SysAid

Thinking About a Service Catalog,

But Your Organization Isn't Ready for It?

By Greg Sanker

A service catalog is important for a customer-focused IT organization. But creating and maintaining one can prove more difficult than generally thought. This white paper offers some practical guidance on how to get started with a basic service catalog, even if your company isn't ready for a full IT service management (ITSM) program.

What Is a Service Catalog?

A service catalog is just a listing of available services, right? So what's the big deal? Unfortunately, the answer is yes and no. In its most simplistic form, a simple listing of all services offered by the IT department is a service catalog, but if you dig into the ITIL® books, it seems a lot more complex. One key reason is that **service catalog management** is a process - not a document, and it's part of a broader **service portfolio management** process that together spans several of the ITIL lifecycle phases.

While a simple service listing can be helpful, it's often created and forgotten. It's frequently the result of someone in the support side of the house, posting it on the website in an effort to be more "user friendly". The folks in the technology teams may not even be aware of the list, and rarely feel any ownership in it.

In all but the most diligent of organizations, these types of service listings get out of date quickly, and when they're out of date, people think of them as irrelevant, and a well-intended effort goes by the wayside. Without the support of other ITSM processes, like change and release management, configuration management, knowledge management, and the rigors of service design, the seemingly simple service catalog becomes very difficult.

So, what do you do when you know you need a service catalog, but your organization isn't ready for a full ITSM program, or when you're having trouble with cultural resistance to the "we need a service catalog" approach?

Have You Ever Played Telephone?

You know that game many of us played as children, where you sit in a circle and one person is given a short phrase whispered in their ear? They, in turn, whisper it to the person next to them, and so on around the circle until the last person finally says aloud what they heard.

Rarely does the phrase come out at the end the way it started. Everyone has a good laugh, as they compare what they heard at their point in the chain. The group marvels at how the phrase changed so much in such a short time. The more detail in the original message, the greater the likelihood of errors. A phrase like "Sarah likes to eat ripe bananas with her coffee after a brisk morning walk in the cold fall air" can turn into "I sort of like bananas with brisk ice tea."

It's all great fun because there's no harm in the communication breakdown. It's a lesson in the importance of clarity in communication.

But what if it's something important? Something like… details of how an IT service is configured, or available options?

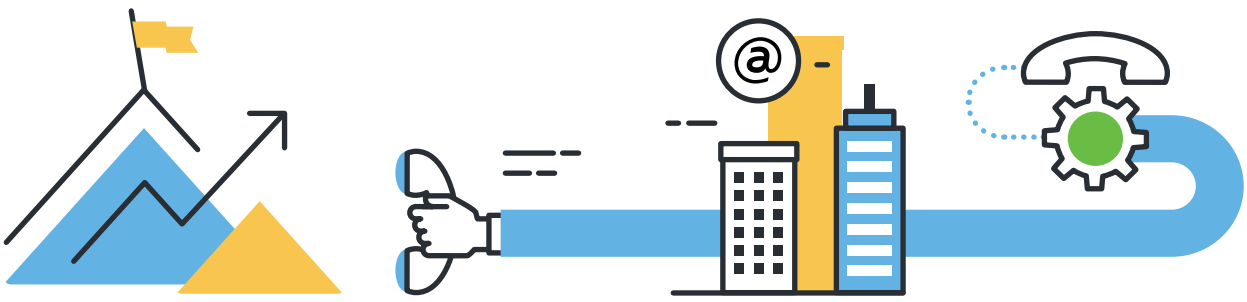
The game of telephone is probably not the best model when the details are important. And yet, so much of what we do in IT is documented and communicated in this very same way. Or, more correctly, UN-documented and NOT communicated.

It's a recipe for confusion and frustration.

Growing Out of Confusion

Unfortunately, this kind of confusion is far more common than any of us might care to admit. Technical people are not known for their love of documentation – neither creating it in the first place, nor keeping it up to date as things change.

It's a known weakness in the world of IT, and because of it, we've developed a system of tribal knowledge to compensate. It's a complex system of local experts and historians who know who did what, when, and why they did it.



