cih Solutions
SERVICE MANAGEMENT INTEGRATORS

Risk reduction through proactive Problem Management

The fourth in a series of white papers from CIH Solutions that discuss topical issues in IT Service Management

Abstract

This white paper looks at how improvements can be made to the Problem Management process to drive greater business value by reducing both risk and cost to service delivery.

Mike Simpson
info@cihs.co.uk

January 2019

This white paper discusses how the Problem Management process can be enhanced to provide a more proactive role in problem resolution. At the same time, this paper looks at how problem management can perform a critical role in supporting an IT organisation's Risk Management function.

Introduction

I have a suspicion that out of all the ITIL v3 processes it is the Problem Management process that is least understood. In part, I believe this is due to an entrenched view that exists in most IT organisations that problem management is basically the same as, or subordinate to, incident management.

This is wrong, as problem management has a wider role to play. First, it is important to realise that problem management is as equally allied to risk management as it is to incident management. Where problem management deals with ***existing*** problems, risk management deals with identifying ***future*** problems.

Clearly, a synergy exists between the two functions. Why is this important? To answer this question, we must first understand the role of IT risk management.

What is IT Risk Management?

Today, more than ever, there exist many high-profile IT risks that if not handled correctly can damage the reputation of the company – I'm referring here to the security risks of cyberattacks.

If you think this might be an exaggeration, then I can do no better than quote from the 2018 Second Annual Report from the UK National Cyber Centre (NCSC)[1]. It states that:

*"Since it became fully operational in 2016, the NCSC's cyber security front line has provided support on 1,167 cyber incidents – including 557 in the last 12 months. The report reveals most of the attacks against the UK are carried out by a **hostile nation state**".*

Note that this statistic refers to a 'hostile nation state' – so we are not just talking about criminals deploying ransomware or amateur hackers going phishing (annoying as they are) – these cyberattacks are a magnitude more serious as they target UK government institutions and infrastructure.

I suspect it is not going to get much better in the future. This raises a further question – are companies prepared to meet these security issues?

"The report reveals that most of the cyberattacks against the UK are carried out by a hostile nation state".

The recent UK Government's Cyber Governance Health Check[2] found that:

'Only a **third** of the UK's top 350 businesses understand the threat of a cyberattack'

And, according to the Minister for the Digital Economy – 'too many firms are losing money, data and consumer confidence with the vast number of cyberattacks. It's crucial that businesses are secure and can protect data'.

So how does this relate to Problem Management?

As mentioned previously, problem management deals with current problems and risk management addresses future problems; i.e. they both address vulnerabilities – internal and external.
As the overarching purpose in an IT organisation is the integrity of service delivery it is critical that the key functions of IT risk management, IT security and IT systems architecture are better aligned with the function of problem management. Together, these functions will form the basis of a much stronger 'core' to identify and reduce the system vulnerabilities that can result in system downtime.

To achieve this, problem management must be re-positioned in a more central role in an IT organisation, and I will revisit this re-positioning in more detail later in this paper.

The aim, therefore, of this paper is to outline a set of improvements that can strengthen the process and move problem management into this central role and at the same time become more ***proactive*** in the way it functions. However, before we can reach that target we need to ensure the Problem Management process is operating effectively.

First, we need to agree a definition of a problem, incident and event as these terms will be used throughout this paper.

What is a Problem?

ITIL defines a problem as - *"a cause of one or more incidents."* It goes on the say *"The cause is not usually known at the time and the Problem Management process is responsible for further investigation."*

Note: Problems can be raised in response to one or more incidents, or they can be raised without the existence of a corresponding incident. For example, if an 'out-of-hours' technical change causes a system failure then it's not an incident. It's just a problem that needs to be logged and fixed before the start of the service hours. Note that incidents only occur during service hours.

What is an Incident?

ITIL defines an incident as - *"An unplanned interruption to an IT service or reduction in the quality of a IT service."*

What is an Event?

ITIL defines an event as *"Any change of state that has significance for the management of a CI."* Events happen all the time and may or may not indicate a problem; and a problem may or may not result in an incident (or incidents). The term also covers an alert or notification created by any IT service, configuration item or monitoring tool. Basically, all the other terms that we regularly use like fault, error, failure can be broadly categorised as events.

Typical Shortfalls in Problem Management

On the assumption that your organisation operates some form of problem management process (not all organisations do), then it is likely that it is not as effective as it could be. Let's start with a look at some typical outcomes of a poorly constructed Problem Management process.

First, if the Incident Management process is not closely aligned with the Problem Management process then this will result in:

- incidents being closed too early
- incidents being resolved without a problem record being created
- incidents not being linked to the correct problem record
- incidents remaining open as no resolution has been found
- incidents remaining open even when the problem has been resolved

The above list indicates that the ***problem*** to ***incident*** management processes are not correctly aligned. There are many reasons why this happens and here are six, in no order of preference:

- Roles and responsibilities are not clearly defined for problem resolution
- IT Service Desk personnel often regard 'problem' and 'incident' as interchangeable
- Problems not documented correctly by service desk personnel, if at all

- Problem detection is poorly coordinated
- Problem tracking across technical domains is poorly executed
- Inadequate root cause analysis

Steps towards proactive Problem Management

The way we set up and operate the Problem Management process within IT Service Management is fundamental to the success of service delivery.
As already mentioned, this white paper looks at how IT risk can be reduced or mitigated through a more robust problem management process and to achieve that problem management must become more proactive.

Basically, problem management can be performed in three different ways, reactive, proactive and predictive as described below:

- **Reactive** - is concerned with solving problems in response to one or more incidents
- **Proactive** - is concerned with identifying and solving problems with known errors before further incidents can occur again
- **Predictive** - is concerned with identifying future problems that have not yet impacted the business.
  Note: - although 'predictive' is not the subject of this paper, it is covered in the CIHS Briefing Note: *Applying Predictive Modelling to Problem Management.*[3]

Recommended improvements

I'm going to outline five improvements that I believe will help create a more effective Problem Management process that will lead to quicker service recovery and reduce future risks. The improvements will also ensure that a meaningful data feed can be compiled as an input to a risk model as outlined in hot spot 4.
In the improvements listed below, Nos. **1** to **3** are basically best practice; Nos. **4** and **5** are for proactive problem management with a focus on trend analysis.

Improvement 1 – create a dedicated problem management team to oversee the entire end-to-end process. This will include the role of a Problem Manager;

Improvement 2 – establish a problem record schema for compiling problem data in a consistent manner for all problem types;

Improvement 3 – improve the interface between the CMDB, the KEDB and the KMDB to build a better relationship between known errors and CIs and sets of CIs.;

Improvement 4 – adopt probability-driven RCA techniques within Problem Analysis to determine the cause of a current problem using a probability rating based on historic problem data;

Improvement 5 – introduce a small set of metrics to monitor problem management performance that can be used to aid analysis of trends.

Improvement No 1 - Dedicated Problem Management

It is often the case that problem management is not performed using a dedicated problem management team to oversee the entire end-to-end process. Instead, there tends to be an 'incident' centric approach that involves technical teams chasing down an incident, finding a quick workaround, and then move to the next incident. This leaves the underlying problem not fully understood or investigated as the problem management task is left to other support teams to follow through – after the incident is closed.

