i-doit®

The six steps: How you start with your IT documentation.

The six steps towards IT documentation.

www.i-doit.com

Content

Foreword

You're now starting a project that will be keeping you busy for the next few months: your IT documentation. Since you're using i-doit you can amend the project name slightly: your centralised IT documentation. This minor change implies a number of necessities but just as many opportunities: after all, something centralised is intended to be used by several people – and these stakeholders will need similar expertise and skills as well as a few organisational rules.

First things first, however, and the question we hear more than any other: How do I get started? And our standard answer: There IS no standard answer, since every company is unique. The specifics of your situation need to be analysed, and a configuration management plan (CMP) can then be developed in workshops. This CMP sets out the agenda and the necessary degree of detail for your individual situation.

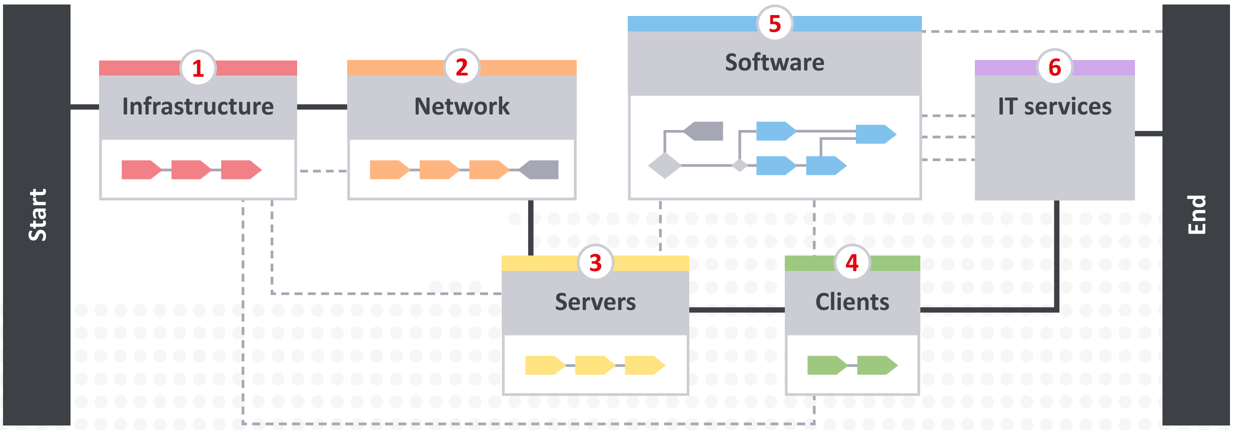
After all, regulations for data centre operators are not the same as those in the public sector, and admin workloads will differ between SMEs and cloud providers.

Nonetheless, there's still an optimum logical sequence that avoids repeating any steps in the process. The sequence also offers us a trip through the opportunities offered by IT documentation. And even if you don't have a specific item on your agenda: just try it out as part of a pilot project. As you'll see: sooner or later, you'll need it.

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

The six steps towards IT documentation.

1. Infrastructure, 2. Network, 3. Servers, 4. Clients, 5. Special case: Software and licences,
6. Server applications and IT services



Step 1:
Documenting the physical infrastructure

www.i-doit.com

Step 1: Documenting the physical infrastructure

“Don’t search – find!” is the motto here. If you’re supporting multiple sites – or even just multiple rooms – you’ll know the problem: So where is X again...?

Whoever you ask in the company about your IT documentation, the answer’s not slow in coming: Why should I know where it is? Since the location is often required for all of the documentation tasks that come later, it’s a good idea to get this straight at the start. The information is static, and changes are few and far between. So time spent here is time well spent.

So why not use i-doit for your next inventory? The end result will not only be an up-to-date overview but you’ll also be documenting many things for the very last time!