You'll recognize it by the types of conversations required to answer even the most basic of questions:

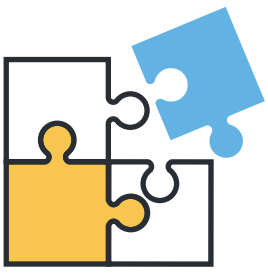
> **Sarah**: "I'm setting up a new mailbox. Remind me what the default message store size is again?"
>
> **Jonas**: "250 MB for the standard user, 500 MB for executives."
>
> **Kanish**: "No, it's 500 MB and 1 GB."
>
> **Jonas**: "When did that change? It's always been 250 MB."
>
> **John**: "No, we changed it when we upgraded the storage array in November, except that I thought we decided to just make all mailboxes the same size."
>
> **Sarah**: "500 MB or 1 GB?"
>
> **John**: "2 GB"

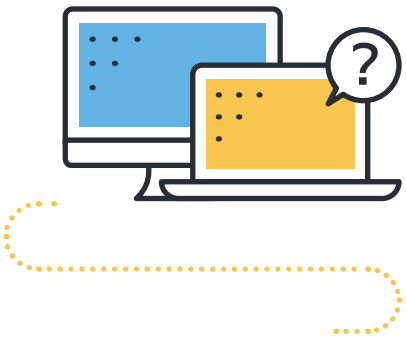
A lot of IT organizations have incomplete, conflicting, or out-of-date documentation about the services they offer. It's bad enough for IT staff to have to wade through this sea of uncertainty, but when it spills over to end users, you start having some serious customer satisfaction issues.

This is where the idea of a clear and concise listing of services usually comes up – let's put together a list of all the services we offer. It's a good idea, of course, but in an organization that's more focused on technology than services, that's also where the troubles begin.

If you know you need such a service catalog, but you only have the most basic parts of an ITSM program in place, you may face some challenges:

- Staff may think: "Our users know what we do"
- IT services evolve too rapidly
- Changes are made with little formal change management
- Siloed organization
- Poor/informal communication between teams - app dev, infrastructure, and support
- "We are technology providers (not services)"
- Lack of service owners
- Thinking *service catalog* is a static document
- "We already have one…somewhere…"





What Is a Service?

In order to build a service catalog, we should start with talking about what a service is.

Here's the ITIL definition:

> *"A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks."*

Which is a great definition that's as academic as it is accurate, and leaves many of us who are just getting started asking: ***"but, what is a service?"***

What we need is something that's easier to understand in an organization that's perhaps been more focused on the technology, and less on what business processes and problems those technologies are solving.

Every organization has services, whether they know it or not. Some services have been around for so long, we just think of them as institutional fixtures. I call these *common law* services.

Everyone knows what they are and if you try to change them, or take them away, you'll have angry users beating down your door demanding their service back.

That being the case, those very users are the ideal place to start – they know what your services are. At least they know the ones they use every day.

Some they like; others they have unkindly nicknames for. Either way – why not ask them? Or at least look at it from their perspective.

Services should:

- Be recognized by users as a **solution**
- Support a **business process**
- Include all the **technology components**

For example:

Let's take an easy example – email. For many organizations, the email service is provided by the Exchange® servers. But most users see email through the Outlook® client software. So is it the Exchange or Outlook service? But wait, the Outlook client has to run on a PC, so the service requires the Exchange server, the Outlook client, and a PC (or laptop). But if the PC isn't connected to either the corporate network, or through some remote service, it can't access the Exchange service.

The typical Exchange and Outlook combination is also used for group calendaring and perhaps shared/public folders.

That being the case, then, what's the actual service? Going back to the points above, users generally view Outlook as a solution to their need for electronic communication that supports the business processes of interactions with other employees, external partners, and customers. They also see it as a solution for: scheduling meetings, conference rooms, and other resources; as well as supporting the business processes of facilitating office and customer logistics.

As you can see, the service is more than simply *Exchange* or *Outlook* – it's an electronic messaging, communication, and collaboration service.

And this brings us to a key point. **Services should be named to describe the business functions they fulfill – not the technologies pieces that support them.**

Identifying Your Services

Recognizing that users are the key to services, here are some ideas to help identify your services:

- Look through your service desk incident data. It's really hard to fix a problem that isn't associated with something a user is trying to accomplish with a piece of technology. Whatever that is, it's likely a service.
- "I need access to…" – things that customers ask to access are nearly always a service, or part of a service.
- Everything that's included in setup for new employees – if you have an onboarding checklist, it can be a good source of 'service' candidates.
- Common language – every organization has it's own language or lexicon for IT services, things like:
    - remote access
    - accounts payable system
    - local names like "the BEAMR system"

Reverse Engineering

As you're trying to begin building your service catalog, be on the lookout for opportunities to create clarity:

- Moments of confusion
- Customer questions
- Conflicting information
- Services that have evolved over time

When these kinds of situations come up, don't hesitate a moment - use these tribal knowledge conversations as golden opportunities to create some clarity.