This is all understandable, but often incident management staff double up as the problem management team, especially where there are insufficient staffing levels. This effectively reduces problem management to a subsidiary role when in fact it should be the lead function. How best to achieve this?
The most effective solution is to appoint a full-time problem manager with a dedicated team who will act as a focal point for bringing together all the various tasks needed to identify, resolve and prevent problems. Figure 1 shows our target organisation for problem management.

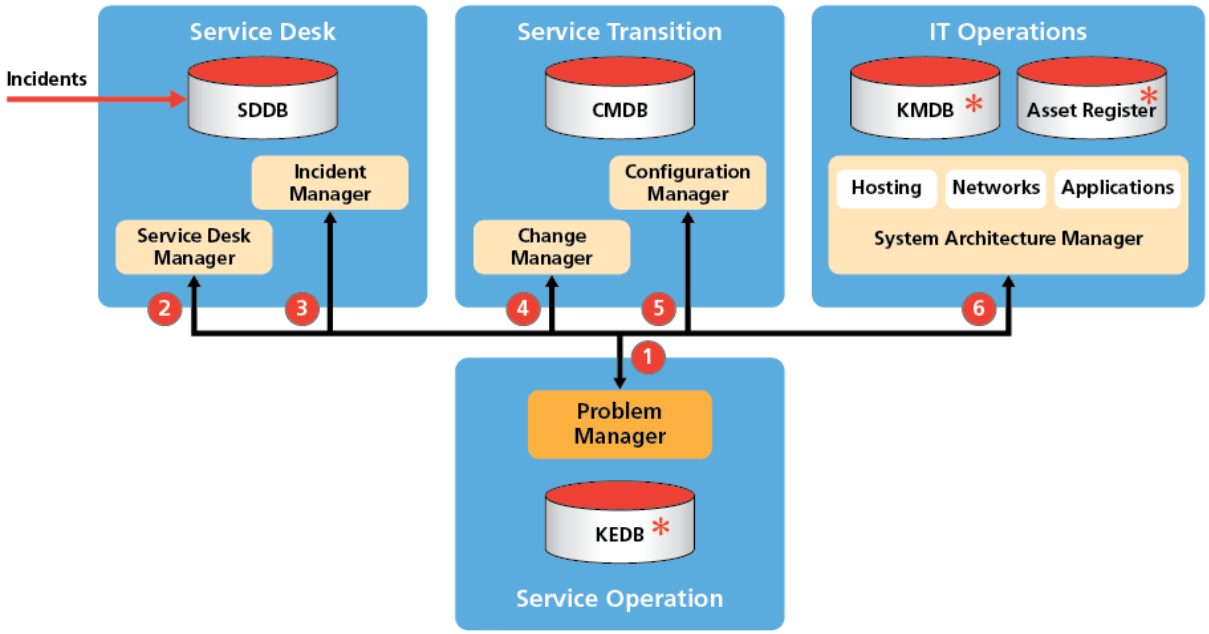

Figure 1 – Target Problem Management Organisation

*Note: Whilst the KEDB, KMDB and Asset Register are shown separate, it is likely that they are one database.

Overview of Key Contact Points

1. **Problem Manager**
   For a problem manager to take control of the life cycle of a problem there must be appropriate controls in place to ensure that problem resolution is managed effectively.
   The problem manager will take control of each problem and will be responsible for coordinating all aspects of problem analysis and resolution working with the incident management team and technical support staff. This includes the ownership of the KEDB and problem records.

2. **Service Desk Manager**
   The Service Desk is usually the first point of contact for the logging of an incident. The problem manager will rely on the service desk personnel who perform a First Line service function. If the incident is difficult to clear or if it is cleared but not the underlying problem, they will invoke the Problem Management process.

3. **Incident Manager**
   The incident manager must involve the problem manager during an incident if the incident cannot be matched to known problems and looks like leading to a major incident. The problem manager will invoke the major incident escalation procedure should the impact become severe and the time to resolve the problem will impact service levels.

4. **Change Manager**
   The change manager and problem manager will work together to implement changes that have been identified during problem analysis, to reduce or prevent problems re-occurring. This may involve an emergency change request (ECR) and one of the responsibilities of the problem manager will be to sit on the Change Advisory Board (CAB) to oversee the implementation of all emergency changes.

5. **Configuration Manager**
   The problem manager must liaise with the configuration manager for two important reasons. First, during problem diagnosis the CMDB will be the main repository for identifying the CI(s) that are involved with the underlying problem. Second, after problem resolution the CMDB may need to be updated with changes to a CI record.

6. **System Architecture Manager**
   During problem diagnosis, the problem manager will need to consult the system architecture manager and Subject Matter Experts (SME) to track the root cause of a problem. The SME will conduct a range of infrastructure tests that cover servers, storage, networks and applications and provide updates to the problem record.

Improvement No 2 - Problem Record Schema

In the first improvement we elevated the Problem Management process to a more central role with closer liaison with the Incident, Change and Configuration Management and IT Operations teams. A key component in the way the organisation communicates is the problem management record. Now, problem records are opened by the Service Desk either at the time an incident is logged or sometime after – and sometimes not at all, if an incident is cleared quickly. Problem records can also be auto-logged when a major incident (Severity 1) is initiated, depending on the functionality of the service desk.

Also, when problem records are raised the initial information like problem type, service impacted, urgency etc. is inputted but is often not completed correctly once the incident is cleared. To some extent this is understandable as service desk operators have by now moved on to the next incident.

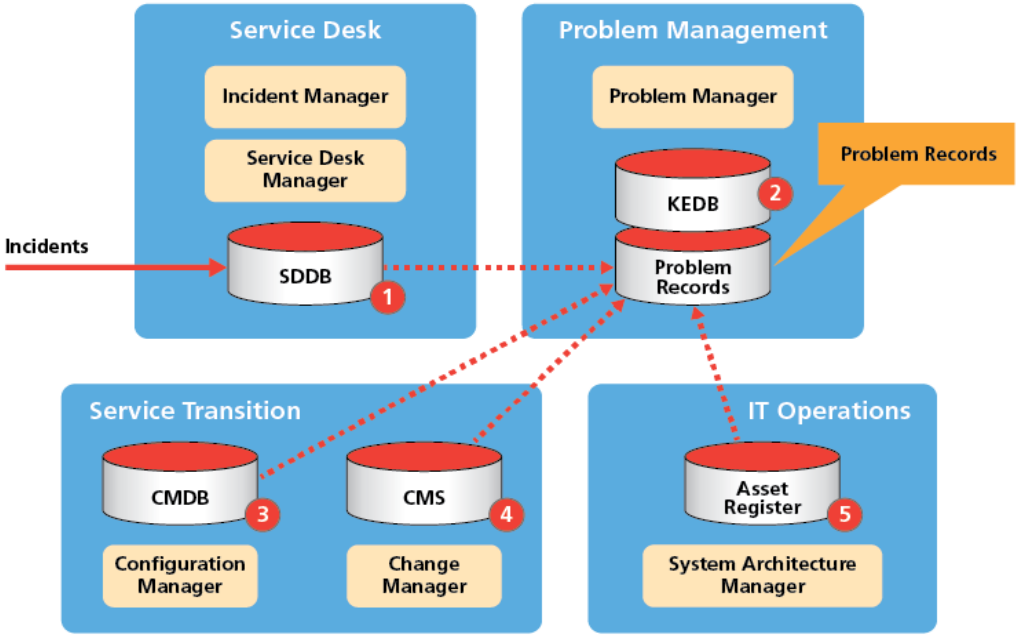

Figure 2 – Data Sources for compiling a Problem Record

Figure 2 above shows the main sources of data that we need for our problem record schema. The list is only indicative as the problem management team will design the schema according to the scope of the IT system and third-party involvement. That said, the table below shows most of what needs to be collected to create a meaningful problem record.

