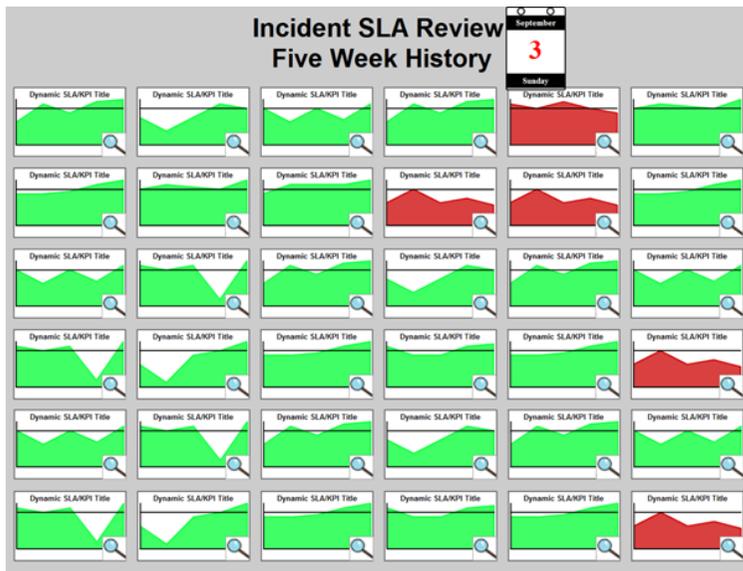
I have gone for neutral and generic colours with a simple heading, but changing the margin colour and/or including a logo in the title can go a long way to putting an organizational stamp on the dashboard.

One word of advice in this regard: please stay away from strong reds, ambers or greens as they conflict with the RAG colour scheme and can detract from its display impact.

Displaying the date via the calendar page is another example of using the specific functionality of the dashboard software, which may not be available across all commercial packages.

To illustrate this, here is an example of a more realistic version of this dashboard:

For the purpose of the Front/Contents Page, any KPI that has not been triggered within any given week is counted as green. After all, if a KPI has not been used, it has not breached!



But I've deliberately chosen this method of displaying the date as date formats vary between countries and can cause confusion for international reporting... I also think it looks cool!

From a functional point of view, the magnifying glass is not really required. Virtually, every piece of reporting software caters from drilling down through the chart itself and dashboards should be as de-cluttered as possible.

So why add a magnifying glass? Magnifying glasses have been used to make it clear what is expected of the audience and where to direct their activities. Some training is usually required for using a dashboard to its full potential, but the more intuitive we can make it, the more likely it will be used. A large benefit, I think, for a bit of screen clutter.

# Level Two: The Main Page



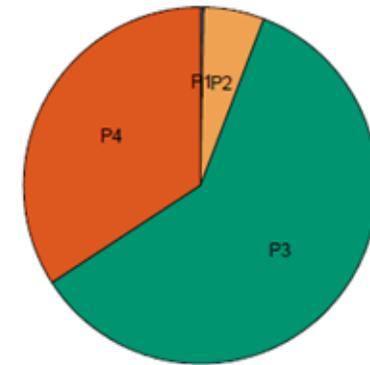
Now, we need to decide what the magnifying glass in the corner of each Area Chart leads to in the main page of the Metric Dashboard, but first, a few general design decisions made upfront which will make life easier.

Obviously, if no Incidents occur for a specific KPI within a week, charts will be empty. Practically, the audience probably won't even view a KPI that hasn't breached (and will therefore have no data in some charts) but provision should be made if they do. A dynamic message associated with each chart identifying it as deliberately empty is enough.

## The Death of Pie

The volume of data within a single chart can vary dramatically with ITIL reporting, making some charts more useful than others.

By splitting Incidents by prioritization, it is likely that there will be a lot less Priority One Incidents upon which to report than Priority Twos and Threes. This invariably leads to pie charts like the following:



To avoid this, I will simply not use Pie Charts in this dashboard, but that is not to say they should never be used.

So, what do we need?

To give a full view of the selected Incident KPI, the following subjects need to be represented:

## Chart 1: How the clicked Area Chart looks in detail

This is where we right that earlier wrong of basing the full Area Chart on the most recent RAG status by presenting each week as a vertical, RAG status bar to clearly show the immediate history.