The email box size conversation is a great example of moments of confusion and conflicting information. When a simple question about details of a service or how it's provisioned or configured requires several people to get a satisfactory answer, you have the opportunity to create clarity. It's a bit of a "write it down so next time we'll know" approach. And that's the heart of reverse engineering your services.

Reverse engineering is a term used to describe the practice of taking an object apart to see how it's built – usually for the purpose of creating a design for building a similar product. In our case, we're going to use it to figure out exactly how our own services are built, so we can create a single, clear description.

Think like an engineer and ask detailed questions about the service in question:

- Details that are in dispute (is it A or B?)
- How are the parts connected?
- Are there other parts to the system (security tokens, remote access, cellular devices)?
- What infrastructure services are required (account administration and authentication, networks, firewalls, storage servers)?
- What business processes does it support (accounts payable, receivable, general ledger)?
- Hours of operation (24x7, business hours); are there maintenance windows?
- How do customers get the service?

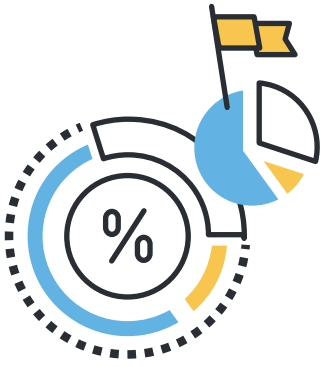
Sources of information include:

- Asking the IT staff who were involved with building and supporting the technologies
- Talking with users of the service
- Reviewing existing documentation
- Looking through old emails
- Leveraging the service desk's knowledge base and common questions

You're trying to piece together the whole story by getting as much data as possible from as many sources as possible. Each unique view gives valuable insight into how the service was originally envisioned, designed, built, and supported.

When pieces of information conflict, as is often the case, it's frequently because each source remembers details at different stages of the services' history. Like the email box example above – different people have different recollections of the size. None of which were wrong, just outdated.

This is why reverse engineering can be such a powerful tool.
In the end, you'll have a complete picture of the service, and there will be little room for disagreement. The result of your research is that, in essence, the tribe has spoken, which goes a long way toward getting buy-in from the various teams.



Creating a Service Description

Armed with this initial information, it's time to draft a *service description*, which is a plain language description of the service, and should address:

- Summary of what the service does, and why users would want it
- Detailed description that describes:
  - Business processes supported
  - Business outcomes facilitated
- For whom the service is available
- How the service is requested and provisioned
- Costs and approvals
- Service dependencies/requirements (e.g. requires a laptop PC)
- Available hours
- How to get support

Be sure to:

- Directly address issues that have come up
- Describe the service in plain language that users will understand
- Describe the service as the user sees it
- Define the service timeframes (available hours)
- Designate the intended use – what business processes are supported
- Target users (company wide, divisional – marketing, manufacturing)
- Describe how the service is configured (e.g. mailbox sizes, retention, backup, and recovery)

Once you have an initial draft, use it to facilitate conversations with both business users and IT staff. This part may require a thick skin, because, whereas it's hard for people to tell you what it is when you have a blank page, it's much easier for them to tell you what's wrong with the way you have it written. And this is exactly what you want. Take all the feedback you can get and revise the draft.

Basically, you're facilitating a tribal knowledge conversation to create clarity. People who know the details of a service are usually happy to share their knowledge, and it's pretty rare for technical people to not support having clarity.

Through the process, focus on:

- Shared interests in clarity
- Engaging stakeholders in the process of developing service descriptions because it:
  - Leverages tribal knowledge
  - Increases ownership and buy-in

Continue the draft/revision process until you can get a broad group to agree that you have it right.

An Example: The Messaging and Collaboration Service

Here's an example of a service listing for the messaging and collaboration service:

Service: Messaging and Collaboration Service[GS2]

Summary: Provides email, calendar, and collaboration tools.

Detailed description: Industry standard email and calendaring tools for all company employees and authorized contractors who communicate electronically with internal and external users.

- Email
- File attachments
- Corporate directory of users
- Personal and shared calendars
- Meeting scheduling
- Shared folders for team collaboration
- Standard message storage – 500 MB

Options:

- Secure message exchange (encrypted email)
- Executive message storage – 2 GB
- Web access to corporate email from any location

Hours of availability: Service is available 24x7, with a 2-hour maintenance Sunday at 1:00am – 3:00am.

Related services: MS Outlook client and MS Windows PCs or laptop (limited support for Mac). Requires remote access service for telecommuters and travelers.

Costs: $5.00/month per user

How to request: Please call the service desk to request the service.