	Source	Examples of Problem Record Content
1	SDDB	Incident type, service impacted, severity and urgency
2	KEDB	Association with known errors, previous CIs impacted, previous resolutions, associations with other incidents, changes, root causes and workarounds
3	CMDB	CIs and groups of CIs and dependencies impacted by the problem. Recent changes to CIs. Attributes updated due to a change to resolve the problem
4	CMS	All change requests and emergency changes that are logged by the Change Management System (CMS) and their impact on the CIs
5	Asset Register	Warranty, obsolescence, end-of-life issues involved in the problem

All problem data, regardless of how insignificant it might appear at the time, must be collected to build a full understanding of where vulnerabilities exist in the IT system. Minor problems can (and do) lead to major problems, although not fully understood at the time.

Now, the problem manager can complete these fields in the service desk database, but this can lead to complications about data ownership and access. Far better, in my view, is to create a problem record as a segment of the KEDB and allow ownership and control to reside with the problem management team; thereby providing ***'consistency'*** in the problem record data and managing the life cycle of a problem.

This is the important difference between a problem record and a Known Error (KE) record. The problem record will record the entire life-cycle of a problem whereas the KE record will focus on the root causes, workarounds and the CIs involved with each error. Two important notes on the above:

Note 1 – it is possible to have more than one KE record for any given problem.

Note 2 – comprehensive problem records are needed to 'underpin' an efficient problem management process.

A problem record will also help with our overall objective of proactive problem management by using better analysis to drive IT system improvements.

Improvement No 3 – Interface between the CMDB and KEDB

To get the most use from a KEDB, the content should be shared across several functions as described under Improvement No 1. The benefits of a well-managed KEDB are faster problem resolution times, leading to reduced service downtime. There are several key data interfaces that the KEDB must have to ensure that the KEDB is going to provide the maximum value to problem resolution. The most important of these is the CI data source as this will link a problem to all the underlying CIs and associated CIs that are identified in the root cause of a problem.

The configuration items that go to make up an IT system can have complex relationships. For example:

- as a stand-alone CI with no dependencies
- as a stand-alone CI with many dependencies
- as an over-arching CI that has sub-ordinate CIs
- as a component of something else – another CI
- as a member of a cluster of CIs

It is critical that all these relationships are understood and reflected in the CMDB. It is also critical to the integrity of the KEDB for these relationships to be 'mapped' across to each of the KE records. To ensure that the known errors and associated workarounds are mapped to the correct CIs and associated CI parent/child groupings a **key data** selection and transfer should be set up. More than one CI record may be involved.

Mapping Known Error Records to CI Records

A KE record will hold details of previous incidents, for example known workarounds, incident count, root cause and is used for incident matching during the initial diagnosis of a new incidents. This will allow faster resolution when the same incidents recur.

Note: Only the problem management team is authorised to raise a new KE record, although other parts of the IT organisation can update and amend. This is critical to avoid a duplication of KE records. Figure 3 gives a sample of the type of fields that are needed for both a for CI record and a KE record.

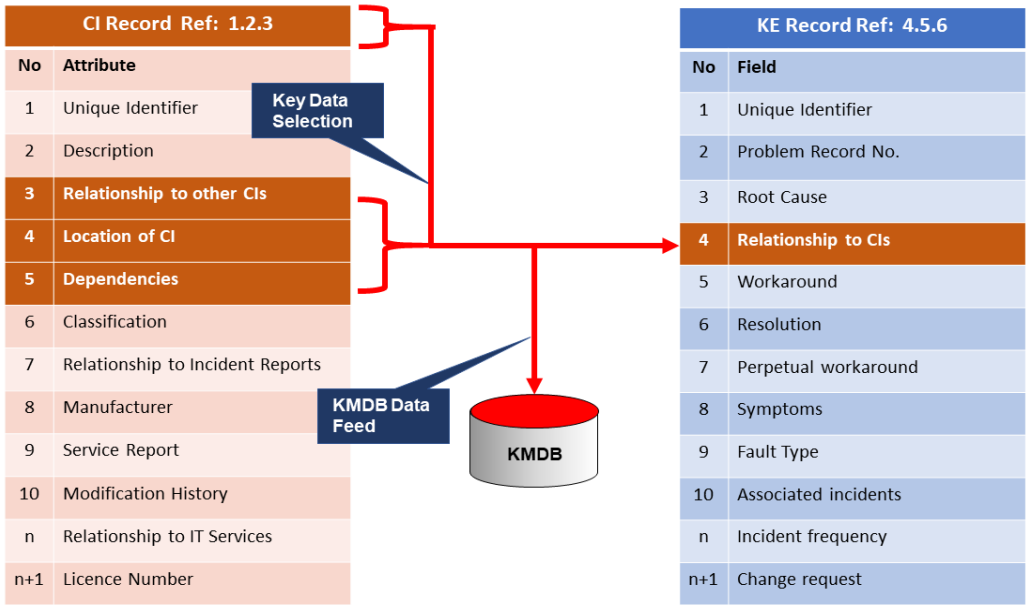

Figure 3 – Mapping CI and Known Error Record Fields

Each KE record will hold details of the CI or groups of CIs involved with a problem. Note, that it is only necessary to capture and transfer the data from CI records that are 'required' to complete the KE record, not the entire CI record content. This can be set up to automatically transfer when initiated by the problem management team. This assumes that both the CI record schema and KE record schema are consistent, which of course they must be.

Once complete the KE record is 'fixed' and date/time stamped to document the actual CIs involved with the problem at the time of the problem resolution – workaround or permanent solution - not after the event. This is important as CIs and CI relationships will alter over time. At the same time, the CI data must be sent to the KMDB to update the CI schemas that are compiled as part of the IT system architecture topology. See CIHS White Paper – *Knowledge Management within ITSM.* [4]

Improvement No 4 – Enhancing the Problem Analysis Process

Problem analysis is central to problem investigation and diagnosis. It is a wide-ranging process that encompasses several diagnostic techniques centered around Root Cause Analysis (RCA). Before we examine different types of RCA in detail, I want to look at the overall Problem Analysis process.

Figure 4 below shows a schematic of the Problem Analysis process that is triggered by the detection of a problem.

