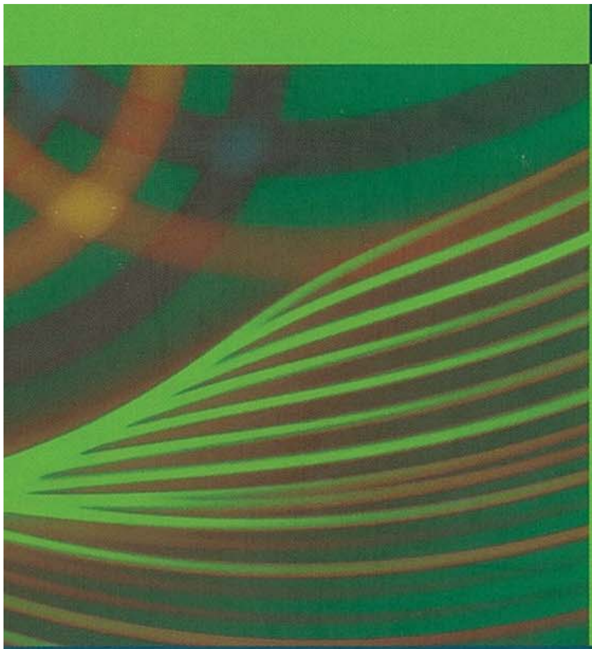


Rob Addy

Effective IT Service Management

To ITIL and Beyond!

Springer

Rob Addy

Effective IT Service Management

To ITIL and Beyond!

With 50 Figures

Springer

Rob Addy
www.effectiveitsm.com

Library of Congress Control Number: 2007929266

ACM Computing Classification: D.2.9, D.2.7, K.6.1, K.6.3, K.6.4, K.4.3

ISBN 978-3-540-73197-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

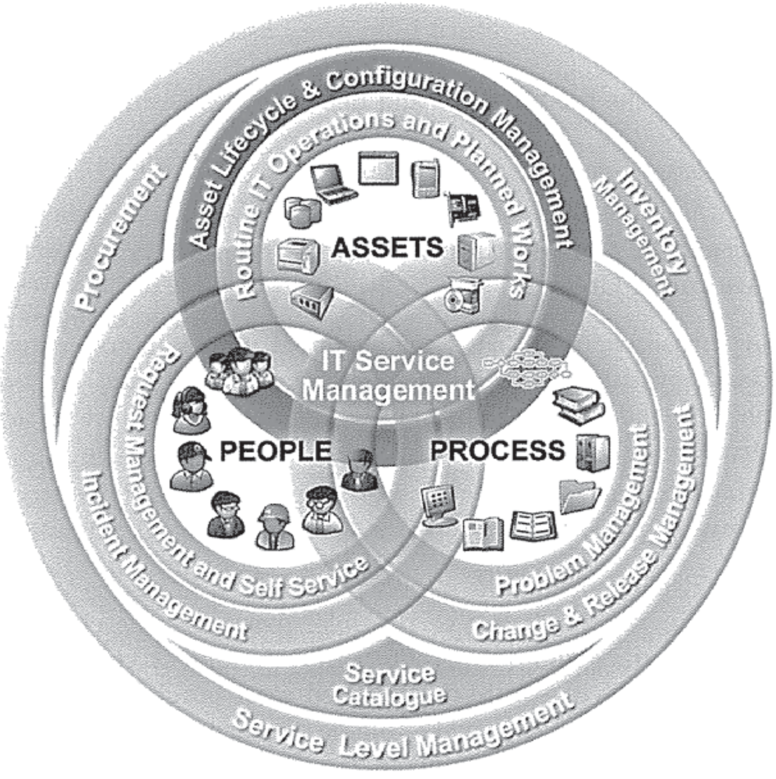
Typesetting: by the author
Production: Integra Software Services Pvt. Ltd., Puducherry, India
Cover Design: KünkelLopka, Heidelberg

Printed on acid-free paper 45/3180/Integra SPIN 12267231 5 4 3 2 1

Contents

19

Asset Lifecycle Management/Configuration Management



Also known as asset management, asset tracking, inventory management, equipment portfolio management etc.

Asset lifecycle management is the end to end process governing the way in which assets enter and exit the organisation. Configuration Management can be thought of as a subset of this process focused upon monitoring the condition and status of an asset (or Configuration Item in ITIL-speak), and its relationships with other assets within the environment. Configuration management is best thought of as the system by which the Configuration Management Database (CMDB) is maintained and managed (either manually or with automatic data feeds).

Traditionally ITIL focused configuration management processes do not go down to the level of granularity that many other industries/disciplines expect from the term. In general it is not used to track BIOS parameters, Card dip switch settings, INI file contents and the like... this level of information being considered too detailed to be tracked and recorded.

19.1 Process Objectives

The asset lifecycle management process should be focused upon the following core objectives:

- Extracting the maximum return from every asset for the minimum cost
- Extending asset life
- Understanding the relationships and dependencies within the IT environment
- Improving asset reliability and availability

19.1.1 Extracting the Maximum Return from every Asset for the Minimum Cost

The primary objective of the asset management process is to ensure that all configuration items are utilised to their best advantage, and are contributing effectively to the reliable operation of the business, whilst incurring the minimum expense possible.

19.1.2 Extending Asset Life

Extending the useful working life of every asset helps to ensure that the total cost of ownership is kept as low as possible. Assuming the changes related to the re-use, re-purposing and recycling of IT equipment are managed effectively then the cost savings from the deferred asset purchases and associated depreciation and/or lease payments can be significant. Proactive upgrades using relatively inexpensive components can dramatically enhance asset performance and capacity allowing the asset to handle increases in system usage and load without requiring the purchase of expensive replacement equipment.

19.1.3 Understanding the Relationships and Dependencies within the IT Environment

A detailed picture of service relationships, dependencies and impact models can significantly assist the IT function to mitigate some of the risks associated with the implementation of changes as well as ensuring that incidents are prioritised based upon their business impact. The increasingly dynamic nature of IT environments means that such models must be flexible enough to react to changes in circumstances. Ideally these models should have the ability update themselves based upon the findings of automated discovery tools and other system feeds.

19.1.4 Improving Asset Reliability and Availability

By analysing historic failure trends the IT function can begin to predict when future failures are more likely to occur. Preventive maintenance activities and proactive component replacements can then be planned in order to reduce the likelihood of an unplanned system outage due to a hardware failure. These actions can contribute to an increase in hardware reliability and consequently improve related service availability.

19.2 Common Issues

Common issues include:

- Asset location tracking i.e. finding out where stuff is . . .
- Asset and Component theft/loss
- Local stock piles/unofficial inventories
- Maverick purchases
- Unplanned and unapproved moves and changes
- Insufficient information available to make valid Repair vs. Replace decisions
- Lease penalties for non-return of like for like equipment
- Vast array of different configurations (Platforms, vendors, standards etc)
- Inconsistencies in data collection and reporting
- Data errors and omissions
- Under utilised warranties
- Massive volume of event management data leads to analysis overload
- Lack of useful performance information to determine actual asset usage/status
- Difficulty accurately predicting asset/component failure
- Over specification of equipment
- Under utilisation of equipment
- Asset reliability
- Asset degradation/Lack of planned preventive maintenance and upgrade
- Accidental disposal of leased assets, incurring punitive fines for non-return
- Environmental impact of IT Asset usage
- Safe disposal of assets

19.3 Process Flows

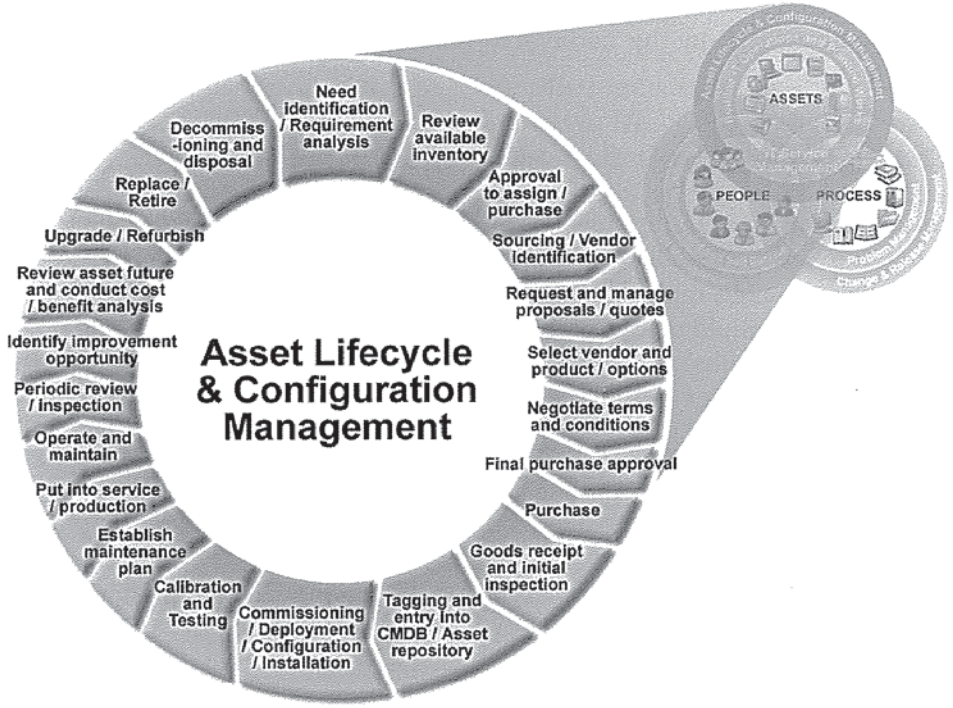

Asset Lifecycle & Configuration Management
Need identification / Requirement analysis
Review available inventory
Approval to assign / purchase
Sourcing / Vendor identification
Request and manage proposals / quotes
Select vendor and product / options
Negotiate terms and conditions
Final purchase approval
Purchase
Goods receipt and initial inspection
Tagging and entry into CMDB / Asset repository
Commissioning / Deployment / Configuration / Installation
Calibration and Testing
Establish maintenance plan
Put into service / production
Operate and maintain
Periodic review / inspection
Identify improvement opportunity
Review asset future and conduct cost / benefit analysis
Upgrade / Refurbish
Replace / Retire
Decommiss -ioning and disposal
ASSETS
PEOPLE
PROCESS
Problem Management
Change & Release Management

Fig. 19.1. High level overview of the asset management lifecycle

Key stages within the asset lifecycle include:

- Need identification to Purchasing
- Receipt to Installation
- Storage
- Commissioning/Deployment
- Routine usage (including ongoing maintenance)
- Inspection and test
- Periodic effectiveness review
- Modernisation/Refurbishment/Upgrade
- Reassignment/Re-purposing
- Retirement/Disposal

19.4 Key Players

Asset Users – People that physically access and use specific assets.