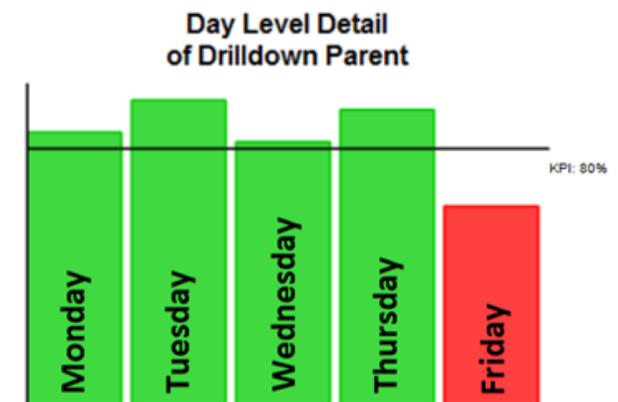
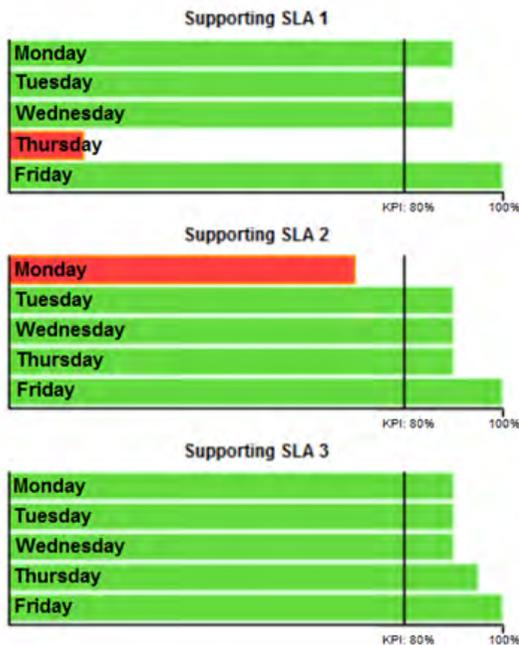


Chart 2: The 'other' KPIs

Most KPIs belong to a logical set: in this case, the SLA KPIs set for a particular service. For an ITIL service this is likely to be: Availability, Response, Resolution and Closed.

In a similar style to Chart 1 for the targeted KPI, these supporting KPIs should be displayed as one RAGed bar per week to provide overall context.

We'll make these charts a bit smaller than the above chart and use horizontal bars.



<sup>1</sup> The Closed SLA is often dropped from reporting with the Resolution SLA marking when an Incident is considered finished. This is often because getting confirmation from the Incident owner as to whether a ticket can be closed is notoriously difficult... unless, of course, the fix has not worked. So a 'no news is good news' approach is adopted, sometimes leading to Incidents being automatically set to closed after a set time period. Either way, the Closed SLA is usually not worth reporting on, but is included here for completeness.



<sup>2</sup> It is important to use the Root Cause as identified by the Service Desk or Resolver Group(s), rather than what the (potentially non-technical) ticket owner thinks is the issue, as this is often incorrect.

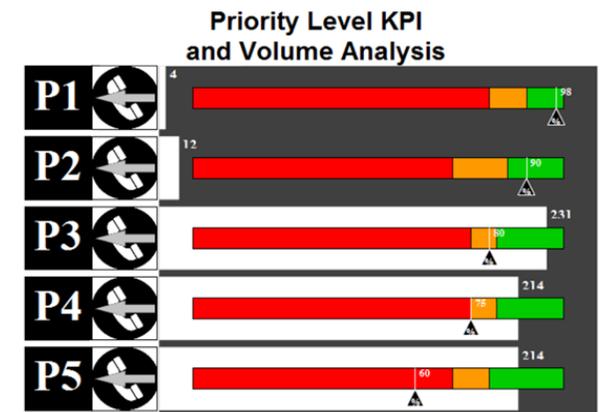


Chart 3: A priority level breakdown

As the front page works on a 'per service' basis, but SLAs and KPIs tend to be 'per service, per priority', therefore it is imperative that this information is displayed in a prominent position on the main page.

This horizontal bar chart is also a good opportunity to display the discrete volumes for each priority. Volume based SLAs should never exist for Incident Management as it is beyond the ability of the Resolver Teams to control.

This chart will only display the most recent full week and be initially linked to the chart 1, but can be redirected to any of the Chart 2s by clicking on them to save the audience having to traverse back and forth from the front page.