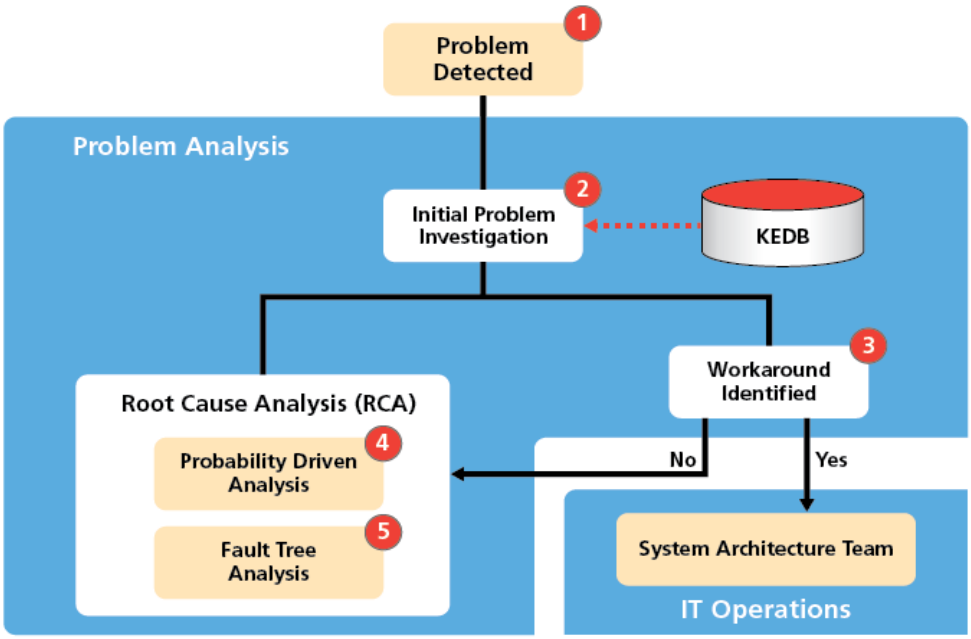

Figure 4 – Problem Analysis Process

1. There are several ways a problem is detected. They may arise from customer calls to the Service Desk or automated detection of problems from system monitoring tools. This is **reactive** problem management. Analysis of historical incidents is usually an ongoing investigation and is based on trends and may result in a problem being identified before an incident is logged. This is **proactive** problem management. Once a problem has been detected all details of the problem are entered into a new problem record.

2. Once the problem has been logged then an investigation kicks off. This first task is to check what changes have been made to the IT systems as these may be the cause of the problem. Next is to look at the KEDB to see if similar problems have been logged before. If they have then this could result in early resolution and closure of the problem by using a proven workaround or permanent solution, like replacing a CI.

3. If the problem is not cleared immediately, then the analysis takes two separate paths. One path will continue to look at finding a workaround to clear the problem, the other will commence Root Cause Analysis (RCA). If a workaround is found then the system architecture team will proceed to put this is place, supported by an Emergency Change Request (ECR) if needed.

   Note: Once a workaround is identified and implemented, the problem record must remain open so that the details of any permanent solution can be recorded. Also, multiple workarounds may be needed to resolve the problem and that may impact the SLA as each workaround could increase the service downtime.

4. If no workaround is found, then the focus must shift to determining the underlying cause of failure using RCA. There are numerous RCA techniques available, like Ishikawa diagrams and 5-Why, but many are not suitable for use in an IT environment. As this paper is primarily about reducing risk using proactive methods, I will look at two types of RCA that are both probability based – **probability-driven** analysis and **fault tree** analysis.

Root Cause Analysis

Before we look at the two RCA techniques, it's worth remembering that root cause is a very broad concept and it is possible to arrive at different root causes for the same problem. If this happens it may be advisable to adopt another RCA technique to resolve the underlying cause. In deciding which RCA technique to use, there is no ITIL standard on which to base this decision, only trial and error and personal experience.

Probability-driven RCA

Basically, this technique is based on the analysis of 'accumulated knowledge' on how the IT systems are designed and operate. Hence, the success of this method depends entirely on the quality of the technical data held in the KEDB, KMDB, CMDB and the Asset Register. Importantly, it also depends on the knowledge and skill of the IT personnel managing the environment and this is critical to the success of proactive problem management.

Figure 5 below shows a simple schematic of the analysis, based on the likelihood (probability rating) of where a problem resides within the IT system. At first glance this appears to be just common sense or intuitive diagnosis – and it is just that. However, the big difference here is that this we are providing a framework that gives structure and discipline to the analysis.

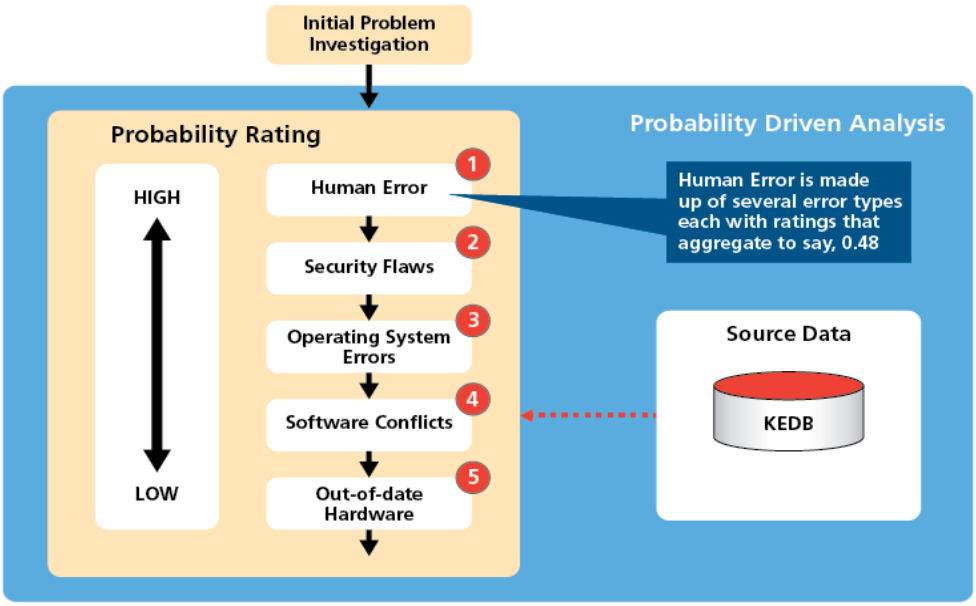

Figure 5 – Probability Driven Analysis

The philosophy behind this approach is quite simple - events do recur, frequently and they often occur in the same part of the IT system. Our starting point for creating the probability rating, is of course, the KEDB. But before we look at that its worth mentioning the recent survey by the ITIC.[5] The ITIC conducts this survey annually. The main reasons for IT system failure in 2018 are in order of magnitude:

1. Human error
2. Security flaws
3. Operating system errors
4. Software conflicts
5. Out of date hardware

I can certainly understand why these reasons are in the top five. Most of them appear in every IT organisation to some degree and would therefore factor in a probability rating.

In Figure 5, actual ratings would be determined by historical analysis of the KEDB. This would include classification of all the problems into groups by failure type. Under each group there will be further classification. For example, Group 1, Human Error, would contain other types of known human error, e.g. problems with patch management. Each type will have its own probability rating – between 0 and 1. These then aggregate to a higher rating (say, 0.48 for Human Error). This means there is a 48% chance that one of the human error types is the cause of the problem. During analysis these ratings are matched to the problem characteristics to give an overall probability as to where the problem resides.

Once this is done then the analysis of every new problem will follow this the sequence. Whilst this is not foolproof, it does provide a good framework to start a diagnosis.

Fault Tree Analysis

The second of the probability-based RCA is called Fault Tree Analysis. At the outset I would say this is a difficult analysis to apply across an entire IT system. Basically, fault tree analysis is a top-down, deductive analysis, using Boolean logic which visually depicts a failure path or failure chain of lower order components. The components in this example are, of course, the CIs. See Figure 6 below for an example of a fault tree.