Asset Owners/Guardians – The persons or groups who are identified as having authority over, and responsibility for, a specific sub-set of assets and may decide how they are used i.e. their purpose, who may use them, their upgrade path and how the costs associated with them are to be distributed/allocated.

Purchasing – The procurement phase is integral to the lifecycle asset management process and requires purchasing executives to focus on more than just unit cost etc. Asset centric procurement ensures that the total cost of ownership, as well as non-financial measures are considered when selecting vendors and specific hardware models.

Configuration Specialists/Managers – Traditional configuration management roles are often little more than auditors of the current IT estate. However, in the future it is likely that the role will expand into the evaluation of technology advances, the definition of optimum configurations and the planning of how to efficiently migrate from the current state to the desired configuration.

Asset Manager – Individual with overall responsibility for the asset management process.

19.5 Process Metrics

- Number of units of specific asset classes, types by manufacturer, location, model, business unit etc
- Proportion of the IT estate that doesn't conform to optimised standard configurations

19.6 Asset Lifecycle Management and Configuration Management

Lifecycle management is the inclusive term used to describe all of the discrete activities associated with an asset during its existence. It covers every facet of the asset and every interaction it has with the wider IT infrastructure in general. Configuration management processes sit within the wider asset management process and cover the ongoing need to understand what you have out there at any given point in time. Configuration management should not be considered in isolation, but as part of a holistic asset management system focused at wringing every last drop of value out of an organisation's valuable assets.

19.7 The Configuration Item (CI)

"I am not a number - I am a free man!"

Number 6 (Patrick McGoohan), The Prisoner

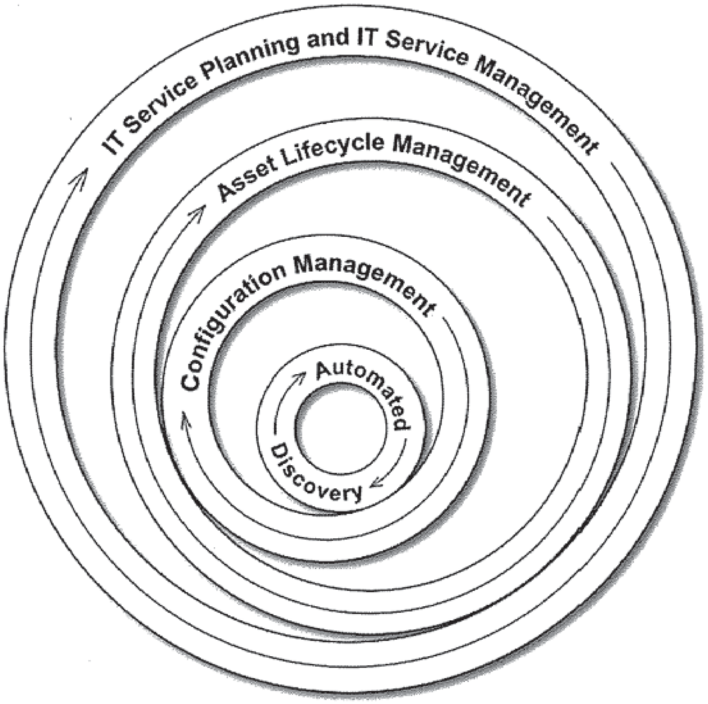

IT Service Planning and IT Service Management
Asset Lifecycle Management
Configuration Management
Automated
Discovery

Fig. 19.2. Relationship between asset and configuration management

If one were to take ITIL literally, then pretty much everything within the IT arena from the pot plants, to the employees (not that it is always possible to tell the difference in some specialist IT departments) would be considered as separate configuration items. Whether the ITIL authors expect every member of staff to be bar coded and/or fitted with a RFID tag is unclear, however it is obvious that such broad brush statements do little to help those who actually have to implement such things.

Within this book you will find that the terms configuration item, device, asset, equipment, system, instance, hardware, application etc are used interchangeably. Such use of language should not inhibit understanding and hopefully the use of these synonyms will enable the text to flow appropriately within the relevant passages. Reaching for the trusty dictionary once more we find:

"Configuration – i) the arrangement of the parts of something, ii) (computing) the particular choice of hardware items and their interconnection that make up a particular computer system"

So extrapolating this definition we may define a configuration item as:

"Any element of a computer system and/or its underpinning infrastructure i.e. the physical and logical components that come together to deliver a capability that may or may not be leveraged as part of an IT service"

Put another way, configuration items are distinct instances of stuff. 'Stuff' being the fabric of the world within which we live, work and play.

19.7.1 Defining the Appropriate Level of Granularity

The sheer scale of the task of creating and maintaining a full and accurate record of every IT asset within the business is often daunting. For this reason, businesses often decide to initially focus upon a specific subset of their asset portfolio. The following criteria can be used to help with the selection of items to be included within this initial project:

- Categorisation based – i.e. restricting monitoring activity to configuration items of a specific asset class etc.
- Cost thresholds – i.e. the tracking of configuration items above a certain financial value
- Risk based – i.e. tracking configuration items that play a significant role within service provision and support

Although the above criteria may be used to identify an initial phase within a CMDB project, there is no reason that an organisation must ever make a record of every piece of IT equipment. Within the finance world, the term 'materiality' is used to define the relevant levels of journal entries, transactions and asset valuations that are insignificant to a company's overall financial position. Where differences in trial balances, P&L statements etc do not exceed this threshold they are ignored on the basis that it would cost too much to investigate and rectify the anomaly than to simply document the discrepancy and move on. Such an approach can also be taken with configuration items – What is the level of 'materiality' within your business for IT assets?

19.7.2 Common CI Attributes and Specific Characteristics

Having decided which configuration items to track and monitor, the next decision to be made is on the level of detail to record against each item. Tracking too much detail will incur an excessive administration burden, too little will deliver minimal benefit to the business. Everything from a teapot to an enterprise server can be adequately described by a dozen or so attributes, pretty much everything else is overkill. Twelve attributes? Surely that can't be enough? Well yes, it can - provided you don't start confusing your attributes with your relationships...

"Attribute – a property, quality or feature belonging to, or representative of, a thing"

The only minor problem is the fact that those twelve characteristics will be different for each different type of item. Beware of systems with hundreds of detailed attributes on a configuration item record. Such data overload will invariably be ignored by the people actually using the CMDB, as they will (rightly) assume that the quality of such data will be dubious at best, and will always fail to contain the one key piece of information that a person needs to solve a specific technical issue.

Core attributes may include:

- Classification i.e. what type of thing is it?
- Description i.e. which one of its type is it?
- Unique identifier i.e. what is the one thing that is specific to it and nothing else?
- Status i.e. what is its current circumstance/situation?
- Key milestones i.e. when did important stuff happen to it?
- Primary use/role i.e. what is it for?

These core attributes are applicable to practically all configuration items and provide the basic details that are necessary for many ITSM processes to function adequately.

19.7.3 Specific Characteristics – Getting Personal

However, sometimes it is necessary to go beyond these core attributes and add a few more pieces of data to improve the level of intelligence, or the impression of intelligence at any rate, within process automations and to aid with the decision making process undertaken by IT personnel. The following list outlines some of the most common types of asset specific data points that are used within business rules, workflow logic and process automation engines that underpin today's "expert" systems:

- Dimensions i.e. the physical size of the thing
- Capacity i.e. the amount of stuff that the thing can handle
- Configuration settings i.e. how it is currently setup
- Standards support i.e. the protocols and standards that it can use or be used by
- Versioning i.e. which versions of the thing is it and/or what version of things does the thing support, use, leverage etc
- Costs i.e. how much was it, and how much is it costing on an ongoing basis?

19.7.4 Dynamic Data Models

The pertinent details regarding a CI may vary throughout its useful life as its role and function changes. For example, someone working with a server that is used to run an application platform layer will be interested in the Physical RAM installed, the number of CPUs and their speeds etc, however over time this machine may become slow in comparison to newer models and may be repurposed (i.e. redeployed within another role) as a file server. At this point, someone working with the machine will be more interested in the number of hard disks (including their capacity, speeds and buffer sizes) as well as the number of disk controllers/IO cards installed. It is therefore necessary to be able to dynamically change the data schema for an individual asset or asset class over time in order to ensure that the current information requirements of the business can be satisfied fully.

