

Finally, a Service Request Process



A significant enhancement to the ITIL Service Lifecycle v3 has been the addition of the Service Request process and the Service Catalog. All corporate internal IT organizations already have a Service Request process in place, but it may not be formalized, or as effective as their Clients wish it would be. Now with ITIL v3 the process is formally recognized and supported by a service catalog repository. Without recognizing Service Request as valuable IT business process, and appointing a process owner to manage it, a large IT organization can struggle in provisioning IT products and services to their Clients. As the corporate IT organization grows so does technical specialization. For a relatively small company, provisioning a laptop for a new employee may involve only 1-2 technicians, but for a large organization many different departments may be involved just to get a new employee set-up with the right hardware, software, tools, accounts and so on. Getting a multi-task service request, such as *new employee*, through a large organization not only needs to be completed in a timely fashion, but also effectively and efficiently.

Service Strategy

In ITIL v3, Service Request begins by defining your IT Service Portfolio, which includes the IT Service Catalog. The Catalog is the subset of the Portfolio that is visible to Clients. Each service listed in the Catalog must be fully defined and supported. From a Strategy perspective, the Catalog represents the virtual projection of the IT organization's actual and present capabilities.

Service Catalog Management

A major component of the ITIL v3 Service Design phase is Service Catalog Management. Service Catalog Management ensures the information contained within the Catalog is accurate and reflects the current details, status, interfaces, and dependencies of all the services being offered by IT. The Catalog acts as the central repository for all IT Services in which all areas of the business have an accurate view of the services being provided.

Elements of a Service

When creating your Catalog, there are many attributes or elements which define each service. Some of these elements are visible by the Client, while others are there for internal use and provide a complete picture of the service. Elements in which the Client can view are items like description, cost, service level or provisioning objective, approval requirement and maybe even picture of the product. These are important to a Client to be able to make a decision on the services they need and when they should be delivered. Internal elements of a service definition may be items like Service Owner, sequencing, and dependencies. A service may have a dependency upon another

service before it can be completed. For example, a desktop really cannot be installed for a new employee until an office has been identified.

The Service Request Tool

There are an ever increasing number of Service Management applications which can provide the customer-facing web portal to define and publish IT Services. Some applications provide a comprehensive solution across all the Service Management disciplines, while others are designed exclusively for managing Service Requests. In either case, an IT organization, of any substantial size, needs an electronic tool to support an IT Service Catalog and a Service Request process. A Service Request tool can

... an IT organization, of any substantial size, needs an electronic tool to support an IT Service Catalog and a Service Request process.

provide a standard for the definition of services, a workflow interface for service provider groups, a web-portal catalog presentation to Clients, and a *shopping cart* ordering interface.

The Uniqueness Service Request

Many companies who do not have a Service Request application, but yet have IT requests from Clients, usually force the process into their Incident Management tool. Incident records such as these are categorized as *Request* or *Order* to uniquely identify them. Plus, they are assigned an artificial severity rating to further identify them as a Service Request and not really an incident. This type of tool can work for simple single-provider requests, but in many large IT organizations the simple requests can involve multiple providers. Even a simple

request for a new software application can require approvals, software deployment, installation and account authorization which may be provided by multiple IT teams. So now a seemingly simple request has an organizational workflow which requires a sequential order of tasks to provision the new software application for the Client. In other words, Task A has to be completed before Task B, and Task B must be completed before Task C can be done. Most Incident Management systems just cannot handle these types of requests, and when they are *forced in*, it usually requires IT support staff to manually manage the workflow to ensure the request is completed.

Other types of requests, which can be a challenge for an Incident Management system, are ones in which tasks need to progress simultaneously. Take for example, the *New Employee* request. This type of Service Request can have tasks such as *new network account*, *new SAP account* and *new eMail account* which can all be provisioned concurrently by different IT teams. Here again, if the system is not designed to handle this type of request then it may require manual *hand-holding* to ensure it is completed correctly and timely.

Commodity -vs- Non-Commodity

Services that are performed on a routine basis, and which can be fully defined, would be classified as a *commodity* type of service request. These types of requests may be done on a daily, weekly, or monthly basis, where the service objective is repeatable and predictable, and the cost is consistent. An example of this type of request is an eMail Account creation service. Once a model account has been developed, subsequent accounts have minimal

variation, so the cost and service objectives should remain stable.

Other services, which Clients request, may not be as clear and predictable, and may require further analysis to provide an estimate on duration and cost. A request such as this may be classified as a *non-commodity* type of service request. An example may be a request for a new SAP Financial Report. When this type of request is submitted, the IT group providing the service should request specific information from the Client at the time of the initial request, in order to produce a cost and time estimate. This will require the IT service provider to analyze the work effort involved, to provide the Client with an estimate of cost and time.

A central web-portal catalog interface should be able to accommodate all types of requests, and minimize the need for Clients to go directly to IT staff for support.

A Service Request process designed to handle both *commodity* and *non-commodity* types of service requests provide the structure to offer a comprehensive set of services to your Clients. A central web-portal catalog interface should be able to accommodate all types of requests, and minimize the need for Clients to go directly to IT staff for support. All requests submitted through a web portal or Service Desk are assigned the appropriate priority, contain the correct information to provision, and are then assigned to the correct service team.

Finally, having a Service Request process as part of the ITIL Service Management Lifecycle, lays the ground work for IT organizations to formalize their process and contribute to its further refinement. Since Service Request is

truly a customer facing process, it needs to be the best it can be to support the ever increasing demands of the business.

I. Til (a.k.a. Kevin S. Miller) is a Sr. Consultant with Interphase Systems, Inc of Plymouth Meeting, Pennsylvania. Interphase Systems is an established IT consulting company that focuses on computer systems validation & compliance, and IT enterprise planning within the pharmaceutical, health care, and financial services industries. For more information you can visit www.interphasesystems.com or call 610-276-5500.