The reason this is difficult is that the analysis is based on the probability of a component failure. Now, whilst failure of hardware can be represented by MTBF, the same cannot be said for software. Software differs from hardware in many respects. The number of bugs in an application is not an indication of the reliability, so unless a software vendor publishes reliability figures it's best to leave software out of the model. Clearly, a limitation.

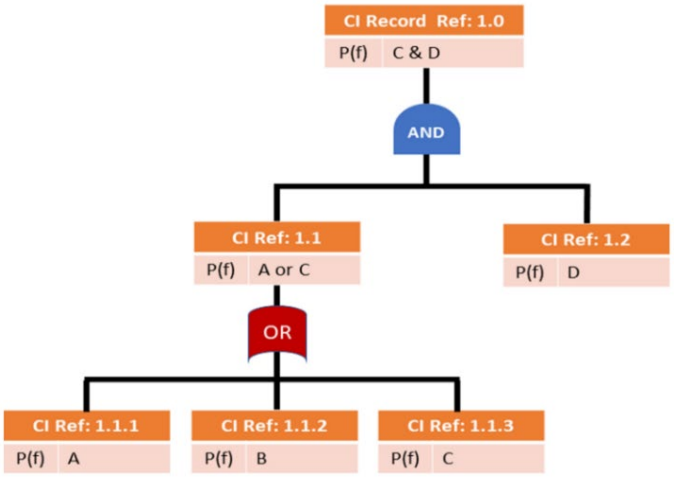

Figure 6 – Fault Tree Analysis

However, it is possible to apply this analysis to just 'sections' of the hardware infrastructure. To do this we would need to assign a probability factor P*(f)* to each of the hardware CIs in that section. The values most commonly used when calculating the level of reliability are MTTF (Mean Time to Failure) or MTBF (Mean Time between Failures) depending on type of component or system being evaluated. Once the P*(f)* for each component is known, they can be added as attributes to the CI records. The Boolean logic represents the relationship and so the probability of failure gets aggregated to the top CI.

For a computer system, the weak points are usually the mechanical parts – for example hard drives and power supplies on servers. The Fault Tree Analysis looks at individual components and the likelihood of failure of one or more CIs by aggregating all the probabilities in a branch. We can calculate the overall probability of failure of a CI by looking at the weakness of the underlying CIs. As I mentioned, it is not an easy exercise to collect reliability figures from vendors, some do, and some don't. That said, I think it is still worth investigating.

Improvement 5 – Problem Management Metrics

The correct use of metrics in the ITSM environment is an excellent way to monitor both service delivery improvements and, conversely, a deterioration in service delivery. Although metrics by their very nature lag events, they are useful in spotting the direction of a trend. Once a threshold has been crossed that can be the trigger to intervene.

I also believe that the single biggest mistake IT management make with metrics is to choose too many. With metrics, it really is a case of less is more.

A careful selection, well managed for data integrity, will give more meaningful results than a vast array of metrics that are inaccurate and poorly understood.
For the Problem Management process, I suggest just five metrics. These are:

Metric 1 – Incident Repeat Rate
Metric 2 – Problem Resolution Rate
Metric 3 – Problem Workaround Rate
Metric 4 – Problem Re-Open Rate
Metric 5 – Customer Impact Rate

The above list is just my personal selection and will focus on the volume of repeat problems that cause further incidents, when it was thought a problem had been cleared. We also need to know the impact on the SLAs. The results are expressed as a set of ratios that can be compared against benchmark ratios set by the IT management team.

These metrics are Key Performance Indicators (KPI) on how effective the Problem Management process is at managing problems. All the data needed for these five metrics will be available from the Service Desk, problem records and the KE records. Figure 7 below shows the data capture points for calculating the metrics over an agreed monitoring period – typically weekly.

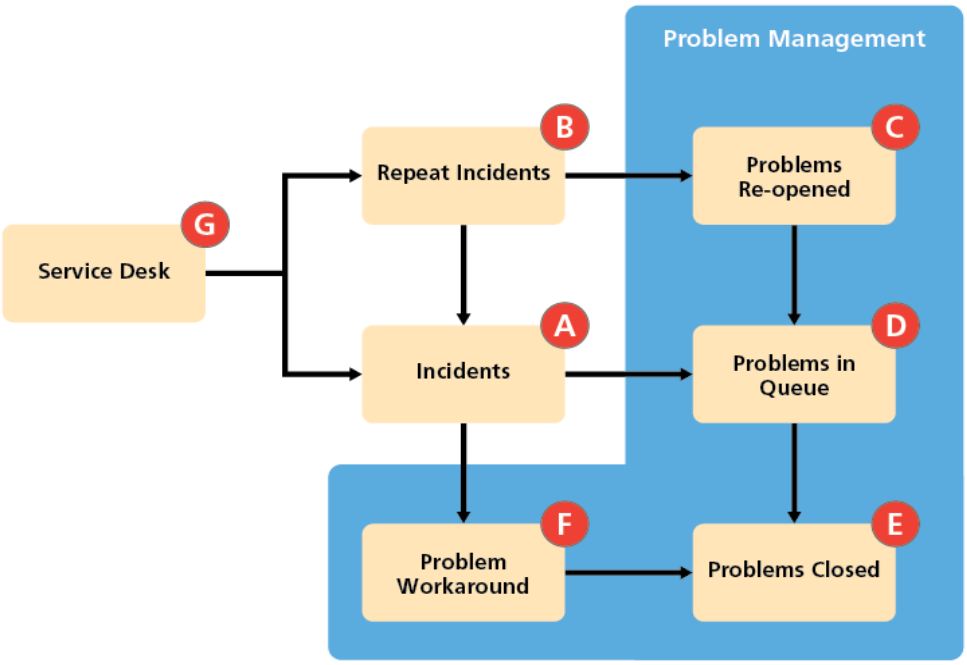

Figure 7 – Data Capture Points

Data Capture Points

A = Number of Incidents
B = Number of Repeat Incidents
C = Number of Problems re-opened
D = Total Number of Problems in the queue
E = Number of Problems Resolved
F = Number of Workarounds
G = Number of Problems with SLA impact

Calculating the Metrics

Once the data has been collected the metrics can be calculated using the simple formula shown below.

Metric	KPI for Problem Management	Calculation
Incident Repeat Rate	What percentage of all incidents are repeat incidents	(B ÷ A)100%
Problem Resolution Rate	What percentage of problems were cleared over the monitoring period?	(E ÷ D)100%
Problem Workaround Rate	What percentage of workarounds were there for the number of problems over the monitoring period?	(E ÷ D)100%
Problem Reopen Rate	What percentage of workarounds resulted in a problem re-occurring?	(C ÷ F)100%
Customer Impact Rate	What percentage of problems impacted Level 1 and 2 of the SLAs?	(G ÷ D)100%

Example Hot Spots

I've identified four typical hot spots where our 'enhanced' Problem Management process can add additional value and reduce risk. Figure 8 is based on a simplified ITSM organisation that could be either a MSP dedicated to external clients, or an ITSM organisation providing IT services to an internal client. The IT Operations can be either internal or external hosting with or without applications support. It assumes that all key components are under the control of the IT organisation.

For this paper it is assumed that the IT Operations is in-house and provides hosting, communications and applications support - within an overall governance framework. There are four examples of hot spots shown in Figure 8 and all show how the 'improved' problem management process can reduce risk in each.