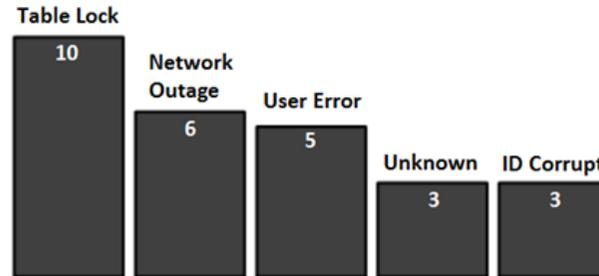
These charts all focusing directly on the SLA/ KPIs, but some supporting information is definitely required.

First, we shall add some Root Cause analysis to inform the audience what is causing the Incidents in the first place.

### Chart 5: Breached Root Causes

To support Chart 5, we will add a basic bar chart to show the Root Causes that are responsible for KPI breaches. If one or more Root Causes are responsible for numerous SLA breaching their KPI, it suggests a weakness in support.

### Top 5 SLA Breaches by Root Cause



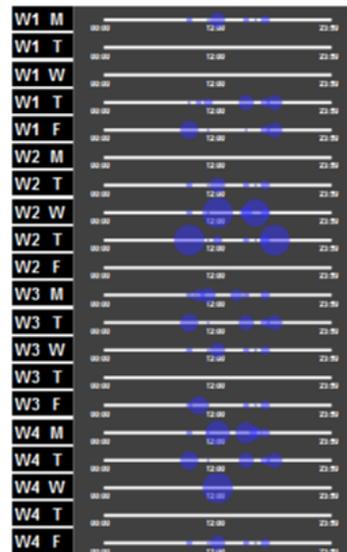
Lastly, let's add two final charts to the dashboard to provide the audience with some more context.

### Chart 6: Incident Log Times

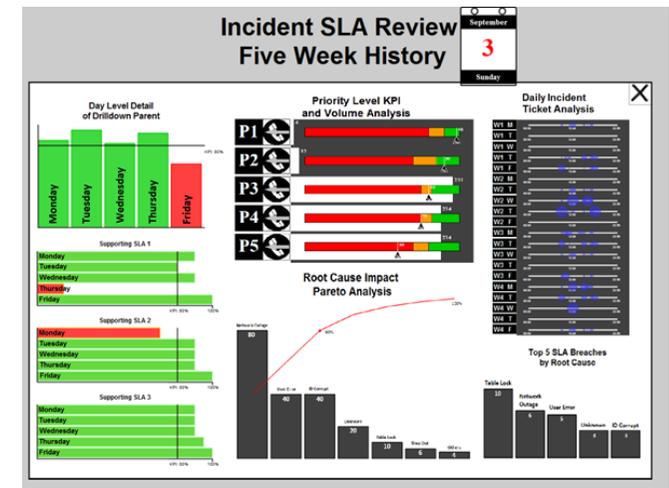
Illustrating when Incidents are registered throughout the day may not add much value in the case of single digit Incident volumes, but larger volumes may reveal important trends.

We will depart from using a bar chart and use a more exciting option available here, namely a Bubble Chart. We will keep a unified look by standardizing colours across the charts to be more aesthetically pleasing, and group the more detailed charts together, so they are distinct from the RAG bar charts, which are a legacy from the front page.