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

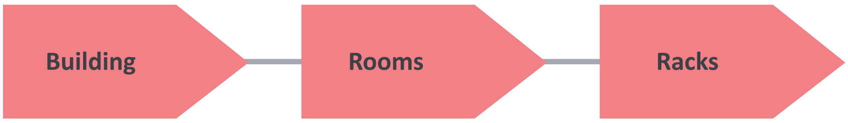
Location-based documentation made easy

Assume that each object that corresponds to a location now needs to be documented once and once only. All other objects are linked to it relationally. All of these can be physical locations:

- A chassis (to ‘locate’ modules correctly)
- A rack (into which chassis or servers are installed)
- A workplace (with which equipment is associated)
- A room (which contains racks or workplaces, for example)
- A floor (which contains rooms)
- A building (which contains several floors)
- A campus (where buildings are located)
- A city (in which the building or campus is located)
- A state
- A country

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

Seems complicated? Well, the basic version, for a straightforward computer room configuration, would be as follows.



Racks are located within a room and rooms are located in a building. Everything else is left out to start with.

Since each element is only created once, you can spend a little more effort on this one-off documentation job. Other properties should be added in addition to the name – such as the exact address, the contact persons (for access or alarms), the telephone number of the extension in the computer room, etc. Always have the worst case in mind: if worse comes to worst, it won't be you who needs to sort things out, but people who may never have seen or visited the facility!

What should be done with missing details?

Often, companies inform us that some details are missing: rooms don't have unique names or there isn't a room number system. Accordingly, the assumption is that documentation cannot continue for these locations. So the project seems to stop before it even gets going.

Our tip: get started anyway! Make up a naming scheme if you have to, but don't wait around – the worst case won't return the favour! Use aliases and add a human-readable explanation in the comments field if necessary ('second room on the right next to the lift on the first floor, room number unknown'). For server rooms where external labelling is prohibited, you can post a notice on the inside of the door instead:

This room is i-doit room # r1105.

(second room on the right next to the lift
on the first floor, room number unknown)

In an emergency, it all comes down to communication…

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

So what are the next steps after this?

There's plenty that can be done once the locations are documented:

Floor Plans

The i-doit Floor Plan add-on gives you a graphical representation, and photos (e.g. of building plans) can also be stored in the system. Later on, you'll probably want to mark out the exact locations of systems, racks – as well as your Wi-Fi coverage.

Rack management

We'll soon be coming to the equipment slotted into your racks. If you also locate these correctly within the rack, i.e. with the right unit details, you'll be rewarded with a graphical representation reflecting the reality of your server room. Finally!

Facility management

Don't restrict your management to just rooms: your inventory should include your workplaces, users, extensions, tables, chairs and fire extinguishers. Initially, you should of course focus on your mission-critical IT components!

Documenting your building systems

Fire extinguishers, UPS, EPS, air-conditioning units, etc. all have one thing in common: an inspection label (often issued by the TÜV here in Germany) with details of when the next equipment inspection is due. Taking steps to ensure these deadlines are met may be important for reasons of insurance, legal compliance or simply the basic needs of your business. Why not have i-doit remind you of these dates automatically? Or simply create an automated report listing the inspections that need to be planned for the following month.

Cable management

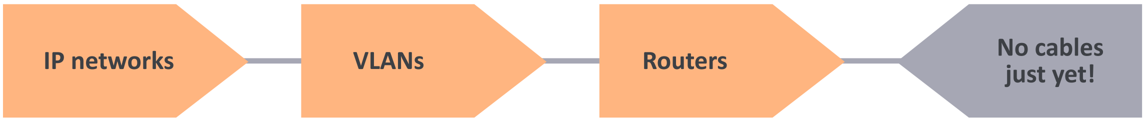
Cable joints often need to be documented. For patching work in particular, this is a great help when answering the perennial question: Which cable can I unplug? i-doit knows the answer. And i-doit doesn't care if you're mapping power or network connections. But is this the right time to start? In our opinion: No. More on this anon.

Step 2:
Network

www.i-doit.com

Step 2: Network

What's the minimum we need to obtain a pragmatic set of network documentation that, while it doesn't go into unnecessary detail, is still sufficient to let us access all of the information we need at a later point in time?



IP networks (OSI layer 2 + layer 3)

IP ranges are documented as separate networks. After all, these networks were planned in detail, implemented with great effort and are often to be found on outdated Visio diagrams. Our recommendation: Put an end to your file-based documentation and get everything into the i-doit database so that future use cases can properly benefit from the information. Later on, the other network devices – whether physical or virtual – will connect to these networks. The end result is a complete overview of assigned and unassigned IP addresses, a summary of ports used, and the ability to execute a wide range of queries and automated activities on your equipment directly from within i-doit (ping, nslookup, custom scripts,…).