19.7.5 Core CI Relationships

There is an old adage that one can judge the character of a man by the company that he keeps. The same can be said of a piece of IT infrastructure. The measure of the value of a CI is determined by its interactions with the rest of the environment and the various roles that it performs. Every asset can be considered to have its own unique sphere of influence. This influence will extend to, and impinge upon, a wide variety of systems and processes and may ultimately spawn numerous relationships and interdependencies with other entities within the wider business environment. Primary relationships tend to include the following items:

- Technical dependents – i.e. the physical dependencies (upstream and downstream) between the thing and other things
- Logical dependents – i.e. the processes, systems, applications and services which rely upon the thing
- Performance objectives – i.e. the levels of availability, responsiveness, output conformity etc that are defined within associated SLAs, OLAs and contracts
- People – i.e. the current owner, custodian and user(s)

19.7.6 Secondary Relationships

Secondary relationships is largely a misnomer as these links are sometimes every bit as critical to the business as those defined previously. However, such relationships are generally considered to be of less importance than the core relationships. Examples of secondary relationships would include:

- Location – i.e. where the CI can be found now, where its been and where it is yet to go
- Components – i.e. the things that go to make up the thing in question
- Contractual – i.e. what terms and conditions apply to this CI
- Organisations – i.e. the manufacturer, supplier, maintainer etc

19.7.7 Indirect Relationships

It is sometimes necessary to iterate across a relationship map, or hierarchy, in order to find things that are indirectly linked to a configuration item. Such relationship crawling enables the advanced analysis of complex problems that seem to have no linear or direct correlation between contributory factors. As well as this advanced analytical capability, indirect relationships are often used to determine implied relationships without having to incur the management overhead associated with defining and maintaining many many direct links. Examples of such indirect relationships that are often calculated automatically for use within back office processes include:

- Department/Team – i.e. the groups that the people related to it belong to
- Cost centre – i.e. the financial account codes associated with the people, location, departments or business units related to the thing

When considering the CI/Asset level data requirements for your ITSM solution, the following questions may help you to take a step back from the common crowded over laden data heavy asset related user interface and take a more pragmatic view:

- What information do you really need to record and maintain about each CI?
- Why do you believe that you need to store this information?
- Who will be using the specific information? Are they likely to trust the stored data or will they gather it afresh each time they need it?
- How will they be using the information? i.e. Is it a part of a calculation? Does it aid decision making? etc
- Who will be responsible for maintaining the data point and how will they do this?
- What is the cost/benefit of collecting and maintaining this level of data?
- How can the data be stored and/or presented to ensure those that need the information can get to it as efficiently as possible without negatively impacting those system users that do not need to know the data?

19.7.8 CI Classification

For details regarding categorisation concepts and potential pitfalls etc please refer to the Incident Management section of this book where the advantages and disadvantages of various approaches to ticket categorisation are discussed in depth. The concepts outlined there are equally applicable to configuration items as they are to incident reports – well kind of anyway.

19.7.9 CI Lifecycles/Status Models

Just as different types of incident will require a different lifecycle and process flow, so will configuration items. Software applications will have a different status model to physical hardware items such as servers, and servers will have a significantly different lifecycle to PDAs. Each element of the infrastructure will undergo its own journey through its own specific lifecycle at its own rate and may initiate smaller sub-lifecycles during specific phases of their existence as required. For example, a server may transition from procurement to commissioning to production etc and during that high level process flow may experience numerous modernisation cycles where its capabilities are reviewed and upgraded whilst it continues on its way to its ultimate decommissioning and disposal… Equally an asset may have sub-statuses defined to give the wider IT community additional information as to the current situation, for example an "In production" status may have sub-statuses of "Backup in progress", "Database being re-indexed", "Routine Maintenance ongoing" etc to ensure that other system users are aware of the likely cause of any short term performance degradation that is experienced.

19.7.10 Cookie Cutter Fun – The CI Catalogue

The CI catalogue enables the asset manager to define a series of templates. These templates define a standard configuration which may be requisitioned and purchased before being implemented via a change request. The purpose of such a catalogue is two fold; i) it enables the user community to see what has been approved as acceptable to purchase/deploy and ii) it enables the current infrastructure to be compared against the idealised model held within the catalogue.

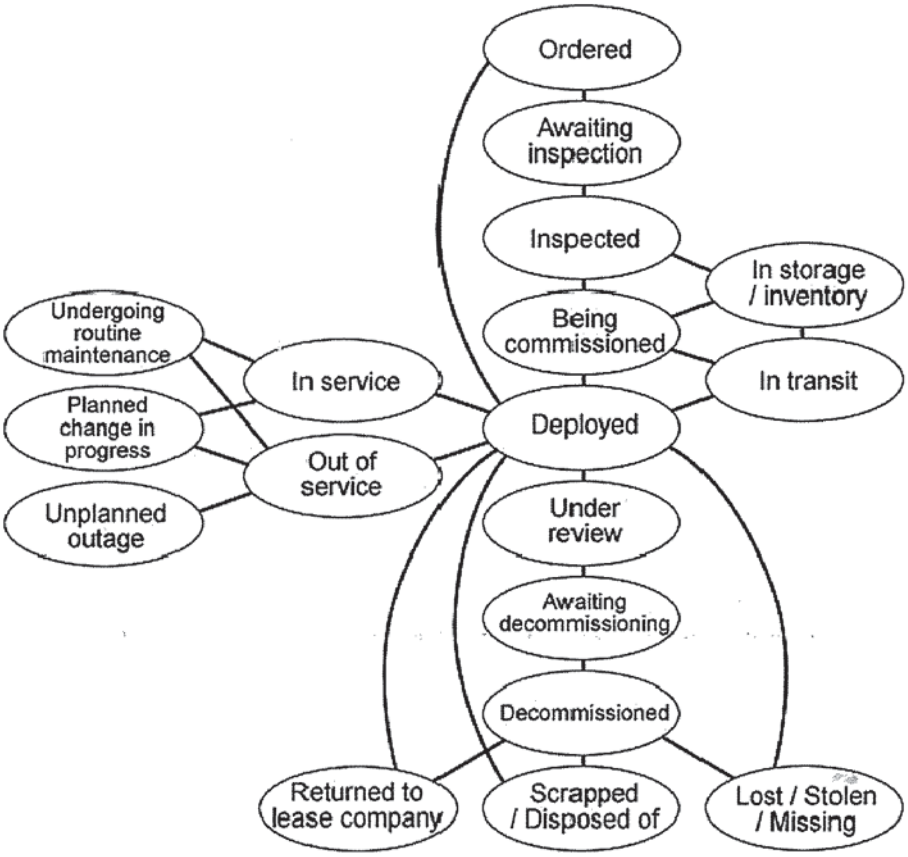


Fig. 19.3. Example status model for a physical asset

19.8 The Configuration Management Database (CMDB)

19.8.1 I am Spartacus . . . no, I am Spartacus . . .

Recently, every ITSM solution vendor has put their own particular spin on the CMDB story. Hailed as the 'single view of the truth', the 'heart of your ITSM system' or the 'manager of managers', CMDBs are perhaps the most over hyped aspect of the ITSM market at this time. ITIL's lack of definition of the nature and purpose of the repository goes some way to explaining why so much confusion has arisen. Object orientated architectures, cross asset class support, federated data sources, dynamic relationship models and intelligent reconciliation engines add to the marketing noise and do little to improve the comprehension of the average IT manager on the street.

19.8.2 A Filing Cabinet by any Other Name would Hold as Much!

In essence the CMDB is a repository to hold information regarding the IT environment and the ITSM processes which support it . . . Countless column inches within the trade press, a plethora of vendor whitepapers and hundreds of hours of presentations at sales meetings and user forums have been used to discuss the perfect CMDB design. And yet this pre-occupation with achieving the optimum CMDB architecture is often used as an excuse for inactivity. The objective of the CMDB is to efficiently and effectively support and underpin IT service management processes – nothing more, nothing less. You could spend months defining the ultimate CMDB architecture – then again, you could start improving services instead . . .

19.8.3 The Emperor's New Clothes

Just as the little boy in Hans Christian Anderson's classic fairy tale saw, the current CMDB phenomenon lacks any real substance or innovation. It is a rehash of pre-existing concepts and products to bring an illusion of freshness to the marketplace. As one sales executive remarked to me when his corporate marketing machine unleashed its own particular brand of CMDB related propaganda into the world, "Didn't we already have a CMDB?". The answer of course was "Yes", but don't let the marketing folks hear you say that! By asking this single question, this sales person had seen through the marketing hype and cut to the very heart of the matter. If only the rest of the industry could be so perceptive, then we could actually get on with having meaningful dialogues regarding the proactive improvement focused use of ITSM related data . . .

Chances are, if you have implemented a software solution to manage you IT support issues, that you already have a CMDB of sorts. Whether or not it can meet your current and future business needs is another story. But you probably already have something that could lay claim to the title of "CMDB" if you wanted to.

19.8.4 Data! Data! Data Everywhere . . .

There are very few green field sites in the world that have no incumbent ITSM solution or IT related tools in place. There will inevitably be multiple silos of information around the organisation, holding similar data about the same configuration items. This is why CMDB vendors talk of federated data models where legacy systems retain ownership and control of their data but are leveraged by the central master CMDB to give a holistic view of the world. The theory being that the central repository does not have to replicate the data that is distributed throughout the infrastructure, instead it merely stores pointers to the remote datasets and details of how it is to be accessed (including any real time translation or transformation processes necessary). Real time transactional integration engines then connect to the remote information silos as required and return a combination of local and federated data points to the end user seamlessly when requested. Like I said, that's how it works in theory.

Perhaps the classic hypothetical example of data federation is an ITSM system which manages the day to day operation of IT assets in parallel to an ERP system which maintains the fixed asset register entries for the IT assets and calculates the depreciation, book value and residual value on a regular basis. In the event of a terminal failure of an IT asset or component, the ITSM system is able to automatically connect to the ERP system in real time and determine the current financial value of the item, the availability of appropriate spares within inventory etc, and can then use this data point as part of its "repair or replace" business logic to provide real time decision support information to the asset manager. Unfortunately, few commercial systems are at this hypothetical level yet. Federated data models are a much talked about, but seldom delivered, functionality at present. One should particularly beware of vendors claiming to deliver federated data layers which are nothing more than a series of point integrations that enable a user to drill down into an external data source from within the native user interface. Launching a hardware vendor's customer support site and navigating to the configuration of a specific piece of hardware is a good trick, but is it really a federated database?

The cynics amongst you may remember the previous marketing campaigns from the major ITSM vendors which asserted that multiple silos of information, or disconnected islands of data, were a very bad thing indeed. In fact, these vendors spent many years trying to force unnecessary data consolidation upon unwilling customers that one has to wonder if they have all recently suffered from corporate amnesia, or perhaps dementia. Maybe, but It may also be that they have all decided that if they can't beat them then they may as well join them . . .