Building a Basic Service Catalog

After you've complete the hard work of creating the service descriptions, building the service catalog is relatively easy. The basic structure is to group services together under headings that make sense to the users.

The structure shown below is applicable to any implementation of service catalog, whether web-based or in an ITSM tool. Once you've done this work, implementing it in a tool is actually pretty easy.

Things to keep in mind:

- Always group services whenever applicable, as customers would expect
- Include users in the process as much as possible
- Make your service catalog easy to find and search
- Avoid naming services specifically by product names (MS Outlook, etc.)
- Use descriptive names (to avoid confusion and aid in clarity!)

In the following example, the headings are the categories under which services are grouped, with a short description of the types of services that are included. It's important to use descriptions that make sense to your users – even if they're not technically 100% accurate; you want to make it easy for users to find what they need.

Email, calendaring, and collaboration services

Electronic messaging, calendaring, and personal productivity apps

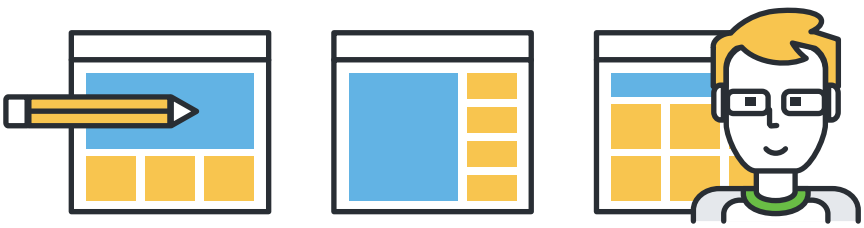
Client computing services

Centrally managed desktops, laptops, and kiosk computers

Phone and voice services

Desk and mobile phone and audio-based services

I'm using an HTML drill-down format, so if you click on "Email, calendaring, and collaboration," it expands to show the available services. Here, the short descriptions are directly from the service descriptions you developed above.



Self-service spam management

Provides tools to manage spam email

Company email

Provides corporate email address, email, and calendaring

Email distribution list

Create and manage corporate email distribution lists

If you click on the individual service, the full service description (like the messaging one above) should be shown to give users a complete understanding of the service.

In my practice, I've noticed that one of the biggest challenges in implementing a usable service catalog is overcoming the fact that IT people often think differently about technology services than customers do. It's very important to get customer feedback on the service catalog. If users expect email to be called 'email', don't call it 'messaging' in the catalog. Likewise, if they only know the accounts payable system as 'BEAMR System', guess what – you need to list it that way in the catalog.

You also want to get feedback from the service desk on the kinds of questions users ask about services they are looking for. If they ask about details that aren't covered in the service description, it's a good idea to add it. As I'll mention below, you'll need to periodically review and revise service descriptions.

Keeping It Current

Here we get to the final challenge you'll face in implementing a basic service catalog – keeping it current. The key is to tie into any processes that make changes to production services:

- Change management
- Project management office (PMO)
- System/software upgrades
- Implementing new features
- System fixes that change the behavior of the system

If you're fortunate enough to have a change management program, make sure you're included on any email distribution lists and attend change advisory board (CAB) meetings regularly. As changes are discussed, be on the lookout for changes that will impact any aspect of the service from a user perspective. This includes adding or removing features or changing any service parameters.

Ask project teams to create/revise service descriptions as needed for new or changed services. The service desk should expect updated service descriptions as part of the new/revised service onboarding. As you're building support for your service catalog, you may have to make these updates yourself. The key is to make the catalog part of daily operations. Any time there are conversations about service details, pull out the service catalog and include it in the conversation. Follow up by making any necessary updates.

The service desk should keep track of times users are either confused with a service or experience something different than described in the service description. **Never waste a good moment of confusion to help create clarity.**

In addition to tying in to these processes, you should have a scheduled periodic review of the service descriptions. The review is intended to catch anything that was missed, or any updates that are needed based on customer and staff feedback.

Congratulations – It's a Service Catalog!

It really is possible to create and maintain a service catalog in any IT organization. Leveraging moments of confusion and lack of clarity creates the win/win opportunity to engage subject matter experts and get tribal knowledge out of their heads and into documented service descriptions.

If you're willing to play the role of detective, be on the constant lookout for any changes in services, and facilitate conversations with updated service descriptions, you'll no doubt succeed with keeping your new catalog up to date. The more the service catalog becomes part of daily operations, the more people will come to refer to it and rely on it. The more reliance, the more acceptance.

Plus, you've now done the hard work, so you'll be prepared when your company is ready to implement the service catalog in an ITSM tool!