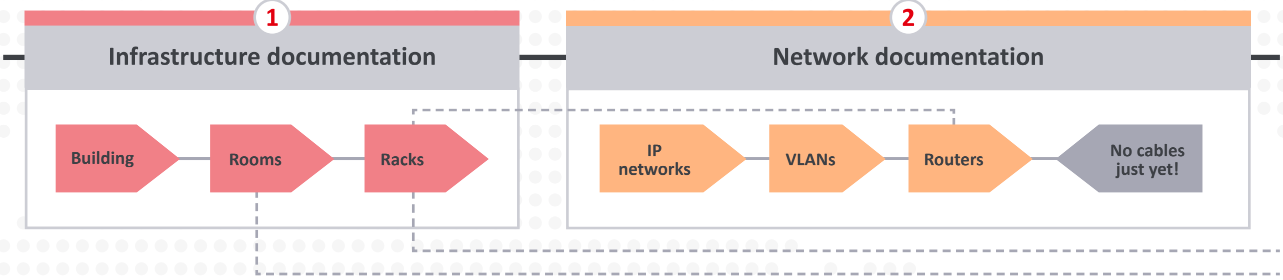
VLANs

These are increasingly found in organisations, and need their own management and documentation if you want an up-to-date overview of the correct configuration of routers and user devices. VLANs are connected to the corresponding IP networks.

Routers

These, i.e. layer 2/3 routers/switches, are then 'slotted into' the racks already documented, i.e. located in your documentation. Entered for the correct units, a view of the rack is built up that also accurately reflects the physical reality. Similarly, you now connect the networks together via the correct ports.

The figure below shows the interconnections:



Cables?

The initial equipment and networks are now documented and connected up. So is this now the right time to start documenting the cabling? It might seem tempting to get started on cabling here, but this still isn’t the right point in time. Not least because getting cabling documentation right needs plenty of time, planning and patience. But we still need this time to move forward with the overall documentation.

What are the possible branching points here?

With the networks now documented, the next steps are as follows:

WAN connections

Particularly in the event of faults, it’s very important to have the right contact persons and hotline numbers to hand. Typically, you’ll also need:

Contracts

Contracts with providers, incl. service level agreements and the agreed fault resolution processes can be managed as an attachment. As mentioned previously, here is also the documentation for the Monitoring integration.

Monitoring integration

Integrate your existing network monitoring or set up a new system. i-doit gives you the support you need:

- Live status updating between monitoring and CMDB
- Definition of the monitoring points from i-doit, enabling automatic configuration of the monitoring system.

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

Step 3:
Server documentation

www.i-doit.com

Step 3: Server documentation

Everyone has them, but they are rarely properly documented. Initially, the level of detail isn't important: brief but up-to-date is better than detailed and obsolete! Documenting your servers is a job that simply has to be done, since sooner or later the following situations will come to mind:

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

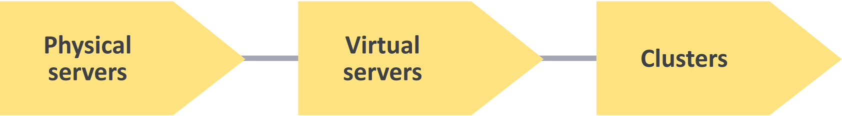
Outages and service continuity
Worst-case scenarios are very rare, but the server configuration will have to have been documented. With some disasters, recovery is not a viable option. Maintaining up-to-date business and service continuity planning is now mandatory in many industries and organisations.

Migrations, updates and virtualisation
If major changes are planned in the server and application environment, you'll actually need two sets of documentation: the actual set (for the current servers) and the future target set (for the new servers).

Application and service documentation
Since these run on the servers, it's advisable to already have the servers in your dataset.

With documented servers, you have the option of going 'deep' into the details as well as 'up' into the world of services and applications.

In terms of the order, we once again recommend first documenting the objects that might end up being reused – so hardware first and virtual devices afterwards. And clusters consisting of multiple servers after that.



Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

So what should be part of every server documentation?