19.8.5 Having One's Cake and Eating it . . .

"Federation – a loosely coupled collection of separate entities which co-operate with each other, and operate together, for mutual benefit whilst retaining overall control locally"

Federated data models allow incumbent legacy systems to continue to function as before, minimising the risk to the business from having to rip and replace numerous systems at once. They also enable the organisation to avoid sometimes difficult internal discussions regarding the ownership of data and who has the right to update it. Many vendors dodge the issue of maintaining federated data by stating that a request for such an update is passed to the administrator or owner of the external data for them to review and implement. In reality, such a labour intensive process is highly likely to be unworkable and so organisations are faced with a choice; either they live with data quality and data consistency issues or they invest in transactional middleware which can support the application of sufficient data validation and business logic testing to inbound update requests before automatically performing amendments to the federated repository.

19.8.6 Trusted Sources and Data Confidence

Not every piece of data relating to the same configuration item will be consistent and correct. Different data sources will have different levels of data quality, and within a specific data source the level of data quality may fluctuate significantly. Determining the level of confidence that one has in a particular subset of data from a particular data source is often a difficult task, based upon incomplete information and gut feeling rather than detailed information. The following factors should be considered when evaluating data quality:

- The age of the data i.e. the amount of time since the data was collected/last updated
- The level of control in place regarding updates i.e. data validation routines, restricted access etc
- The mechanism used to update data i.e. manual data entry versus automated data capture and input
- The number of people or systems that are permitted to update/modify the data source
- The particular competence of the system populating the repository i.e. is it a generic discovery tool reporting upon an uncommon operating system or is it a native vendor supplied tool that should have better access to the details of the O/S in question.

Unless a sufficient level of confidence in contributing data sources can be established, and defined within robust business logic and automation rules, then the reconciliation process will remain as labour intensive as before.

19.8.7 Populating the CMDB

Given that a CMDB can be thought of as a filing cabinet, unless it is filled with information it is nothing more than an empty cupboard... Now a filing cabinet that is crammed full of valuable records is one thing, but an empty one is something else completely! So it is essential to populate and maintain the CMDB. The following steps will help you on the way to making your filing cabinet potentially useful:

- Identify data sources
- Rank data sources for trustworthiness
- Determine which data points from which sources are going to be considered as the master
- Perform data cleansing (e.g. de-duplication, corrections, applying defaults for omissions etc)
- Integrate with data sources/Extract required data
- Define import methodology and schedule/frequency
- Establish and implement data validation rules
- Import data into the single system
- Consolidate data and/or reconcile multiple datasets to form master
- Define data synchronisation requirements
- Periodically refresh external data sources with master data
- Periodically re-import changes from source datasets and reconcile
- Define controls to maintain master data
- Enforce controls

19.8.8 Leveraging the CMDB – i.e. using it . . .

There is little or no point in having a CMDB unless it is used. The more the CMDB is used the greater the return will be upon the resources expended in its creation and ongoing maintenance. Every single process within the ITSM arena has the potential to touch the CMDB to access information or to use it to store information related to specific instances of the process inputs and outputs. In short, the CMDB should be an embedded part of every IT process and should be used by every member of the IT function on a daily, if not hourly basis.

Given its pivotal role it is perhaps interesting to see some vendors claiming to be able to overlay a shiny new CMDB on top of an existing brown field site. How can such an approach pay dividends unless every system used to manage ITSM processes is plumbed in to the central source of asset data? Surely the level of real time bi-directional integration required and the associated professional services effort needed to actually realise such a vision would make the payback or break even period for such a project unworkable. . . Or are such claims merely a thinly veiled disguise to hide the need to rip and replace your underlying infrastructure? Caveat emptor!

19.8.9 Common CMDB Related Issues

The following list describes some of the most common problems encountered when implementing and using a CMDB:

- Too many configuration items being tracked i.e. level of granularity being set too low
- Poor data quality (Errors, omissions, duplicates etc)
- Overly complicated data models with excessive numbers of asset classes and relationship types being modelled.
- Numerous disparate data sources with different levels of data quality
- Consolidating reports across distributed and/or object orientated data models can be troublesome
- Gaining consensus within the business regarding who should be the owner of particular elements of the data landscape

19.9 Meet the Parents . . . It's all Relative

"The foot bone's connected to the leg bone,
The leg bone's connected to the knee bone,
The knee bone's connected to the thigh bone,
The thigh bone's connected to the back bone,
The back bone's connected to the neck bone,
The neck bone's connected to the head bone,
Oh, hear the word of the Lord!"

Dem Dry Bones, Unknown

Relationships are important. Relationships have always been important. Recently, IT management has become convinced, helped largely by a concerted marketing campaign from some of the biggest vendors in the business keen to foist their latest and greatest relationship visualisation toolsets upon the unsuspecting IT executive, that relationships are all important. Placing too much of an overt focus on the relationships and dependencies between the configuration items that support business services can have a detrimental effect upon the overall quality of service delivered . . . How so? Concentrating solely upon mapping the relationships between physical elements of the infrastructure may leave other factors that contribute to service quality out in the cold. External factors such as network load/congestion, shared resource consumption, contention between multiple business services, demand spikes, personnel availability etc. can all have a significant impact upon service delivery and overall service quality. It is important to bear in mind that it is not necessary for a machine within a defined dependency hierarchy to go down for a service to be impaired beyond use.

Assuming that service dependency modelling is done as part of a holistic approach to IT service management then it can be a useful tool in the resolution and prevention of business impacting events and issues. When configuration item relationships are used to describe more than mere linear binary dependencies, they become even more powerful and enable the IT organisation to prioritise incidents in real time according to the risk of service interruption or impairment.

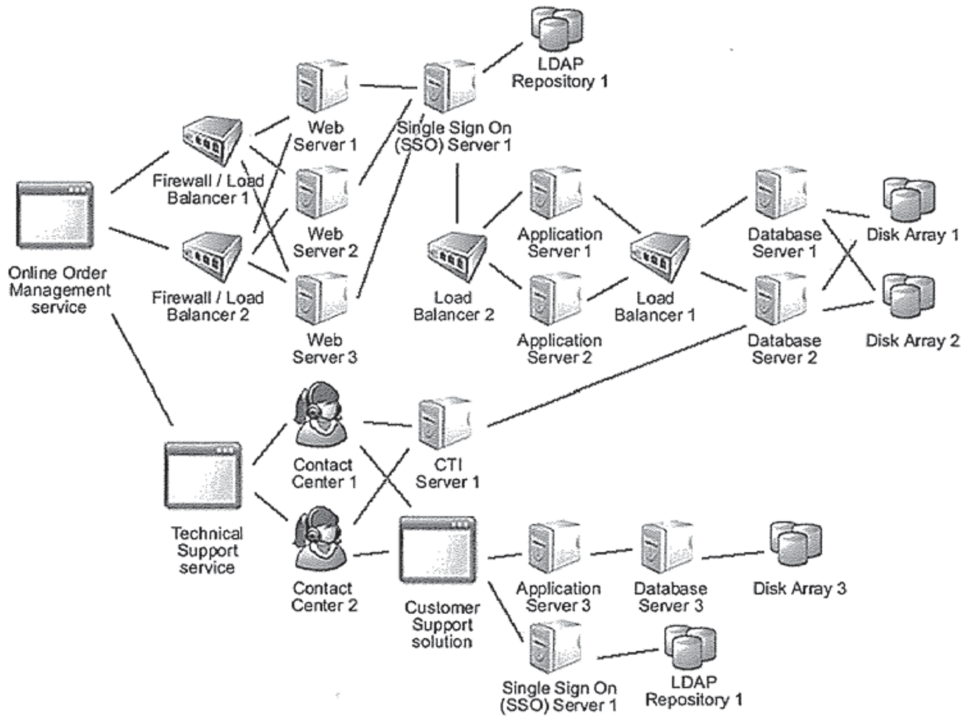


Fig. 19.4. Example relationship map for a business service

19.10 Elements of a Relationship

Naturally for a relationship to exist their must be two or more parties involved. At the lowest level this is sufficient to define a link. But in order to be truly useful, relationships must be supplemented with additional data points such as:

- Relationship type
- Ratio
- Uniqueness/Exclusivity
- Direction
- Initiation date/time
- Expiration date/time
- Impact effect
- Weighting factors
  - Parameter
  - Contribution

19.10.1 Relationship Type

A description of the nature of the relationship between the parties involved. The following list outlines an example of some relationship types that may be found within an IT environment:

Entity type 1	Relationship type	Entity type 2
Configuration Item 1	... is dependent upon ...	Configuration Item 2
Configuration Item 3	... acts as hot backup for ...	Configuration Item 4
Configuration Item 5	... forms part of ...	Item Group 1
Individual 1	... is a member of ...	Team 1
Individual 1	... is the primary user of ...	Configuration Item 6
Individual 2	... is the manager of ...	Team 1
Configuration Item 6	... is situated within ...	Location 1
Business service 1	... utilises ...	IT System 1
Individual 3	... is a user of ...	Business service 1

19.10.2 Ratio

A description of how many parties of each type may be involved in the relationship. Typically, relationships are 1:1, 1:n, n:1 or n:n where n is any number. Relationships involving more than two parties may leverage the concept of a relationship group to manage the additional links or instead the relationship object itself may be capable of understanding that there are more than two elements to the relationship. An example of a 1:n relationship would be the link between a team lead and their employees (except in cases of matrix management obviously). A load balanced application dependency chain on the other hand may have multiple n:n relationships between the relevant pieces of infrastructure and software that go to make up the service.

19.10.3 Uniqueness/Exclusivity

There are two types of exclusivity that can apply to relationships:

- Mandatory uniqueness
  Some types of relationship may only be able to be made once and may never be amended. For example the relationship between a piece of hardware and its original manufacturer is an exclusive relationship – the equipment can only be initially built once and by one organisation.
- Historical exclusivity
  Some relationships may only be in place between one set of objects at any point in time. The institution of marriage is a case in point. Excluding polygamy and bigamy, a marriage must only exist between two parties at any point in time. People can be married multiple times but not during the same period.