Incident Management – improved linking of incident records with problem records to aid service recovery;
Configuration Management – better integration between the Problem Management process and the CMDB to reduce risk to service delivery;
System Architecture – use of historical data on problem types to improve future IT system design;
Risk Management – use performance data to drive a risk model as part of the Risk Management process.

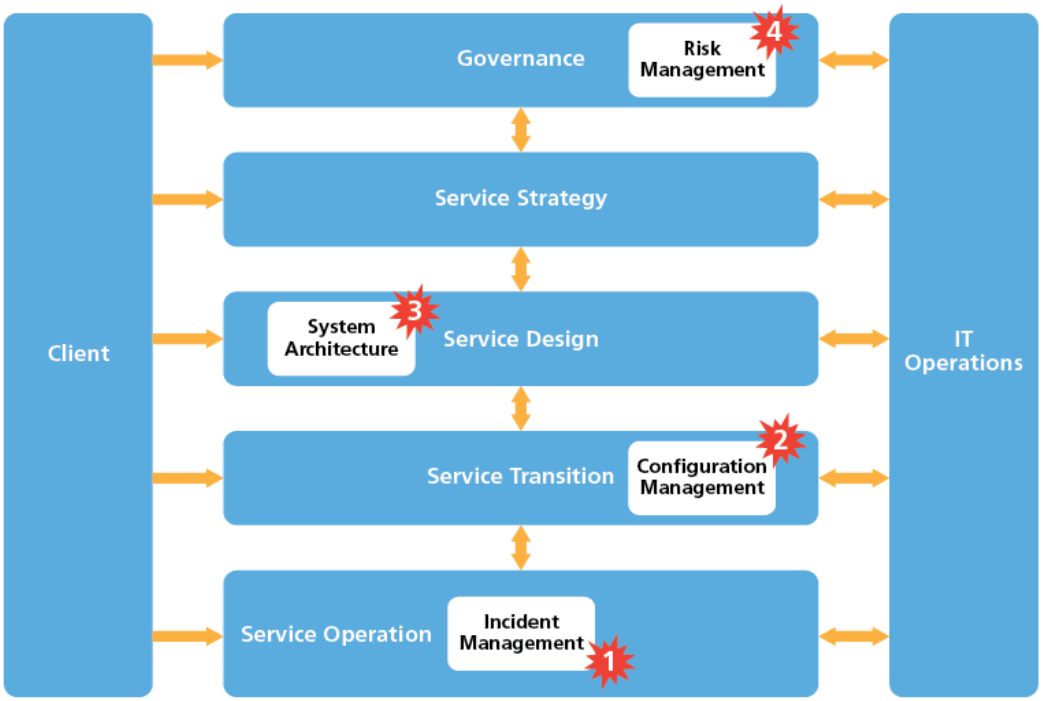

Figure 8 – Hot Spots

Hot Spot 1 – Incident Management

In the first of our hot spots I want to take a closer look at the relationship between incident management and problem management. For the problem management organisation outlined in Improvement No 1 to operate effectively, there are a few guiding principles that must be followed. These are:

- Problems must be tracked and managed separately from incidents
- Problem Management and Incident Management processes are parallel activities
- Problems can be raised without incidents being logged
- Problems can remain open after incidents are closed
- Problem records may be opened (initially) by incident management but only closed by problem management
- All Incidents remain the responsibility of the Service Desk until resolved
- All problems remain the responsibility of the problem management team until resolved

Figure 9 below shows a simplified schematic of the Problem Management and Incident Management processes and the main touchpoints where the two interact. As these are both complex processes only the main touch points are discussed.

One or more incidents that are logged by the Service Desk usually indicate an underlying problem and the Service Desk will at this point kick off a new Problem Record **(1)**.

Note: As most service desk platforms contain a problem record database it is possible that this could be used by both the incident and problem management personnel, however, I believe it is preferable to have a separate problem record database as part of the KEDB as outlined in Improvement No. 2.

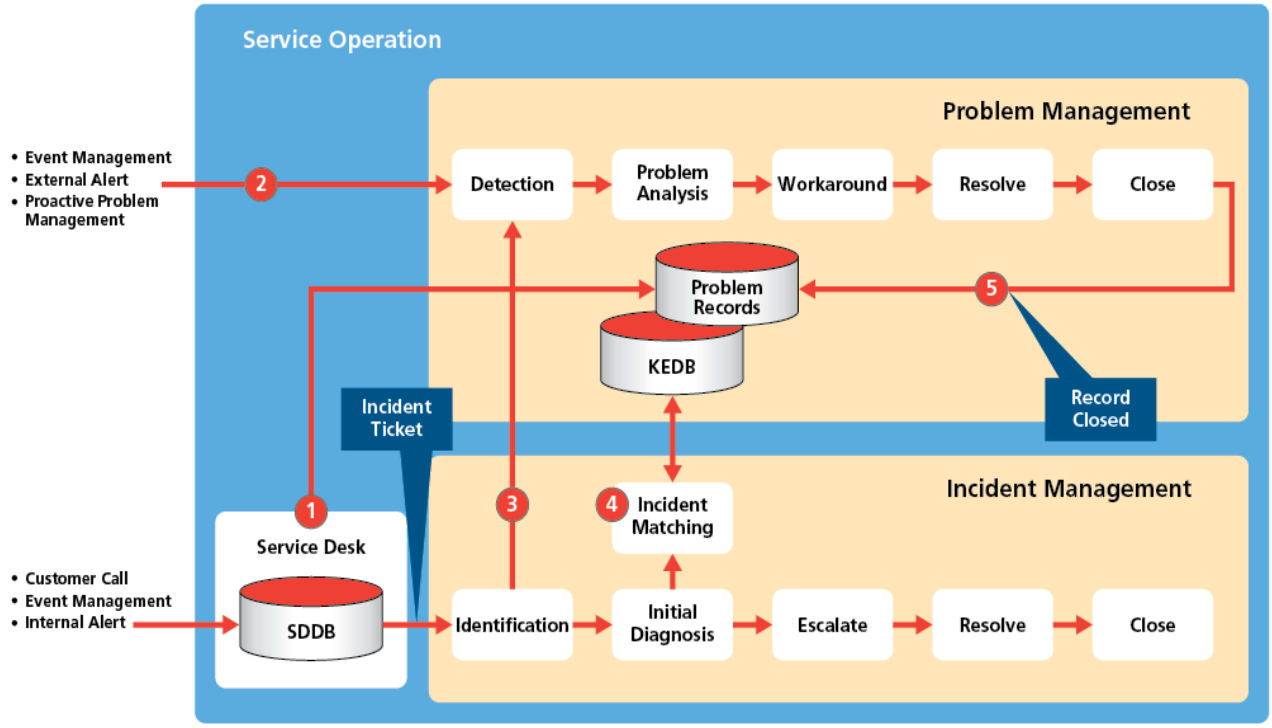

Figure 9 – Key Touchpoints for the Problem and Incident Management Processes

A problem can be detected directly by the problem management team **(2)** and that also results in a new problem record being raised. The same underlying problem may (or may not) result in an incident being logged with the Service Desk – e.g. the failure of an overnight batch job (a problem) that is fixed before the service day opens. Hence, no need to raise an incident as there is no impact on the customers.

Sometimes the Service Desk identifies **(3)** an incident that is obviously a major problem, in which case Problem Management and senior management are alerted so that the escalation procedure can be implemented whilst the Problem Management and Technical teams work to resolve.