Let's assume the worst case: a server needs to be set up again to provide a specific function in the network, the server backup cannot be used, and the individual who set up and then regularly maintained the original server is also unavailable. A team has to be improvised, and this team needs to be able to read, understand and apply the documentation in the shortest possible time frame.

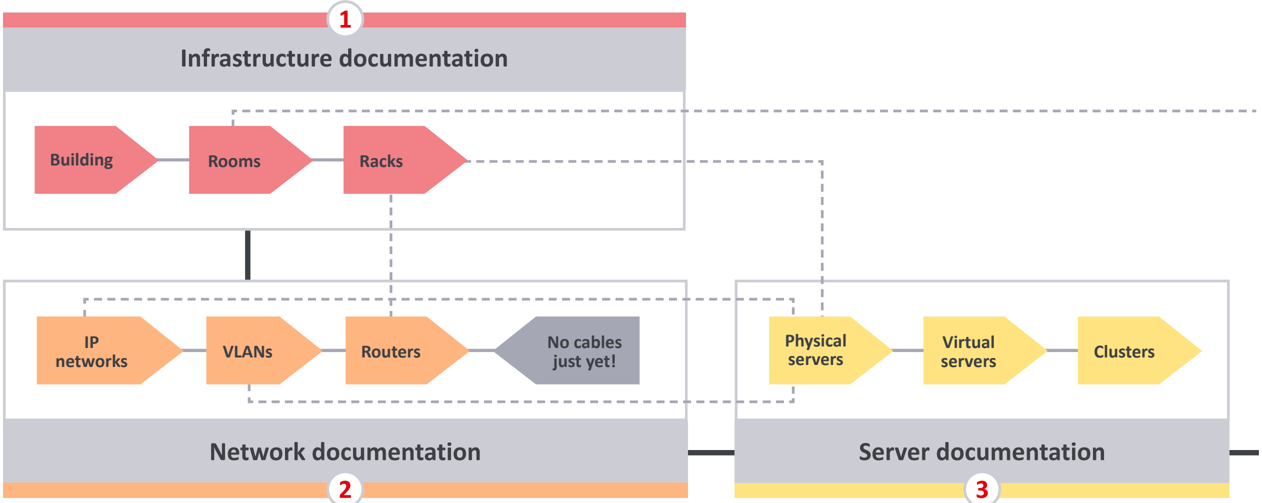
Alongside the hardware configuration, physical or virtual, the base version from which the changes were made also needs to be known. Following this, the individual features of the device must be specified. Anything that is designed to provide a specific function but is not part of the standard is worth documenting.

- Hardware details
- Location – a couple of mouse clicks, thanks to our preparatory work
- Network connection – equally simple
- Contact person
- Storage configuration
- Operating system, software, versions. By this point, it's advisable to consider making use of automatic discovery. Documenting the latest version of all this software manually is time-consuming – and the time is better spent elsewhere!
- The same applies for storage connectivity: it's not an easy task to complete manually if the data cannot be provided automatically.

Detailed server documentation

The depth of detail can be very deep indeed: here are a few ideas about what is still left to document:

- Administrative accounts used (local or in the directory)
- Passwords used (with link to the password safe entry or search criterion, or stored encrypted in the CMDB)
- Modifications to files – the best option here is to store the files or the deviations from the standards for these files within reach of the CMDB
- Modifications to the configuration (operating system's own firewall?)
- Installed software (ideally including installation screenshots for all individual steps)
- Installed licences (best to simply document these in the CMDB; here, too, automated reminders about upcoming expiry dates is truly a key benefit!)
- Changes to the configuration of the installed software – either with screenshots or with configuration files; if stored chronologically, this helps others to understand the steps that have led to the current configuration.



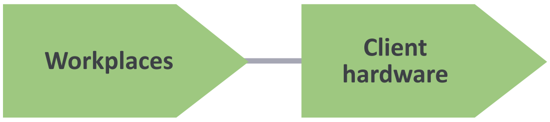
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

Step 4:
Clients

www.i-doit.com

Step 4: Clients

Before we tackle the documentation of the client hardware or peripherals, please take a minute to consider the concept of the documented workplace.

