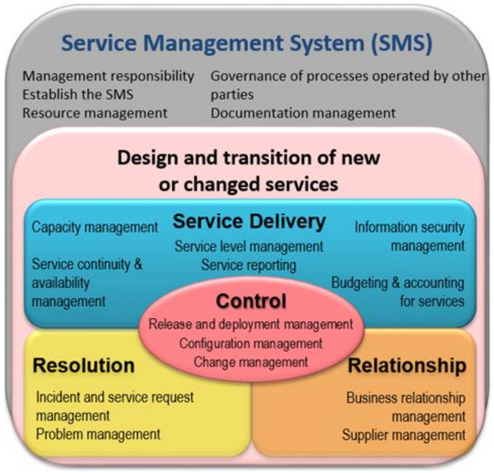
ISO 20000 IT Service Management

ISO/IEC 20000 helps organizations deliver quality IT Service Management through a comprehensive process approach. An IT Service Management System (ITSMS) provides standards on anything from Service Desk support to full-fledged IT system deployment. ISO 20000 offers internationally recognized best practices through an integrated process approach designed to meet the expectations of both businesses and customers. It sets forth the structural requirements for an ITSMS from service planning and service delivery to resolution and release management. ITSMS implementation is designed to be a bedrock upon which to build continually improving service management systems. It is fully scalable to organizations large and small or to customers internal and external.


Service Management System (SMS)
Management responsibility
Establish the SMS
Resource management
Governance of processes operated by other parties
Documentation management
Design and transition of new or changed services
Capacity management
Service Delivery
Service level management
Service reporting
Information security management
Service continuity & availability management
Budgeting & accounting for services
Control
Release and deployment management
Configuration management
Change management
Resolution
Incident and service request management
Problem management
Relationship
Business relationship management
Supplier management

1. Standard versus Good Practices

We see many different types of organizations attempting to utilize various best (good) practices, standards and frameworks such as ITIL®, COBIT®, Sarbanes Oxley etc. However, sometimes the notion of "adopting ITIL" can be overwhelming. If you'll simply build your IT offerings, based on the ISO/IEC 20000 standard, a lot of the rest falls into place naturally. Think of ISO 20000 as the roadmap and think of the frameworks, controls, etc. as a set of tools to get there. Since ISO/IEC 20000 requires continuous surveillance audits to verify the progress of your ITIL deployment, you'll have the motivation to keep the entire organization engaged and committed to quality processes over the long haul.

2. Get A Killer Differentiator

IT companies are in abundance everywhere. What makes you so different? ISO 20000 certification differentiates you from your competitors of relatively the same size and to stand out as higher quality from the big players. It works.

4. Because Your Customer Demands It

This is especially true on government contracts. As an industry, there has been little way to show consumers of our services that our technology offerings are built responsibly and on good practices. Many government contracts are requiring your organization to be certified to even bid on their work. Those that aren't requiring it will at least ask your intentions in regard to achieving it in the RFPs. This is your opportunity to get ahead of it. Don't wait for it to be required.

In summary, you should ask yourself, "**Can I help either ensure higher quality for my customers, or can I help the business sell more stuff by having this certification?**"

The ISO 20000 has a set of three processes for controlling our service: Configuration Management, Change Management and Release & Deployment Management.

With these processes we can control the configuration of the elements that make up our service (servers, software, etc.); we can control changes occurring in the service (change server, change of an agreement, etc.); and we can control deliveries that we send to our customer (What do we deliver? When? And under what conditions?).