### Daily Incident Ticket Analysis

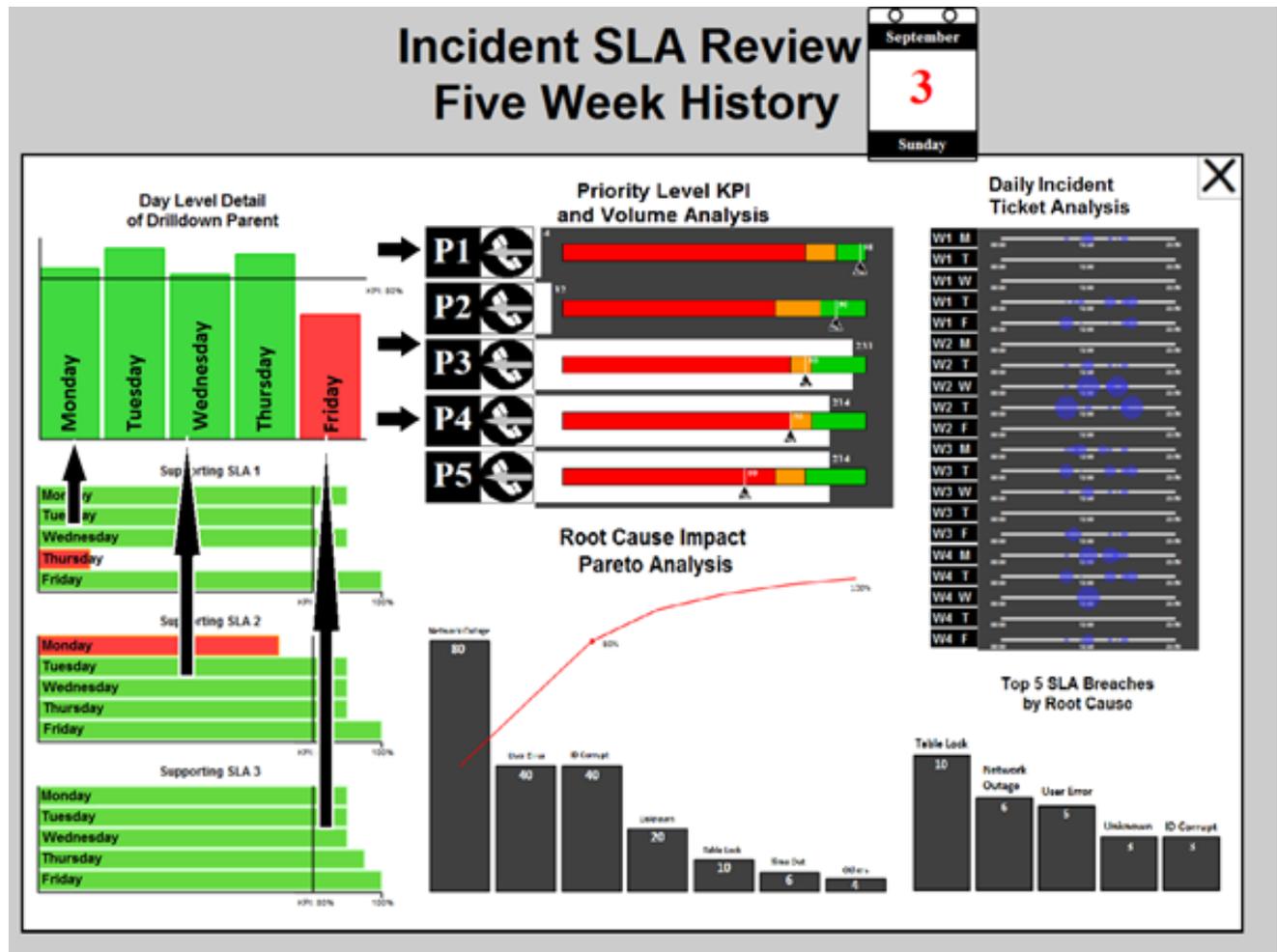


And, bringing it all together (the chart title relates to the above bullet points for reference):



## Drilldown Functionality

The following illustration shows the flow of drilldowns within the main page, with charts 1 and 2 acting as an addition index to the front page:

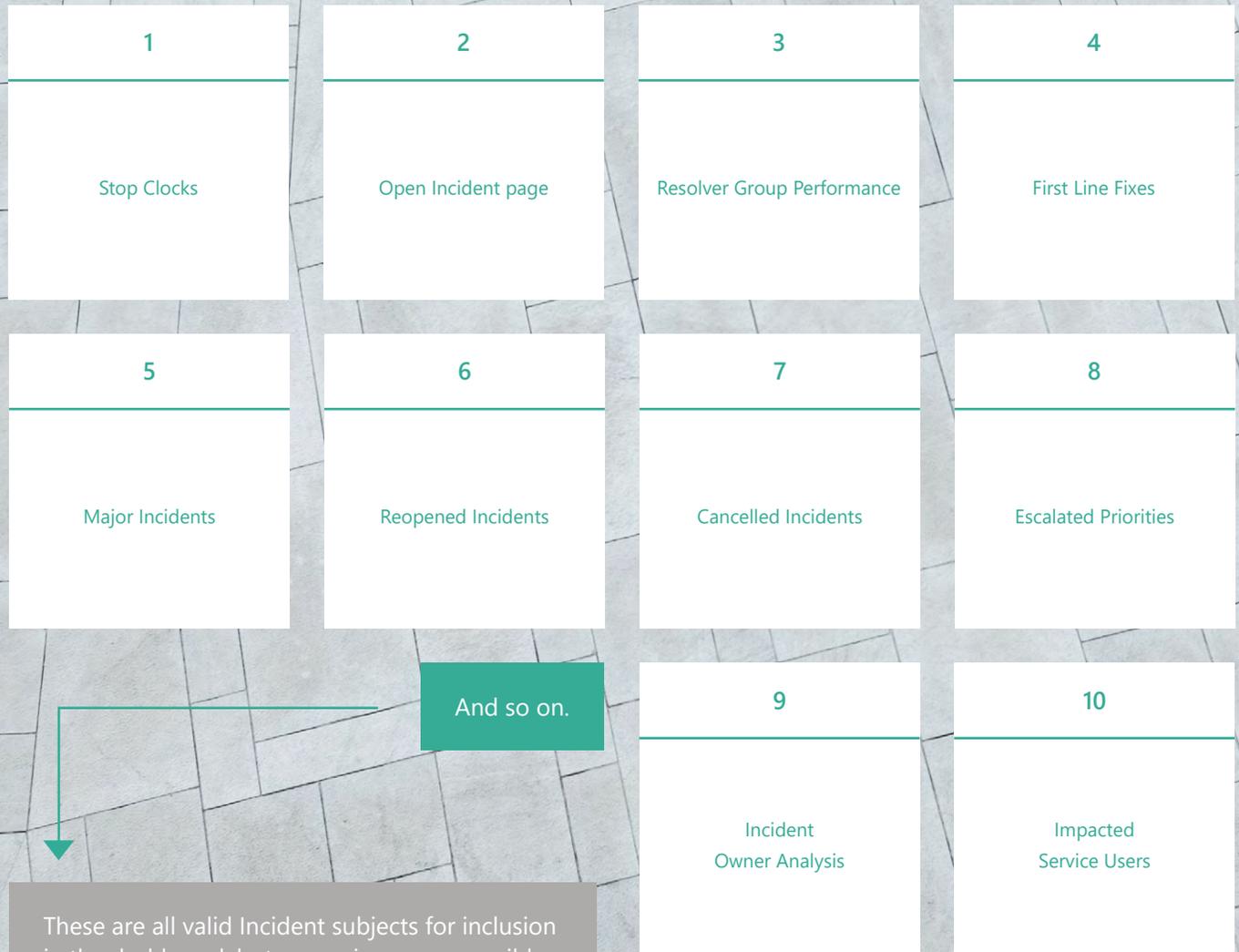


# What isn't there?



There are a host of other supporting information that could have been included in the main page which could add value to Incident Management reporting.

The list that follows contains the first ten that came to mind, but there will be more:



These are all valid Incident subjects for inclusion in the dashboard, but cramming every possible metric that can be charted into one dashboard is the thinking I am challenging in this guide.

The ideal solution is to create an integrated dashboard set that enables a full view of Incident Management metrics, but, unfortunately, this is beyond the scope of this paper.

# Hierarchy Equals Volume

As a general rule, the higher up the organizational hierarchy the Metric Dashboard audience resides in, the broader the range of Front Page metrics. In the case of this dashboard, if the audience was aimed at a Team Leader, the range of SLAs on the front page would be much shorter.

Indeed, if the list of SLAs becomes too short, the Front Page may be superfluous unless the dashboard is intended for mobile phone distribution and space is at a premium.

## Types of Metric Dashboard

The SLA flavour of this dashboard presented above is a great solution (if I do say so myself!) for monitoring the health of system support. It could easily be applied to any part of the ITIL solution that is supported by a multitude of metrics.

## Getting the Data

Populating the Metric Dashboard will differ depending upon the software being used to support the ITIL implementation, which makes it virtually impossible to provide any specifics in this regard.

The illustrations provided in this paper are taken from an actual dashboard which is fed from a CSV (Comma Separated Values) which is a format most data related software can output to and is populated by one (relatively simple) SQL statement.

# Summary



The dashboard we have created in this guide is not the answer to every dashboarding requirement, but neither is throwing a mishmash of charts on a page!

What I hope this example has done is illustrate that with some forethought and planning there is a world of dashboard opportunity that is still largely untapped.

Once unshackled from the mindset that a dashboard is one layer of charts and the use of drilldowns is limited to functional benefits, a lot more meaningful data representation can be achieved.

This dashboard enables the audience to see any failings in the overall SLA results and drilldown to a more detailed view, with each layer being clear in communicating its point, which should always be the goal with any report, including dashboards.

**Orbus Software UK London**

**Orbus Software US New York**

**Orbus Software AUS Sydney**

**Orbus Software RSA Johannesburg**

© Copyright 2016 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to:

[marketing@orbussoftware.com](mailto:marketing@orbussoftware.com)

[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com) | [www.orbussoftware.com](http://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