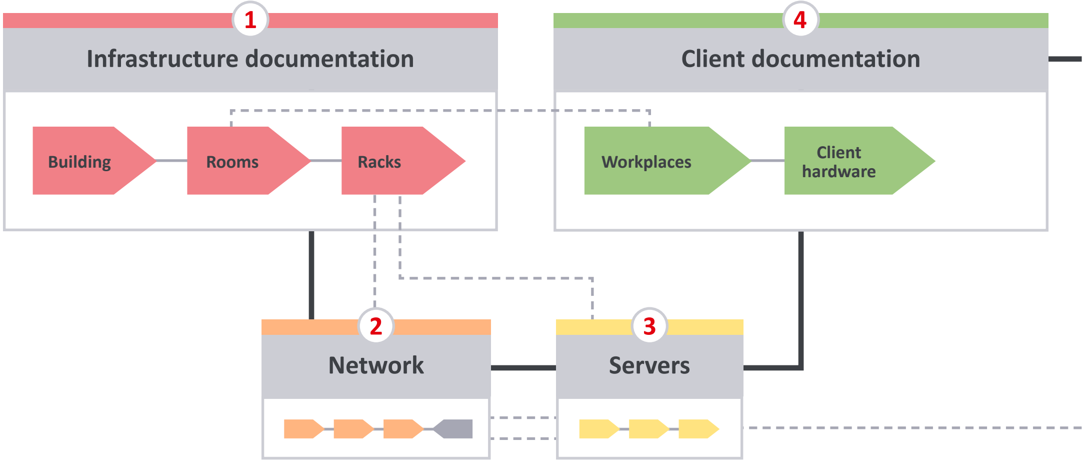


A workplace is a kind of 'virtual location name' which can, in turn, be used for linking objects to. Especially in this age of increased mobility, this lets us distinguish between mobile and fixed equipment. An example:

A workplace is ...

- assigned permanently to a location (room);
- allocated to one or more people;
- allocated to one or more departments; and
- a kind of 'container' for permanently installed equipment – printers, phones, scanners, monitors, docking stations, chargers and permanently installed PCs.

Another advantage offered by using workplaces is the assignment of network ports (patch boxes), smoke alarms and other items of operational equipment. Workplaces can also be marked permanently on the floor Plans. Another use case is documenting home office workplaces. We recommend that you define workplaces before starting on the actual client documentation. The context can again be understood by a look at our flow chart:



Asset management

While documenting or carrying out a client inventory, it is advisable to familiarise yourself with the methods used in asset management:

- Issuing sequential numbers for equipment that are unique company-wide
- Printing out and affixing labels for straightforward identification
- Use of barcodes to enable simple scanning with barcode readers
- Use of QR codes for additional data, use of tablets and mobile devices as scanners, and using hyperlinks to the i-doit objects.

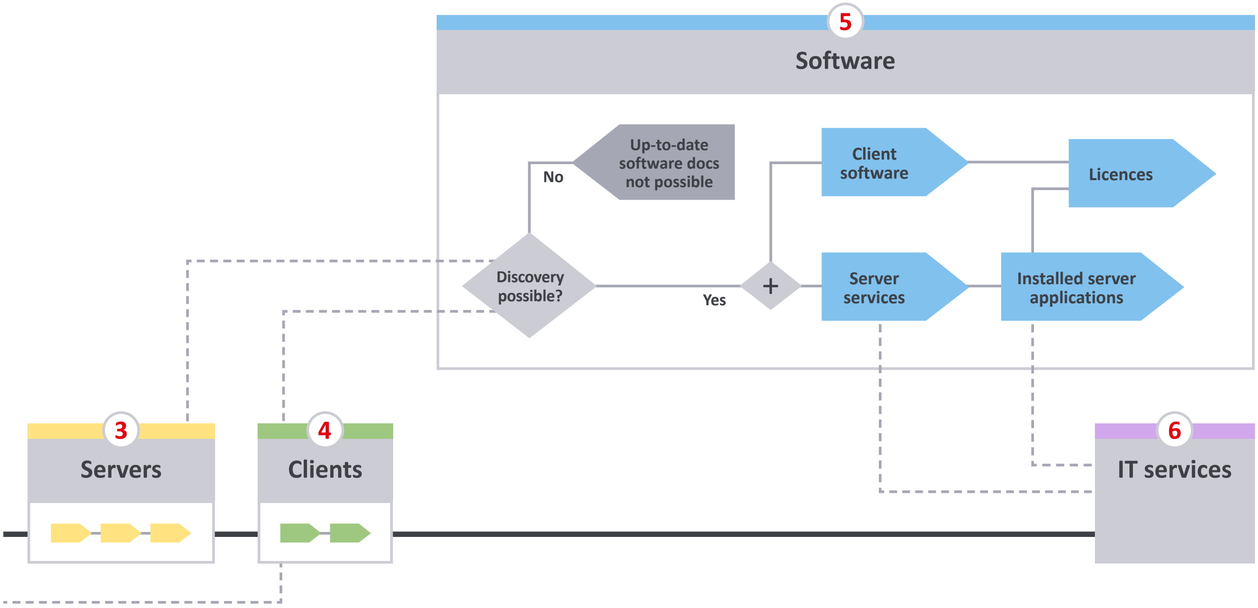
All these techniques are supported by i-doit: a little investment in making your system easier to find will translate into real time savings in the future.