Configuration Management

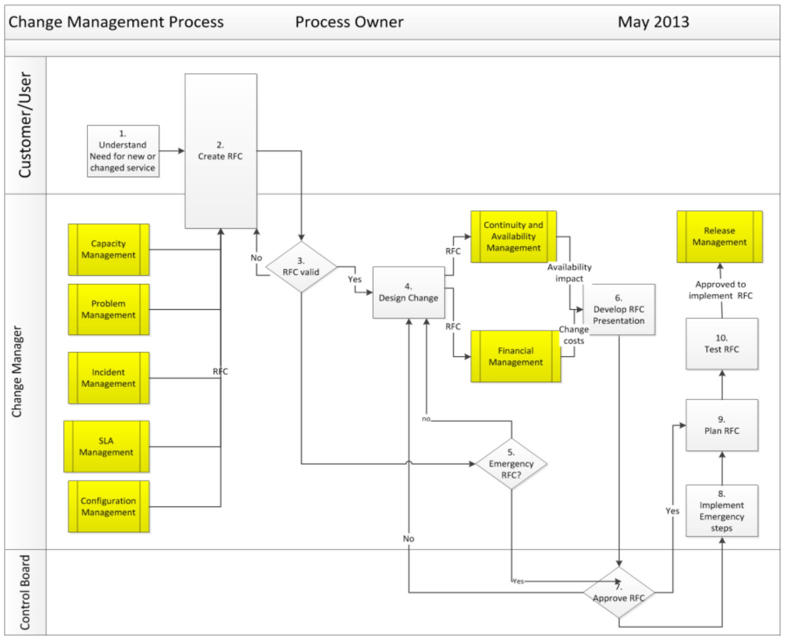
It is important to identify the elements that are used to manage the service (these elements, in ISO 20000, are called Configuration Items). For each Configuration Item we need to save information about it, such as name, version, responsibility, owner, dependencies (for example, an Operative System is related to a computer.), etc. The ISO 20000 does not define these parameters; they must be defined by us. For these Configuration Items, we will make a snapshot of its setup and this will be stored in a database called CMDB (Configuration Management Database). From there we can retrieve the configuration of a Configuration Item, if necessary. Therefore, it is very important to establish a procedure to periodically check the integrity of the CMDB, because the information that it contains is critical to the business.

Another important concept is the baseline, which can help us to control the configuration of the Configuration Items before passing them to a production environment. If we do not establish a baseline before moving into production, we risk losing the stable initial configuration, which may involve malfunction. Now let´s see an example: A server could be a Configuration Item, and the settings of its parameters (OS version, hardware model, HD capacity, memory, etc.), would be stored in the CMDB. Furthermore, this Configuration Item is related to software. Finally, if we want to transfer them to a production environment, we must first establish a baseline with both.

Change Management

This process is closely related to other processes (for example: Configuration Management, Release and Deploy Management, Capacity Management, Service Level Management), because it can be used to manage any change. For management, we need a workflow, which includes a Request For Change (RFC), which goes through an approval process. The RFC can be a form with fields to complete the information about the request (at a minimum: name of the person who requested the change, date of the request, priority, Configuration Items affected and the description of the request).

After the RFC is reviewed by a responsible party that will assess risks, impact and benefits for the business, the RFC can be approved. In this case, we can make changes, but there must also be defined guidelines to go back to the state before the change, in case the change has not been made correctly. Without this approval process, anybody can make any change, which can be a problem because we have no control in our documentation and in our management system.



All RFC must be registered and saved in our system. For example: We must change the server configuration, because it needs more memory. We need to generate a RFC, which is verified by a responsible party, and if approved, changes are made. (We also will need to change the information about the Configuration Item in the CMDB.) If the change fails at some point, it can be restored to the initial state.

Release and Deployment Management

All conditions for release and deployment of products must be established, and we must define a plan where all deliveries and deployments are planned at the time. Therefore, we need to define a plan for deliveries, which will contain dates, frequency and types of delivery. Furthermore, we need to define a plan for how to deploy the product in the customer´s system (installation of software, plugins, DLLs, etc.), and it is possible that we need to perform changes during the delivery, in which case we have to use the process of Change Management.

Also, we need to define a procedure for emergency deliveries, because it is possible that our customer needs the product quickly. You must have in mind that in any delivery or deployment, errors can occur; then, we must have defined a reverse procedure to recover the previous configuration. To avoid errors in a product environment, we must define a test environment, where we can try the product before performing the delivery. For example, we provide to our customer an updated version of our software, but we need to try it in our test environment beforehand. If all is good, afterward, we can install it in the production environment of the customer; if it fails, we must return to the initial configuration of the system (before installing the latest version of our software).

As you have seen in these three processes, the ISO 20000 provides useful tools to control our services. Can we manage a service without controlling the configuration of its elements? And without control of the changes? Or without control of the deliveries? Without ISO 20000, the answer to all these questions will likely be no, so we have a new reason to work with it!