19.10.4 Direction

Many relationship types have a directional element to them. For example "X is the manager of Y" has a very different meaning to "Y is the manager of X"! It is therefore necessary to clearly define on which side of the relationship each related entity resides if the true meaning of the link is to be preserved.

19.10.5 Initiation and Expiration Date/Time

Without start and end dates, relationships become stateless. This restricts their usefulness significantly as there is no means of identifying the scope and status of the relationship model at any point in time and prevents it being compared against historic or future versions. The completion of initiation and expiration dates for a relationship also enables one to determine whether the relationship is current (present), historic (past) or planned (future). Planned relationships should be managed carefully to ensure that the predicted link is actually in place, via suitable validation mechanisms, when it becomes due...

19.10.6 Impact Effect

The impact of one entity within a relationship upon the other entity, or entities, is something that is often over simplified. The nature of the impact of a failure on one side of a relationship may be linear, non-linear or based upon a combination of other external factors. Linear impacts describe scenarios where the impact upon the related item is directly proportional to the level of failure on the other side i.e. a system failure on one side can be considered to cause an automatic system failure on the other, similarly, a performance degradation of 50 % would be seen to reduce the performance of the related item by half. In today's IT environment it is relatively uncommon for large production systems to have single points of failure and as such a total failure of a single element of a service dependency chain is likely to have a negative impact upon service capacity and performance without taking the service down completely. Put simply, the impact effect describes how the capacity or capability of an entity is linked to the capacity or capability of those entities to which it is related.

19.10.7 Impact Calculation Rules

A series of business rules may be defined against each relationship record to define how each party to the relationship may affect the other under a series if predefined circumstances and/or scenarios. Such rules can come in a variety of forms:

- An algebraic expression to calculate the relative capacity of each entity into which characteristic data from both sides of the relationship may be substituted before performing the evaluation
- A sequential series of rules, or test cases, that have a corresponding capacity rating assigned against them in the event of a match or pass. The rules are tested in sequence until a pass is identified and the corresponding capacity value is set
- A banded approach which defined ranges within which asset parameters of characteristics may be in order to consider the asset capacity to be at a certain level

19.10.8 Weighting Factors/Contribution to Capacity

The following list describes some of the asset attributes that are sometimes used within impact effect calculations:

Parameter	Contribution to the relationship
Disk capacity	Megabytes of free space
Processing power	Total CPU capacity
Memory available	Available RAM
I/O throughput	Number of connections and speeds

19.11 Configuration Management

Configuration management in an ITIL context is all about knowing what you have and how it interacts with your other stuff. Minor details such as where it is, what it's for, how much it's costing you, who is responsible for it etc seem to be optional extras in the land of ITIL. The ITIL configuration management process covers the identification, recording, and reporting of IT components, including their versions, constituent components and relationships. The basics of configuration management, according to ITIL, are as follows:

- **Planning** - Defining the purpose, scope, objectives, policies and procedures relating to Configuration Management. *I.e. Deciding what you want to do about Configuration Management and writing it down...*
- **Identification** - Selecting and identifying the configuration structures for all the infrastructure's CIs, including their 'owner', their interrelationships and configuration documentation. It includes allocating identifiers and version numbers for CIs, labelling each item, and entering it on the Configuration Management Database (CMDB). *I.e. the Administrative task of logging asset details...*
- **Control** - Ensuring that only authorised and identifiable CIs are accepted and recorded, from receipt to disposal to ensure that no CI is added, modified, replaced or removed without appropriate controlling documentation. *I.e. Maintaining a paper trail for all activities involving assets...*
- **Status accounting** - The reporting of all current and historical data concerned with each CI throughout its life cycle e.g. tracking the status of a CI as it changes from one state to another. *I.e. Keeping records to be able to demonstrate how and when things changed...*
- **Verification and audit** - Audits to verify the physical existence of CIs and check that they are correctly recorded in the Configuration Management system. *I.e. Checking that the records are correct and recording the fact that you checked...*

Configuration management can, and should, be much more than a record keeping exercise. In addition to the traditional defined ITIL role, configuration management teams can increase their contribution, and therefore their value, to the business by beginning to perform the following tasks:

- Impartial analysis of technical advances, product innovation etc and its usefulness to the business
- Determining the number and types of standard configurations needed within the infrastructure
- Technical definition of standard configurations for the various operational areas of the business
- Analysis of the currently deployed IT environment to determine the level of compliance with the target configuration model
- Identification of the changes necessary to move existing equipment towards the relevant standard configurations
- Development of rationalisation plans and proposals
- Status reporting on progress towards a rationalised configuration model

19.12 Automated Discovery

Automated discovery tools work in a variety of ways to discover and identify IT devices (either by passively monitoring network traffic and extracting machine details from the packet headers or by proactively broadcasting messages across the network and waiting for devices to respond). Having discovered and identified a target device, the discovery tool then attempts to collect data about the system. This auditing activity can be achieved using remote commands such as MSI, SNMP etc or via a dedicated client agent application that is installed upon the discovered machine to interrogate the system configuration before transmitting the results back to a centralised server.

19.12.1 Horses for Courses...

There is currently a vast array of different discovery tools available within the market. Not all discovery solutions perform the same function and this should be remembered when evaluating various tools in order to ensure that one compares apples with apples. Although all discovery tools do share some common functionality it is the differences that are more interesting. The three main types of discovery tool available today include:

- Configuration discovery
- Topology discovery
- Security discovery (Systems administration/Vulnerability testing)

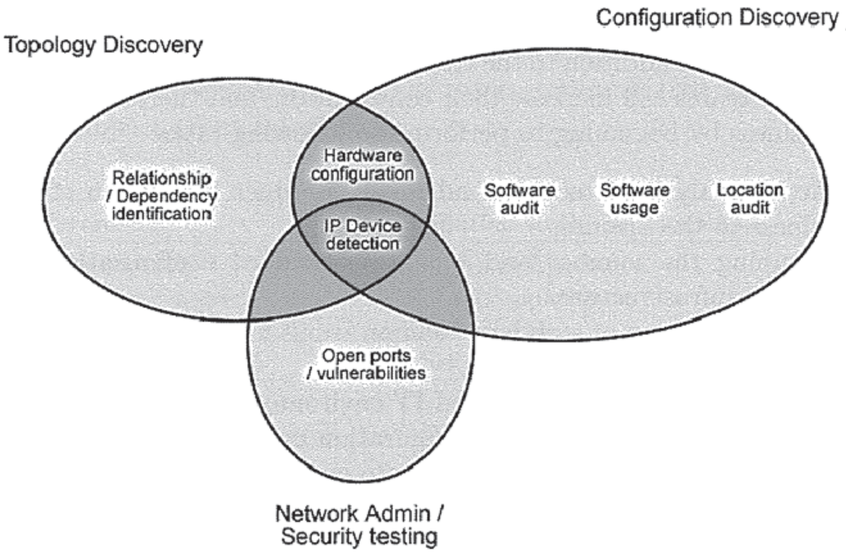


Fig. 19.5. Relationship between various automated discovery toolset types

19.12.2 Configuration Discovery

Configuration discovery tools perform periodic audits of the hardware configuration and the installed software on a target machine. The level of detail and accuracy of information gathered will vary from vendor to vendor and will often depend upon the operating system running on the target machine. For this reason it may be appropriate to use multiple discovery tools within an environment to ensure that all of the required data is captured accurately.

19.12.3 Topology Discovery

Topology discovery can be thought of as the dynamic identification of interdependencies between networked configuration items to form service relationship models and dependency hierarchies. Topology discovery tools typically use a combination of network traffic analysis and client application configuration details to build a map of data flows between solution components.

19.12.4 Security Discovery (Systems Administration/Vulnerability Testing)

Vulnerability scanners use a variety of broadcast protocols to interrogate network devices, probing for open ports and other security related issues.

19.12.5 Frequency of Discovery Sweeps/Audits

Management must temper their unnatural desire to run discovery sweeps continuously if they are to prevent them negatively impacting the wider IT environment. Realistically, the frequency of discovery sweeps should be based upon the anticipated level of change expected and may be different for different portions of the infrastructure e.g. mobile machines or machines in public places should be audited more frequently than machines located in a secure server room. When defining an audit schedule the following factors should be considered:

- Network impact (due to the associated data volume and the number of network trips) of conducting the audit (particularly for agent-less discovery solutions) and transmitting the results back to the central server
- Processing capability of the discovery server (i.e. is the server able to deal with the backlog of results that it needs to process)
- Frequency of change within the IT environment (i.e. there is little to be gained from auditing an area which changes infrequently every hour)
- The needs of the business i.e. what is the IT function trying to achieve from its automated discovery activity

19.13 Reconciliation – Turning Data into Information

"Reconcile – to make, two apparently conflicting things, compatible or consistent with each other"

Reconciliation within an IT context can be considered as the process of comparing one set of data against another in order to identify unique records (i.e. additional records or missing records), to enhance one set of data with information from another or to determine instances where the details of a record have changed (i.e. differences). In other words, IT reconciliation is a high tech game of snap where each player has literally thousands of cards...

It should be noted that reconciliation does not necessarily have to be between a CMDB dataset and a set of automated discovery results. Reconciliation may also be used as a part of the following scenarios:

- The enhancement of discovery data with financial and contractual information by reconciling the CMDB data with a data feed from an ERP system
- A comparison of the current situation against a predefined target situation e.g. a future 'blue sky' view of the environment after a planned major series of changes have been implemented, in order to track progress
- The assignment of person related information (e.g. user, owner, custodian etc) against hardware data e.g. by reconciling an application usage log containing IP addresses and usernames against a list of machine specific IP addresses.
- The enhancement of discovery data with additional and/or more reliable technical data points from a specialist, or niche, discovery toolset

Reconciliation is typically a two phase process involving two datasets, a master/source and a target. Initially a target record is matched against the master/source dataset to find the corresponding entry. This matching can be done using a series of ranked pairing criteria e.g. machine name, serial number, IP address, MAC address etc, or a combination of such attributes to find a unique match. The match may be done by a direct logical test or by formatting or manipulating the data (e.g. converting both sides to uppercase text to eliminate any case sensitivity issues etc) prior to performing a comparison. Where a match is found, the two records are usually tagged with a unique identifier to allow easier reconciliation on subsequent reconciliation runs. After matching, the two records will be compared and a series of logical checks performed. These checks will be defined as business rules and may compare attributes and/or associated child records e.g. installed software, associated components etc in order to identify differences or exceptions.