Step 5:
Software – a special case

www.i-doit.com

Step 5: Software – a special case

Unless you do not use any proprietary software at all (and we have yet to meet that company), you'll certainly be interested in the subject of licences and installations. And if you ask your boss about the reports needed from the new IT documentation, then 'Licences' will be near the top of the list. There's a lot of money at stake here. Fees for licences bought – and fees for licences that should have been. And, of course, fees for licences no longer even needed. Keeping track of things here requires one thing above all: documentation. There's no doubt that manual methods will only get you so far here – this is really the point at which you need to look into acquiring some kind of automated system. But not to worry: the licences for these systems are very inexpensive and freeware is also available for some kinds of tasks. We illustrate the connections in our process flow.



From our own experience and that of many of our users, we know that without discovery, documenting the installed software, or even the physical clients, printers and servers, is simply an impossible task. The expenditure and effort required to acquire and deploy this software is minimal. And when the licence fees and the output are compared to the manual effort required to achieve the same results, it’s unlikely that any other arguments will be needed.

Our opinion here is clear and simple: the list of installed software and versions cannot be kept up-to-date without discovery. A manual software inventory will always have gaps. And the topic of licences – which shouldn’t be seen as being 1:1 with installed software – also remains far out of reach.

So what has to be included in the documentation in order to complete the process?

Server software

As a next step, installed server software is used to model the business-critical applications. If required, this can also be constructed by hand, since the information (except versions and patch levels) usually changes rarely.

Server services

These are used to construct infrastructure services and to identify dependencies. This information is also fairly static, since services are not created or modified on a daily basis. In principle, this information can also be put together manually. Although the automated creation of the list of services currently running can be highly informative – especially in terms of services that should not be running!

i-doit®

Step 6:
Key applications and IT services

www.i-doit.com

Step 6: Key applications and IT services

We've already highlighted it a number of times: in IT service continuity management, restoring critical applications is the highest priority to ensure business processes can restart as soon as possible. This isn't news, but it does require one thing: team effort. Apart from your admin, you'll need your networking experts and business department colleagues – sometimes even external service providers. It's at this point that you soon realise that there are lots of loose ends: why is something actually running, why does it need these resources? No-one really knows. If it ain't broke, don't fix it? OK, but make sure you document it!

If you're starting from scratch here, it's a good idea to utilise standard tools for application analysis. OBASHI can be helpful here – or service providers who specialise in the analysis of production systems. First of all, however, we actually need a list of the IT services or applications where we are going to start with our documentation work.

What's important?

Many IT units – and the actual size of the company is entirely irrelevant here – have a hundred or more applications running on their servers. Some are legacy services (that still haven't been finally withdrawn for whatever reason), some are brand new (in 'testing' which is now de facto semi-production), and this makes it hard to see what is 'just running' but which is actually keeping the company going. How can we get on top of things here and find out what is 'critical', so as to make a start on the documentation?