Initial diagnosis **(4)** is undertaken by the incident management team to determine if an incident has occurred before – 'Incident Matching'. This is performed using the KEDB that holds details of all previous incidents, root causes and workarounds.

The incident management team can search against the KEDB using symptom and fault type matching. This can result in the problem being shut down quickly in which case the incident is closed, or a proven workaround is used. In both cases the problem management team will keep the problem record open until the root cause of the re-occurring problem is resolved.

If it becomes clear that the incident has no immediate solution, then the problem management team will continue to investigate the problem and keep the Service Desk informed so they can continue to monitor and advise the customer. Once a problem is permanently resolved then the problem management team can close the problem record **(5).**

Note: not all problems have a solution as these may be systemic to the underlying IT architecture. In such cases the problem management team will communicate all permanently open problem records to the IT architecture team for further investigation in future system upgrades.

Hot Spot 2 – Problem Management and CMDB Interface

This hot spot looks at the important relationship that exists between the Problem Management and Configuration Management processes - KEDB and the CMDB. In Improvement No 3 we discussed the schema of a KE record and the need for all the CIs involved in a known error to be recorded after problem resolution – either by a workaround or permanent solution.
Figure 10 shows the sequence that will lead to all the impacted CIs being captured and input to a KE record.

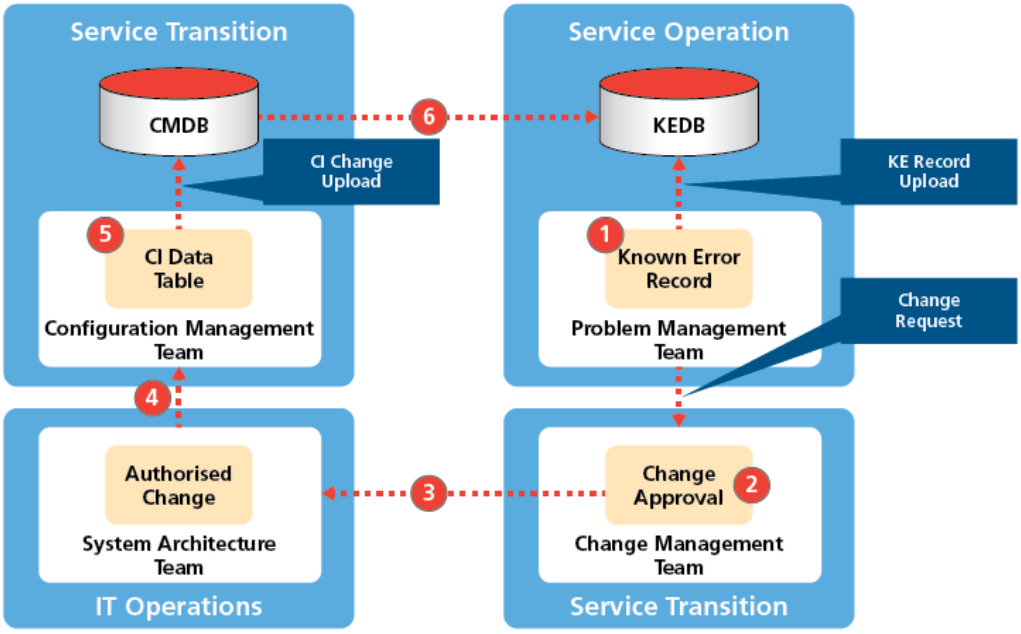

Figure 10 – Problem Management and CMDB Interface

The sequence begins with the problem management team creating a new KE record and uploading this to the KEDB **(1).** After the initial problem identification, analysis and workaround has taken place (as described in Hot Spot No 1) both the problem and incident management teams will know which of the CIs are involved in the problem. These will be entered manually by the problem, incident or technical management teams.

However, this is only part of the final process. If the problem has been resolved – either by a workaround or permanent solution - then it is highly likely that a change(s) to the CIs involved are needed. For example, this may be a firmware upgrade, a CI swap out or a change in CI configuration.

These changes must be submitted **(2)** formally to the change management team using either the Emergency Change Request (ECR) or Request for Change (RFC) process; but in conjunction with the technical teams who are implementing the change. Once approved, **(3)** the technical team will implement the change.
The change is then implemented and tested to confirm that the problem has been resolved.

The CI change details are now sent **(4)** to the configuration management team to create a CI Data Table for uploading to the CMDB **(5)**. The CI attribute data can now be extracted **(6)** for input into the KE Record.
(See CIH Solutions white paper – *The Myth of CMDB*) [6].

Hot Spot 3 – Use Data on Problem Types to Improve System Design

In keeping with the theme of this white paper – how to become more proactive in problem management – I want to look at how we can best use the data accumulated from the Problem Management process to help guide future improvements to an IT system. It is well known that IT systems tend to deteriorate over time. This is due to several reasons:

- software version conflicts, that were never anticipated by vendors
- aging servers – obsolete or end-of-life
- cache memory faults worsen over time – an increase in both hard and transient faults.
- patch management deterioration
- out of date warranties – leading to firmware releases being withheld

This deterioration will result in both repetition of known problems and problems not encountered before. Over time these will be captured as part of the problem management process, logged and stored within the problem records and the KEDB. We can then group the data into several categories, for example – problems by frequency of re-occurrence, increasing severity, occurrence of new problems and location of problems within a CI or group of CIs.

These groupings can be used by IT System architects as input to base future system upgrades. This will reduce the risk of future system failures and service delivery outages. In terms of presenting these data I would suggest a visual method rather than a spreadsheet.

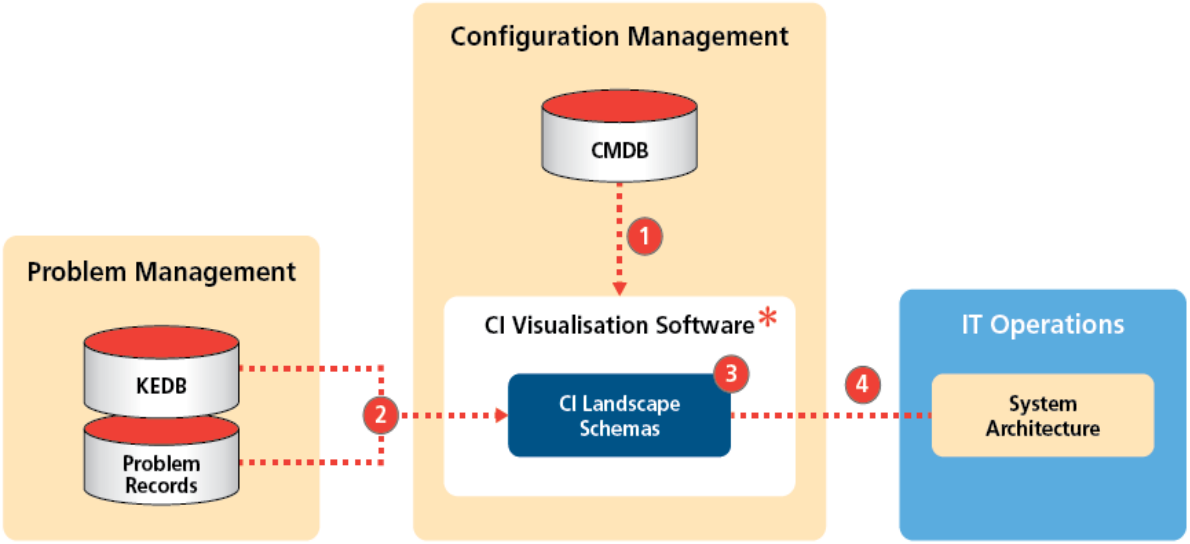

Figure 11 – Mapping Problem Types to the CI Landscape

One possible way is to **map** all the problems and the underlying known errors onto a CI schema of the IT system. The advantage being it is a lot easier and intuitive to view a visual representation of all the CIs as a series of schemas with an overlay.
To achieve this, we first need to map the location of all the CIs in the architecture landscape to create a set of interconnected schemas that visualise all the CI relationships held in the CMDB. Vendor software is available to do this, and Figure 11 shows a simple schematic of how this could be arranged.