Reconciliation processes in all but the smallest of IT environments will involve many hundreds of thousands, if not millions, of queries, checks and validations in order to complete fully as fundamentally every row in the target dataset must be compared against every row in the master data source. Naturally, features such as data subset identification, intelligent querying logic, multi-threaded matching engines and pre-reconciliation indexing and sorting tasks may reduce this burden somewhat but the workload will remain significant. The level of processing power necessary and the timeline for a reconciliation run to finish should not be under estimated. For this reason it may be advisable to dedicate specific hardware to perform this function in order to ensure that the normal operation and usage of the system is not impaired.

19.13.1 Manual Reconciliation (Walk Around Audits, Inventory Checks, Stock Counts etc)

Even the most sophisticated automated discovery tools are unable to answer some fundamental asset related questions. Where is the asset located? Who is using the asset? What is the asset being used for? This basic information is often inaccessible to discovery tools and it is necessary to physically visit locations, interview users or custodians and record this data manually. The results of this auditing activity can then be imported into the service management solution and reconciled against the CMDB to update these CI attributes.

19.13.2 Exception Handling

Where the reconciliation process identifies discrepancies between the external results and the baseline data then an exception report is usually raised. Exceptions are typically one of three types; an expected item not being present, an unexpected item being found or a difference in an attribute or characteristic of a known item. The exception handling process then applies a series of business logic against the exception and takes the appropriate action. The level of manual intervention and the process to be followed for each exception type will vary from organisation to organisation but the following options are commonly found:

- Initiate an investigation (via an incident report)
- Initiate a change request to return the asset to the previous state
- Allocate available software inventory against the asset
- Initiate a purchase cycle to legitimately buy a software license in order to restore license compliance
- Initiate a change to remove the offending software
- Update the CMDB with the revised details
- Update the base record set with additional data points from the matched records
- Perform a virus scan to ensure that the change is benign
- Log the exceptions to a report, put the report in a file, put the file in a cabinet and forget it . . .

19.14 Asset Lifecycle Management – From Soup to Nuts . . .

"Which creature in the morning goes on four feet, at noon on two, and in the evening upon three?"

Riddle of the Sphinx, Greek Mythology

Thankfully, we don't have to undergo the trials of Oedipus to recognise the fact that the passage of time has an effect upon all things, from men to their machines, a natural lifecycle exists which governs and predicts the various stages of existence with uncanny accuracy.

A basic asset lifecycle may include the following phases:

- Need identification/Requirements analysis
- Procurement
- Goods receipt
- Inventory management
- Storage and transportation
- Commissioning
- Normal operations
- Re-definition of role and/or function
- Upgrade/Modernisation
- Decommissioning
- Disposal

19.14.1 Need Identification/Requirements Analysis

Long before the shiny new server is finally delivered; it is nothing more than a glimmer in the eye of some techno geek. But before even that, it is a set of requirements from the business. The role of the asset manager is to review these requirements against existing capabilities to determine if they can be satisfied using the current infrastructure or if new equipment is needed.

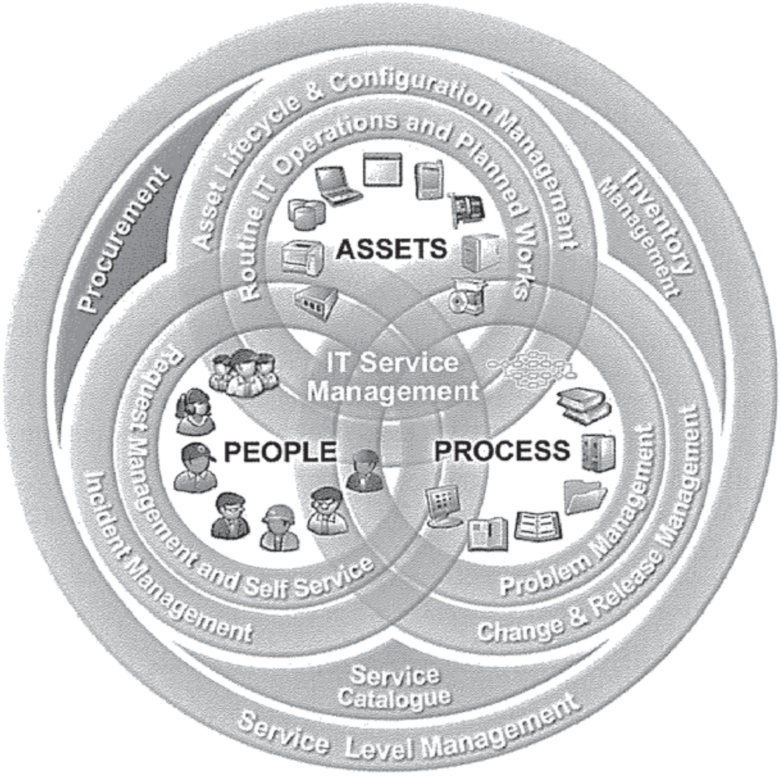
19.14.2 The Black Art of Hardware Sizing and Selection...

Much to the annoyance of anyone attempting to size a box for a particular business application scenario, software and hardware vendors are often vague about the likely capacity and performance characteristics of their offerings. Phrases such as "The performance metrics given are based upon laboratory conditions and should be used for indicative purposes only - actual performance may vary" are commonplace and it is left to the IT function to make an educated guess based upon past experience, peer suggestions etc...

Pitfalls to avoid:

- Over specification of hardware beyond all realms of reason
- The selection of new technology because it's "cool"
- Neglecting to account for peak usage spikes sufficiently
- Equipment with limited upgradeability
- Failing to include projected volume/usage growth in the design

19.14.3 Procurement



Despite IT budgets being cut significantly in recent years, total expenditure on IT hardware, software and services is still considerable in the majority of organizations. Whilst the corporate purchasing function implements formal procurement processes and procedures for larger capital purchasing decisions, routine and replacement purchases are usually left to local IT function. Unless a formal procurement system is used in conjunction with approved supplier and standard configuration listings, costs can spiral out of control and the IT estate can quickly become awash with maverick purchases of unauthorized equipment and multiple unsupported, or unsupportable, configurations.

The procurement process tracks requests from initial purchase requisitions, through formal multi-stage technical and financial approval cycles, to placing the formal purchase order with the vendor. Upon delivery, the process may govern the logging of goods receipts (including partial shipment support and returns of faulty or damaged goods etc), asset tagging and serialization, invoice matching and reconciliation against orders as well as formal cost allocation across multiple cost centres.

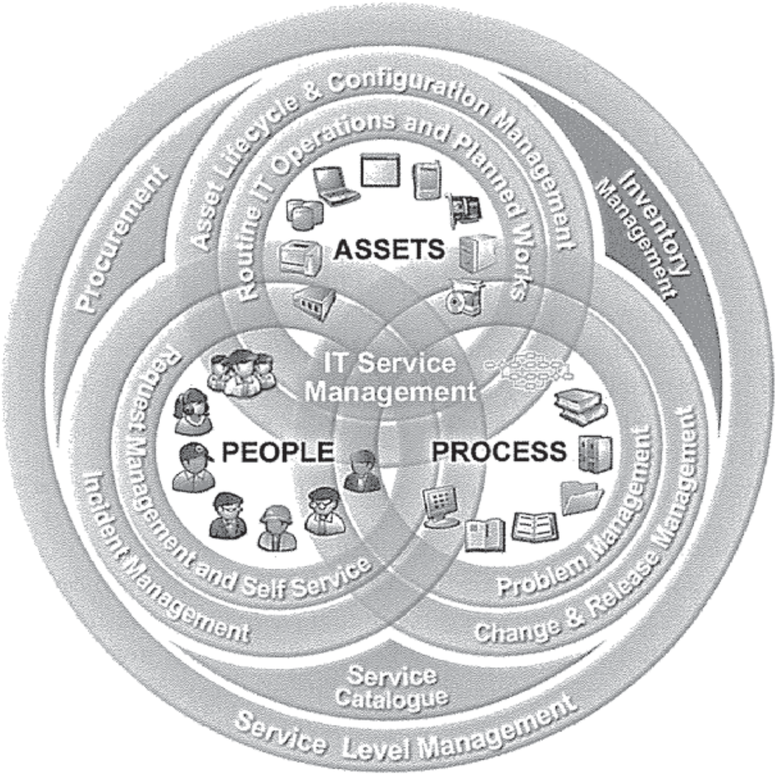
The procurement process should enable the IT function to:

- Leverage corporate purchasing agreements more effectively
- Rank and rate vendors according to multiple key performance indicators such as price, delivery times, quality (e.g. delivery punctuality, completeness and product conformance to requirements) and invoicing accuracy
- Significantly reduce the amount of off-contract/maverick purchasing within the business, reducing business risk and improving the quality of products and services procured
- Negotiate improved rates and terms with external suppliers based upon actual vendor usage and performance data
- Automate routine purchase requests based upon inventory usage and other external events to facilitate just in time equipment and material sourcing

19.14.4 Goods Receipt

Typically IT asset management systems fall short at the critical stage of the process where the wheels touch the road i.e. when the goods are actually received. Goods receipt procedures should be implemented in order to ensure that the business receives what it has paid for, at the quality it expects and that it is recorded and entered into the configuration management database (CMDB) correctly. The goods receipt process must be flexible enough to handle the following use cases/scenarios; complete shipments, partial receipt, multi stage receipt cycles (i.e. including the use of inspection and holding areas etc), inbound inspection, tagging and return processes.