The red line

In every IT unit, there's an area involving one or more systems to which only the initiated are allowed access. A 'magic red line' is drawn around the servers and their associated storage and backup systems. A change here requires an act of parliament. You can be pretty sure of one thing: this is where your employer's money is made – or managed. And that is essentially one of the primary reasons for documenting things: being able to get the business processes that earn money back on their feet as soon as possible.

"Every component in these systems has multiple safeguards, everything's designed to be redundant, so you don't need to note every last detail." Don't be lulled into a false sense of security – you're just asking for multiple cases of Murphy's Law! Instead, see to it that the large sums of money that were available for setting up redundancies for these systems are matched by producing a set of documentation to the latest standards. And woe to those who don't have it to hand when the worst case happens.

The cloud doesn't need to be documented

Another popular misconception. Increasing numbers of systems are cloud-based, are available within minutes, but in the worst case, no-one can remember exactly how it was at the start… And even if you aren't given details of the 'black box' that runs your cloud applications, there are many pieces of information you can document:

- Contract (and end date!)
- Contact person for sales and support
- Support agreement
- Agreements of all kinds for backing up and restoring data, etc.
- Agreed maintenance window
- Login details for administrative accounts
- Service configuration (in general, as well as individual settings)

A key internal process has a major effect here: turnover in internal support staff must be matched by documentation and good practice. If the worse comes to the worst, it may take days to get 'your' data back: even without technical barriers, a provider who was not informed of your new contact person may stick to their policies and refuse to provide access.

The undervalued internal IT services

We could call them 'infrastructure services' or 'data centre services' – we mean the services that actually make it possible for the IT cluster to work in the first place. Examples of these include directory services, name resolution, firewalls, antivirus, database services, etc. This is another area where the list of services that are taken for granted quickly passes the 100 mark. But are they all documented so that they could be restored in a few minutes in an emergency? Can the directory that only yesterday had a new administrative account added for a service be restored to an identical state today? Can it even be restored at all, due to limitations imposed by the operating system? Is a working company directory perhaps needed to run the backup software?

Dependencies

We've now got to the heart of things and the difference between 'flat' IT documentation and the CMDB – the Configuration Management Database. The relationships: these bind objects into dependencies and thus into a hierarchy. The 'service tree' often referred to provides details of the elements that must (or may) be present so that the service works is fully functional to its capabilities.

The premium class: IT Service Management

So far, we've been working on documenting things that are (physical) objects, but our focus now switches to the services that are provided by the IT department. These are not only the technical services that run on computers. They can easily be services provided by the personnel in your IT department. Such as consulting work, for example, or the customer service provided by the department. And yes, even the documentation work that is described in this document and change management handling are processes that necessarily require all of the other IT services and therefore must also be documented sooner or later.

So your service catalogue contains a lot more than just technical items. With i-doit, you can also set up a service catalogue, link the corresponding objects and therefore map out the life cycle of these services.

Once you have completed all of these steps – including a foray into software documentation – congratulations are in order: you have identified all of the levels needed for documentation and can now go into detail in the areas where your company needs it.

So what are the next steps after this?

Automated creation of documentation

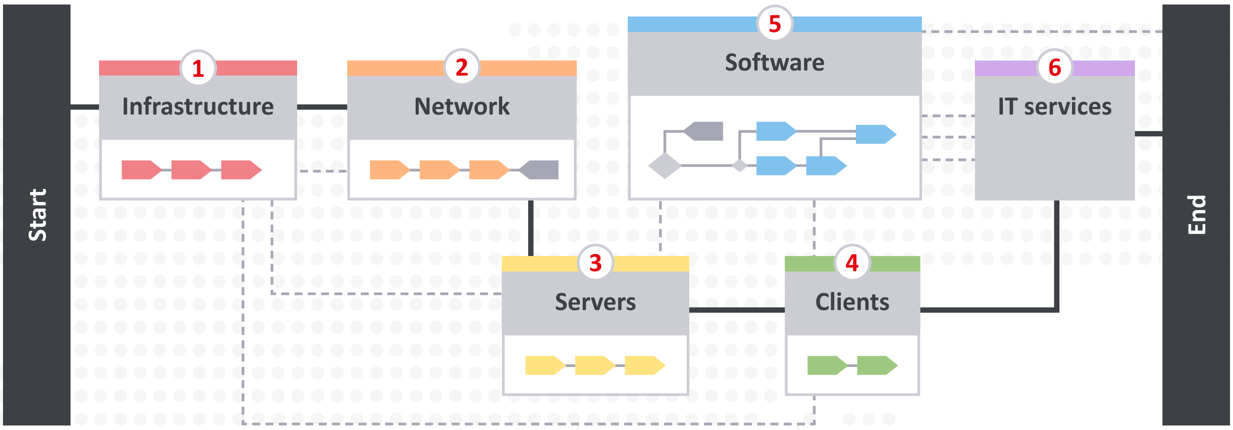
With the help of the i-doit document Add-on, you create up-to-date documentation automatically. Whether this is the Service Continuity Handbook, system notes or simply a daily export of mission-critical information onto a USB drive. Make your life easier, and make use of this information.