The first step **(1)** is to extract all the CI and CI dependencies from the CMDB into the **CI Visualisation** software package and create the landscape of schemas based on CI groupings. For example, all the network CIs will be grouped as a set of schemas, in the form of a hierarchy. This is applied across the whole IT system architecture.

Once we have the schemas the data set **(2)** can be imported into the CI Visualisation package **(3).** The data will have already been codified and sorted to reflect different problem types, for example:

- problem type by frequency (occurrence)
- problem clusters
- recurring problems
- new problems
- problems introduced by changes

Depending on the choice of CI Visualisation software the schemas can be configured to show different problem types using different codes, both as icons and as colour. In addition to the visual aspect, data can be extracted **(4)** so that system architects can investigate the significance of the data in terms of possible system changes.

Hot Spot 4 – Use Problem Data to drive a Risk Model

For the last hot spot, I want to return to the start of this paper, where I emphasise the importance of aligning problem management with risk management. I also refer to the existence of both internal and external vulnerabilities. In Hot Spot No 3 above, I mention how IT systems deteriorate over time and the collection of problem data can be used to guide future upgrades.

However, there is another aspect to this deterioration and that is the opening of internal weaknesses (i.e. internal vulnerabilities) for exploitation by external threats. For 'external' this means cyberattacks. This is where the key functions of IT risk management, IT security and IT systems architecture must be aligned with the function of problem management. Figure 12 below shows how these four functions connect.

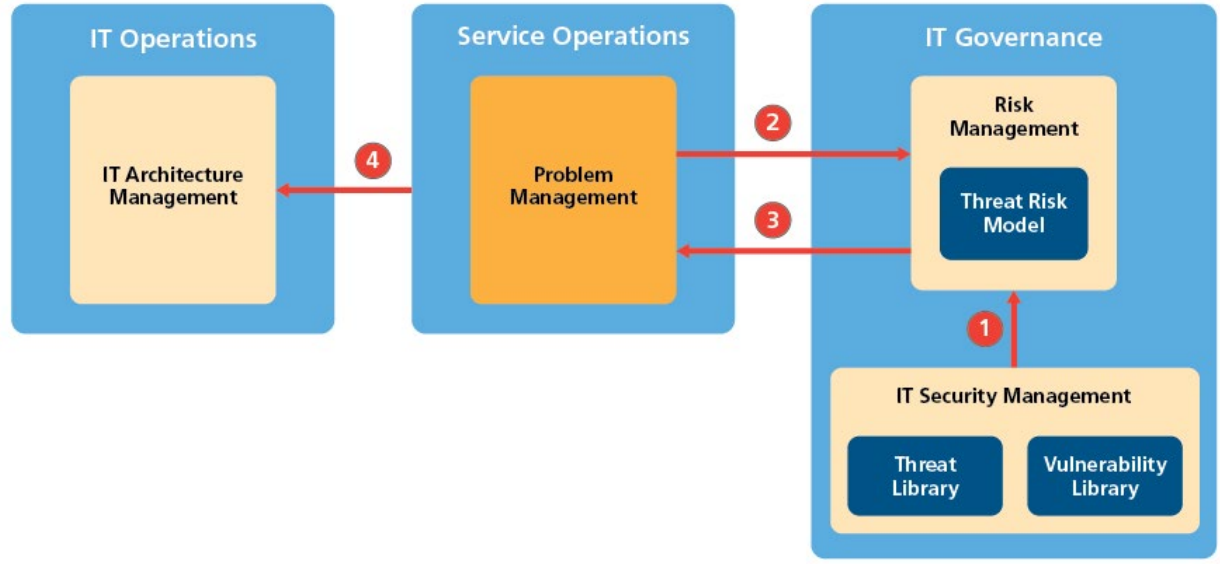

Figure 12 – Data Flow between the four key functions

Together, these four functions will support a **Risk Management System.** The core of a risk management system is a **Risk Model.**[7] Such models are used across an organisation to identify and mitigate several risks, for example – financial risk. However, our focus here is on the modelling of external risks due to cyberattack, so our model will be a Threat Risk Model. So, how we do this?

Basically, this is about mapping known or suspected **internal** vulnerabilities against known **external** vulnerabilities.

Our starting point is data from IT Security Management **(1)** in the form of potential threats **(external)** that could cause a service disruption. For example, a known virus that attacks known vulnerabilities. These vulnerabilities are available from the National Vulnerabilities Database [8] and the current Threat Library [9] is available from the National Cyber Security Centre.

We then extract data from our problem management system **(2)** - this also includes system monitoring data and output from any vulnerability software that may be running.

Data **(1)** and **(2)** are processed by a **risk model** operated by IT Risk Management and are 'mapped' against the known vulnerability and threat libraries.

The risk model generates an output **(3)** that gives a probability rating or severity rating (catastrophic, critical, marginal etc.) in terms of Configuration Items (CIs) or groups of CIs, that are vulnerable to external threats.

Based on this probability, problem management will initiate (through change management) all the architecture changes **(4)** that are needed to eliminate or mitigate these vulnerabilities.

Clearly, this is an oversimplification of a complex subject and would warrant a white paper on its own, but it does offer an opportunity to consider just how problem management can play a key role in risk management.

Conclusion

This paper recommends several improvements to the Problem Management process with a focus on proactive problem management. The emphasis has been on creating:

- a dedicated problem management team
- a more robust problem record schema and create a better interface between the KEDB and CMDB
- probability driven RCA techniques
- greater use of probability-driven analysis in support of problem detection

These steps, although not all encompassing, will enable the Problem Management process to take a more central role – not just in problem resolution – but in risk management.

The methods and techniques described in this paper are based on the Problem Management best practice offerings available from CIH Solutions. For more information please contact Chris Hodder at info@cihs.co.uk. Also visit www.cihs.co.uk

Biographical Note: The author is a partner consultant with CIH Solutions and can be contacted at info@cihs.co.uk.

References:

[1] **UK National Cyber Centre –** www.ncsc.gov.uk
[2] **UK Government's Cyber Governance Health Check** – www.gov.uk/publications
[3] **CIH Solutions Briefing Note** – *Applying Predictive Modelling to Problem Management*
[4] **CIH Solutions White Paper –** *Knowledge Management within ITSM*
[5] **ITIC Survey Results –** www.itic-corp.com
[6] **CIH Solutions White Paper** – *The Myth of CMDB*
[7] **Risk Models –** www.theirm.org
[8] **National Vulnerability Database –** www.nvd.nist.gov
[9] **UK National Cyber Centre Threat Library–** www.ncsc.gov.uk