19.14.5 Inventory Management



Effective stock control and inventory management is an essential element of a holistic lifecycle asset management system. Inventory systems provide asset managers with a detailed picture of asset allocation and geographical/operational distribution. This ensures that the utilization of valuable IT assets can be maintained through proactive dynamic assignment according to business needs and equipment availability. Only then can an appropriate return on investment on an asset by asset basis be assured, guaranteeing the lowest possible total cost of ownership and maximizing the return to the business. Implementing robust inventory management procedures will help the IT function to:

- Minimize unnecessary expenditure on costly IT assets that already exist within the organization
- Reuse/Re-issue surplus and returned equipment before it becomes obsolete
- Reduce stock levels and associated storage costs etc to the minimum needed to provide the required level of service to the business

- Track the location and condition of mission critical components of the IT infrastructure
- Maintain adequate stock levels of consumables and commonly requested spare parts to minimize service down time in the event of a failure etc
- Plan and predict asset usage (including the reservation of equipment) against specific projects and/or cost centres to ensure that equipment utilization is maintained and excessive overhead avoided
- Control the issue and return of all assets effectively (including loan equipment)
- Avoid incurring storage costs for excessive and/or obsolete inventory items

19.14.6 Storage and Transportation

We are fortunate that most IT equipment and related consumables are non-perishable. With perhaps the exception of printer cartridges and screen wipes, IT equipment can be left on the storeroom shelf for an indefinite period and there will be a reasonable chance that, when sometime in the future we pull it down and blow the dust off it, that it may still actually work. That is, until we subject it to the rough and tumble of the delivery man . . . Technology advances have meant that it is no longer necessary to 'park' hard drives before transportation but the opportunity for fatal damage still remains. Most people have experienced a dead on arrival (DOA) delivery and the corresponding frustration and delay. Careful packing helps but it is somewhat inevitable that a small percentage of equipment will not be working when it arrives at its final destination.

19.14.7 Commissioning

Commissioning is the process of configuring a new piece of equipment in order to make it usable and useful to the business. Unless a high end server is commissioned correctly it may become little more than a very expensive heater contributing to nothing more than the ambient temperature in the machine room. Commissioning typically includes; software installation, configuration of the environment, installation of additional hardware components, driver setup, performance tuning, load testing, application of security policies etc

19.14.8 Normal Operations

Under normal operating conditions with all IT equipment working satisfactorily, you could be mistaken for thinking that the life of a configuration manager would be nice and quiet. In fact, there are several ongoing activities that must be attended to, including:

- Performance monitoring
- Utilisation measurement
- Rationalisation planning
- Migration planning

19.14.9 Performance Monitoring

Configuration managers are often charged to keep a watching brief on the day to day performance of the configuration items under their control. Whether this be through the use of end to end transaction monitoring tools or server based statistics, the objective of the exercise is to identify the first signs of a downward trend in performance and to initiate proactive actions to prevent users being affected.

19.14.10 Utilisation Measurement

Understanding the level, frequency and profile of usage a particular asset experiences on an ongoing basis helps the configuration manager to plan for demand expansion and/or to investigate the possibility of sharing the resources of the asset with another process, service, function or role. The goal for every piece of equipment is to make it as utilised as possible, thereby increasing the level of return from it, without jeopardising overall system integrity or reliability.

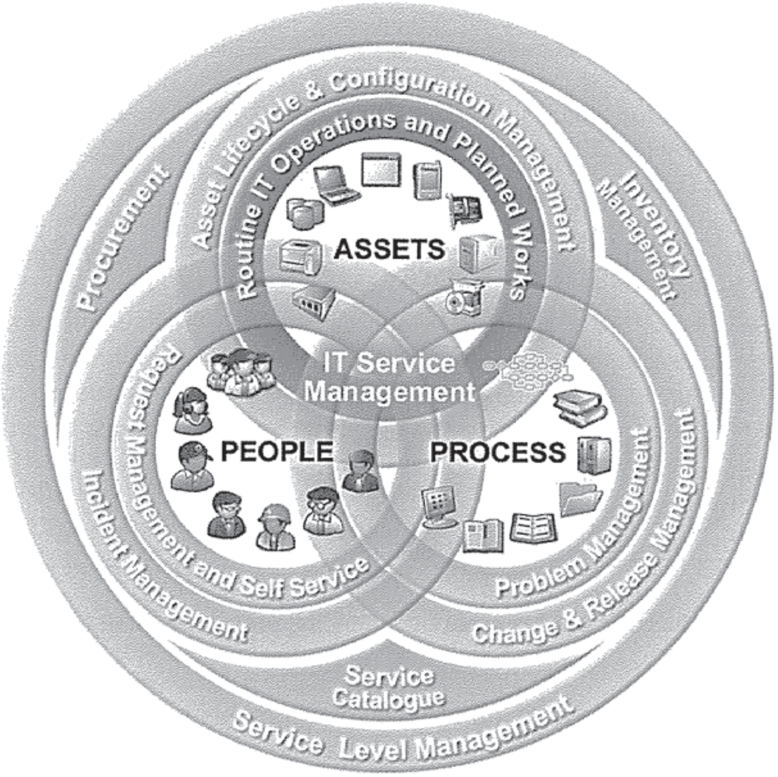
19.14.11 Rationalisation Planning

Just as every procurement professional is continually looking to reduce the number of suppliers used by an organisation, progressive asset or configuration managers will continually attempt to reduce the inherent complexity of their IT infrastructure by reducing the number of hardware and software vendors, platforms, architectures and technologies deployed throughout the environment.

19.14.12 Migration Planning

It is rarely practical to implement significant infrastructure changes in one step. The configuration management team will often need to work closely with change managers to help determine the current status of major changes. By comparing "as is" snapshots of the environment against "as should be" descriptions, deltas can be identified and the relevant plans made to migrate from the current situation to the desired state.

19.15 Preventive Maintenance



Note: Also known as Routine operations or IT operations.

Preventive maintenance activity is aimed at minimising the likelihood of service impacting events by proactively performing routine operations tasks necessary to prevent system issues on a periodic or condition based schedule. IT operations can be thought of as performing the equivalent role of a maintenance team e.g. checking oil levels, greasing nipples, tolerance checking etc within a manufacturing environment. I.e. the preventive actions necessary to keep the IT machine ticking over and to prevent performance significantly degrading due to normal wear and tear...

19.15.1 Condition Based Scheduling

With condition based scheduling the period between maintenance tasks is not fixed. Instead, pre-defined characteristics or attributes of the asset are monitored to determine when the next maintenance activity is required, just as a car service becomes due after a specific number of miles have been completed. IT related maintenance conditions may include the number of

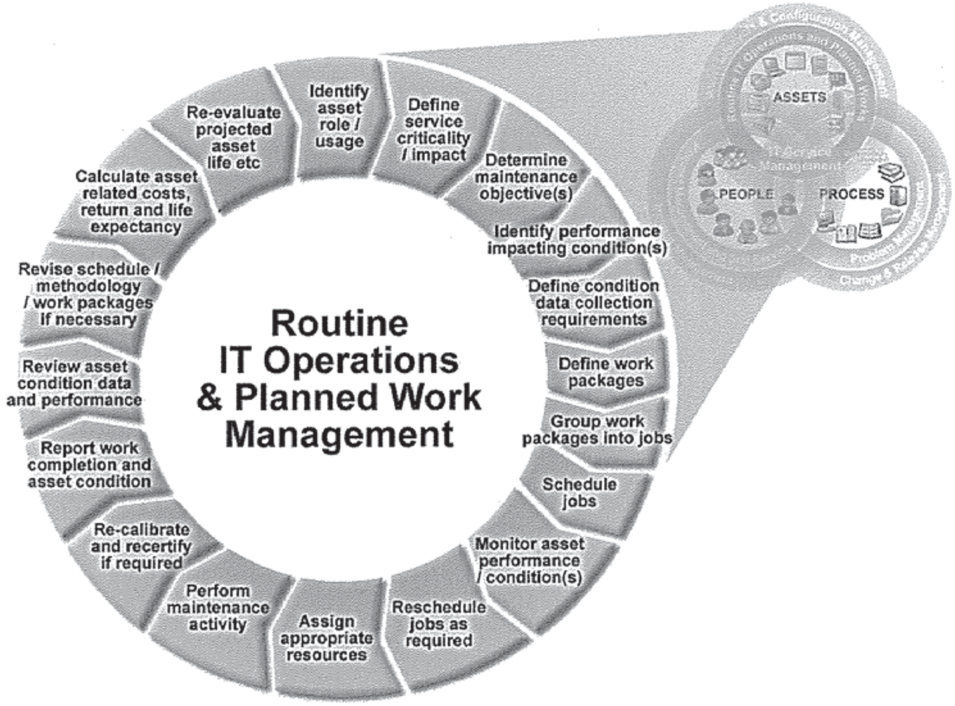

Routine IT Operations & Planned Work Management
Identify asset role / usage
Define service criticality / impact
Determine maintenance objective(s)
Identify performance impacting condition(s)
Define condition data collection requirements
Define work packages
Group work packages into jobs
Schedule jobs
Monitor asset performance / condition(s)
Reschedule jobs as required
Assign appropriate resources
Perform maintenance activity
Re-calibrate and recertify if required
Report work completion and asset condition
Review asset condition data and performance
Revise schedule / methodology / work packages if necessary
Calculate asset related costs, return and life expectancy
Re-evaluate projected asset life etc
ASSETS
PEOPLE
PROCESS
IT Service Management

Fig. 19.6. High level process overview for preventive maintenance

hours of continuous running, number of transactions completed, amount of file defragmentation detected, volume of data stored etc.