Automate what can be automated

The goal is not simply getting data into the CMDB, as is the case with discovery processes. Many use cases involve the control of other systems: the automated provisioning of servers, the automatic configuration of system and network monitoring, and much, much more.

Documenting what is still on the to-do list.

As one example, after successfully completing this project, it's now finally time to get on with that cable documentation job …



Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6

Contact the Team

Any questions? We will be there to get you started up: info@i-doit.com, +49.211.69931-0

Ingo Fries
Sales Manager

Daniel Kirsten
Product Manager

Peter Resch-Edermayr
CMDB-Evangelist

Further Resources

In order to gain an adept use of i-doit, we offer you various possibilities to get more involved in the topic: Would you like to test i-doit pro individually and in your own IT environment? Get the 30 days free trial. The on-premise trial includes all the features and add-ons of i-doit pro. The knowledge base should help you to gain a quick understanding of the software – both for new users starting out, as well as for professional use of the software. In addition, stay up to date through the blog about all the latest news.

Trial:
www.i-doit.com/en/trial-version

Knowledge Base:
https://kb.i-doit.com/display/en

Blog:
www.i-doit.com/en/i-doit-blog

We are i-doit. We do IT. For you.

i-doit is a registered trademark of synetics GmbH with headquarters in Düsseldorf, Germany. More than 30 employees serve over 5,000 users from all over the world from Germany and Austria. Founded in 1996, the company specializes in system security and was originally founded with a focus on planning, implementation and maintenance of adaptive infrastructure solutions. Since 2004, the company has been developing software for affordable and administrator-friendly IT service management.

www.i-doit.com

synetics GmbH
Hildebrandtstraße 4 d · 40215 Düsseldorf · Germany
info@i-doit.com · (fon) +49.211.69931-0

Imprint

Synetics GmbH was originally founded in 1996 by Markus Wolff and Joachim Winkler. During the first decade of its existence, the company has systematically focused on infrastructural issues as a system and consulting firm, gaining extensive experience in the operation and organization of IT environments. Since 2005, the company has been developing i-doit, one of today's leading software solutions for IT documentation, CMDB and security. Based on the knowledge gained from a broad range of project and consulting experience, a core product has evolved that has since been continuously developed in close cooperation with users and since 2013 has been the core business of this corporation.

What drives us

Information technology has formatively influenced and changed the working life over the last 20 years in almost all areas. In addition to the many innovations and facilitations that this development has brought and brings, the demands on those responsible for providing and managing their systems and applications are growing equally.

The resulting activities are today summarized under the term IT Service Management (ITSM). The demands placed on modern IT operations increasingly have no limits on the size of the company and thus demand more care, security and up-to-dateness from medium and small organizations.

To create offers that lead to successful solutions for the user at every budget is the essential motivation for synetics in the i-doit development and marketing.

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6

The author

Peter Resch-Edermayr is a CMDB evangelist at synetics, the Düsseldorf-based manufacturer of i-doit. During his career as an ITSM consultant, he got to know many "good and worst practices", but his sphere of influence was always limited to just a few customers. At synetics, he has been working on better practices since 2014, together with his colleagues and a large number of active users. The vision: to make IT documentation and CMDB more effective. In this way, i-doit is rapidly becoming the standard solution for IT documentation and CMDB, and our clients' projects are becoming ever more sustainable.