The following list of questions will help identify the preventive maintenance requirements of configuration items within the environment:

- Have proactive maintenance tasks been identified for all business critical IT assets?
- Has risk assessment identified potential risks that need to be controlled proactively?
- Is condition based or periodic task scheduling being used?
- If condition based scheduling is selected, how is the condition data being collected and analysed in order to determine when the preventive action is required?
- Is the frequency of preventive tasks optimised to deliver maximum benefit for minimum cost?
- How often is the effectiveness of the preventive maintenance plan reviewed and amendments made?
- Is the successful completion of preventive maintenance tasks tracked and monitored?

19.15.2 A Manager of Managers...

The trouble with much of the routine activity that goes on within an IT department is that no-one knows what the other guy is doing. In fact it is very common for no single person to have a complete picture of all that is in process. Niche tools have evolved over time to simplify the performance of routine tasks. However none of these tools, with perhaps the notable exception of elements of the Microsoft Operations Management (MOM) framework which is slowly starting to provide some level of cross-discipline visibility, allow users to have a holistic view of the IT operations activity.

If a single view of the planned routine operations could be generated it would help with many ITSM processes including change scheduling, impact analysis and incident diagnosis to name just a few. Such a holistic manager of managers would need to tap into the scheduling engines of the following toolsets in order to generate a picture of planned activity on an asset by asset and service by service basis:

- Patch application
  - Package distribution, automated installations etc
- Database maintenance
  - Archiving, statistic recalculation etc
- Vulnerability testing – sweeps
- Antivirus software – scheduled low level scans
- Automated discovery toolsets - audits
- File level operations/Storage management
  - Backups, disk defragmentation etc
- Batch transfers/ETL processes
  - Data synchronisation, reconciliation etc
- Report generation/Bulk print runs
- Back office transaction runs
  - e.g. Billing calculations etc
- Document management tools/Search engines
  - Content crawling, index generation etc

19.16 Ongoing Viability Assessment

Note: Sometimes referred to as an asset effectiveness audit.

A viability assessment can be considered as a periodic review of key infrastructure components (including hardware, software and systems) to determine when they have reached the end of their useful life and how to make best use of them going forward. Such an assessment may be completed individually on an asset by asset basis or on a group of similar assets. The key question to be answered by the assessment is "Is it worthwhile continuing to use and maintain the asset in its current role?"

In order to come to a conclusion the following factors should be considered and weighed up against each other before making a final recommendation:

- Cost/benefit analysis
  - Maintenance contracts
  - Remaining leasing costs
  - Preventive maintenance costs
  - Power consumption (direct and indirect)
- Availability of suitable spare parts
- Availability of required technical knowledge/skills to be able to support the asset effectively
- Recent reliability and availability metrics
- Current performance and capacity
- Upgrade potential
- Physical size (i.e. machine room space required)
- Power consumption/Heat generation contribution
- Risks associated with changing the current configuration

Recommendations regarding the asset may include:

- Continue using it within its current role
- Downgrade its role to less business critical functions
- Upgrade the asset to increase its capacity and/or performance
- Replace the asset with a more up to date alternative
- Adjust its preventive maintenance plan to improve reliability etc

It must be remembered that it is relatively common for seemingly obsolete and redundant hardware to be used beyond its recommended life for mission critical functions within even the largest organisations due to the perceived risks associated with changing. The old adage "If it ain't broke, don't fix it" is often cited as a reason for such technological inertia when in fact it is more likely to be due to the fear of the effects of change...

19.16.1 Estimating the Projected Remaining Useful Life

The remaining useful life of a configuration item can be determined by comparing the current utilisation and spare capacity of the asset against the projected future load required by the business. Useful life estimations should take account of the potential to increase capacity (be that processing power or storage capacity) through upgrade as well as the potential to use the asset within different, less demanding, roles or functions in the future.

19.16.2 Defining an Upgrade Path

Before deciding to upgrade a configuration item, the following questions should be considered and reviewed:

- Is it technically possible to upgrade the asset?
- Is it economically viable (based upon cost benefit analysis)?
- For how long is it anticipated that the upgrade will extend the useful life of the asset?
- What are the risks associated with performing the upgrade?

19.17 Stop the Upgrade Merry go Round – I Want to Get Off!

There is no rule written in stone that states that you must always upgrade applications to the latest and greatest release. Fortunately, IT management is less prone to jump in feet first at the request of the software vendors than it once was and the pain associated with the early adoption of new releases is less of an issue than it used to be. Every upgrade should be reviewed in terms of the anticipated business benefit versus the likely risk associated with the implementation of the upgrade.

19.18 Decommissioning

When IT budgets were plentiful, decommissioning was often limited to putting a screwdriver through a disk drive before throwing the equipment into a skip. Now those days of plenty are long gone and organisations can no longer be as wasteful with their valuable IT assets as perhaps they once were. Embarrassing high profile cases of corporate data loss have also meant that businesses now understand the need to ensure that sensitive information is diligently removed from all machines before they finally leave the organisation.

The decommissioning process should address the following questions:

- Have all transferable software licenses been returned to the organisation's software pool (after uninstalling them from the machine to be disposed of)?
- Is it worthwhile keeping key hardware components as spares for remaining equipment e.g. power supplies, fans, memory etc?
- Have all salvageable components been removed and entered into inventory?
- Has all useful data been backed up and transferred onto the replacement system?
- Have security policies regarding data cleansing been implemented fully?

19.18.1 Lease Returns

Where the equipment that is being decommissioned if due to be returned to the supplier under the terms of a lease agreement it is important to ensure that it is in as near a comparable configuration to its original condition as possible. The terms and conditions or lease contracts vary considerably and therefore it is important that the specific requirements of the agreement are adhered to if punitive penalties are to be avoided. It is fair to say that many leasing companies anticipate significant levels of defaulting on equipment returns and sometimes inflate penalty fines to recover monies they were unable to realise when the lease contract was signed. Seemingly attractive lease rates may have a nasty sting in the tail for the unwary IT manager and it is therefore critical that full end to end asset lifecycle management practices are implemented effectively.

19.18.2 End of Life Planning

It is rarely practical to simply turn off the tap when it comes to removing or replacing key IT infrastructure components. End of life planning should address issues such as user lock outs, cut over activities, parallel running, data backups, production data migration, network traffic redirection, IP address reassignment, license reclamation and support contract cancellation etc.

19.19 Cradle to Grave is no Longer Enough . . .

19.19.1 Environmental Impact and Safe Disposal

Environmental regulations, such as the Waste Electrical and Electronic Equipment (WEEE) regulations in the UK, are becoming more and more common around the world and are beginning to require businesses to take a more responsible approach to the disposal of IT hardware than previously. Some of the hazardous material lurking within your desktop PC:

- Antimony trioxide – Used as flame retardant within desktop cases
- Arsenic – Found within older Cathode Ray Tubes (CRT) inside aging monitors
- Cadmium – Used in circuit boards and semiconductors
- Chromium – Used in steel as corrosion protection
- Cobalt – Found in steel carcasses for structure and magnetivity
- Lead – Found within Cathode Ray Tubes (CRT) inside monitors
- Mercury – Used within switches and housings
- Polybrominated flame retardants – Found in plastic casings, cables and circuit boards
- Selenium – Found in circuit boards as power supply rectifier

19.20 RFID and Asset Management

Radio Frequency Identification (RFID) technology is slowly making its way from the lab into the main stream. Major retailers are beginning to use it in production environments as a tracking mechanism for pallets of goods etc within warehouse operations and the level of usage of the technology will only increase going forward. RFID as a technology has a variety of uses including:

- Asset identification i.e. barcode replacement
- Location tracking
  - Passive monitoring (short range fixed readers)
  - Active monitoring (RFID radar (currently limited to approximately 100 meters in range))
- Localised data storage i.e. tag based data stores
  - Equivalent of a log book to record servicing activity etc.
  - Store of information regarding current configuration settings etc.

RFID is sometimes hailed as the magic pill to cure all of asset management's ills. And yet it is often used a nothing more than a high tech replacement for printed barcodes. Such implementations gain little from the use of RFID and one has to wonder what benefits the businesses implementing RFID in such a manner are expecting to realise.

Asset location detection using RFID is not yet capable of finding the position of an asset to the level of accuracy of RTLS or GPS, nor does it have the range of other real time locationing systems. Passive readers can be effective at tracking asset movements from room to room or building to building but fail miserably to identify machines being moved from rack to rack, or blades moving from slot to slot within a server room. Since physical security measures and network discovery tools are reasonably effective at identifying/preventing unauthorised server moves between distinct locations, RFID currently occupies a "nice to have" spot within the minds of IT executives but lacks sufficient compulsive arguments to make it a "must buy" at present.

Given that using RFID as an asset identification tool only is a waste (as bar-coding gives the same functionality with less cost and all that is saved over manual operations is the time taken to key in a serial number or unique identifier and the potential for human error that is associated with this). It remains to be seen how this technology will be applied more fully within the IT asset management space...

19.21 Linear Assets and IT

Linear assets differ from traditional discrete assets in so far as they are continuous in nature. This means that it is often necessary to use spatial data to reference a specific element or segment of a connected system rather than a unique identifier. Examples of linear assets within an IT context include; network segments, cable runs, virtual environments spread across multiple pieces of physical hardware, distributed application architectures, load balanced web farms etc

19.22 Don't Forget to Play "NICE"ly Children

In the future, configuration management teams will operate in a manner similar to the UK's National Institute for Clinical Excellence (NICE) – This government body reviews the effectiveness and economic viability of clinical treatments for medical conditions and determines which treatments are to be provided free of charge to the UK population under the auspices of the National Health Service (NHS). The body comprises of numerous experts from all fields of the health care profession who meet to review the relative merits of one treatment against another. The purpose of the organisation is help ensure that the NHS's scarce financial resources are used effectively and that the general population derives the maximum benefit possible from the expenditure.

Future configuration teams will spend their time reviewing systems and solutions currently deployed/in use, identifying best practice and approving/authorizing hardware and software configurations for the organisation as well as determining the appropriate availability/distribution strategies. This sort of approach will reduce duplication of effort, diverging technology streams etc and will open the door to improved purchasing leverage as well as increases in reclamation and recycling processes.