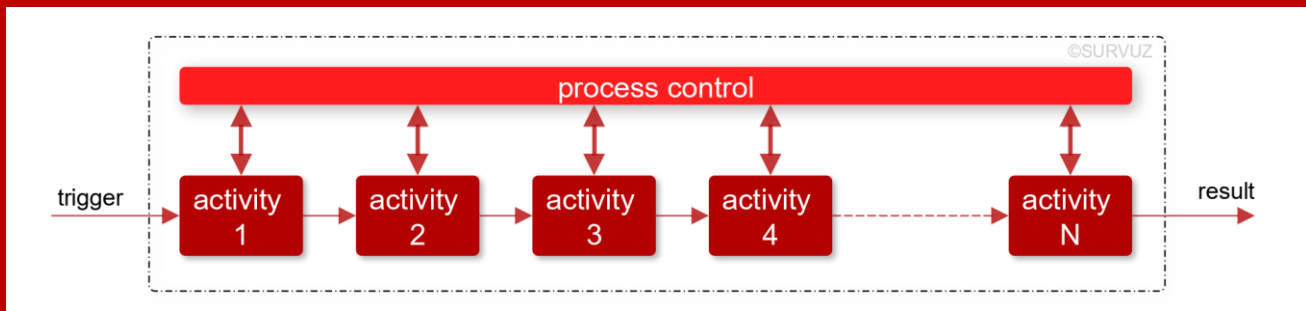


# Demystifying the term PROCESS



## COLOFON

Title: Demystifying the term PROCESS

Author: Jan van Bon

Publisher: SURVUZ Foundation

Edition: First edition, May 2022

© 2020, 2022 SURVUZ

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by the publisher.

Although this publication has been composed with much care, neither author, nor editor, nor publisher can accept any liability for damage caused by possible errors and/or incompleteness in this publication.

## FOREWORD

The terms service, management, and process, are among the most (ab)used terms in economy. In this e-book I'll demystify the term *process*.

All over the world, in any book on process management, you'll find the same definition of *process*. Yet, in all of those books, this definition is only used on page one. The rest of the book violates this definition on each page, discussing only the derived artefacts of processes that are not processes any more. And if these artefacts would *indeed* have been derived *from* processes, the harm would have been manageable. Unfortunately, this is not the case. As a consequence, all economies, all over the world, pay the price of the resulting inefficiency.

And yet – it is *unbelievably* simple to solve this. The only thing it requires is a bit of logic, and the consistent use of definitions. Any high school kid could do it.

---

*This makes it so much more surprising to see smart people, with the required brain mass, use the term process in such an inconsistent way – with such devastating consequences.*

---

**CONTENT**

1 DEMYSTIFYING THE TERM 'PROCESS' ..... 4

2 THE WHAT ..... 6

3 A VERB ..... 9

4 COUNTABLE ..... 10

5 NO GATEWAYS..... 12

6 THE PURPOSE ..... 14

7 THE PIECES..... 17

8 THE PROCESS MODEL ..... 19

9 HOLISTIC ..... 21

10 NON-REDUNDANT ..... 24

11 CONTROL ..... 27

12 CLOSING REMARKS..... 30

# 1 DEMYSTIFYING THE TERM 'PROCESS'



Figure 1. We're living in times of chaos

This little e-book will let you in on some aspects of the *new thinking* about service management that will help you get *in control* of your services in a very simple and sustainable way.

## 1.1 'Process' is one of the *most* abused terms

If you open the average book on process management, you'll always find a similar definition of 'process' on page 1. It will be something like:

---

**Process:** *a series of activities that is triggered by an input and that delivers a predestined output, which is assured by means of control.*

---

The definition may even go so far as to talk about *outcome* instead of *output*. Unfortunately, that average process management book then goes on to provide only information about procedures, work instructions, and other practices – often organized in a practical reference model of business functions. The consequence is that “everything is a process”, and we can no longer see the forest for the trees.

This is the bitter reality for the vast majority of organizations in today's economy and society, in business and in government, in health care and in IT.

## 1.2 A sustainable solution

Fortunately, you can escape from this chaos. The only thing you have to do is to apply a very traditional, almost *old-fashioned* skill:

---

*Think First, Act Later*

---

That approach seems almost lost on many organizations and experts that are focusing on the latest technologies and techniques. Nevertheless, a sustainable future requires a sustainable approach, and that's why that old-fashioned skill may come in handy from time to time.

## 1.3 10 fundamental requirements for processes

Most of what we've learned to call 'process' actually isn't a process *at all*. That's where we got off-track. So let's get back on that highway and see if we can get to that much-desired dot on the horizon: a simple, sustainable, understandable way of organizing our work in a process-driven way. After all, *processes* are supposed to deliver the shortest path between input and output.

The Unified Service Management method (USM) specifies 10 requirements that something *must* meet, before we can use the term 'process':

1. **A *process* describes what has to happen successively, not the who or how.**
2. **A *process* can be interpreted with a verb.**
3. **A *process* can be counted.**
4. ***Processes* are not depending upon practical conditions (∅).**
5. ***Processes* have a customer-relevant and unique purpose.**
6. **A *process* can be divided into sub-processes, but that does not change the process.**
7. **A *process* model organizes the processes.**
8. **An integral *process* model includes all service management activities.**
9. **In an integrated *process* model, each activity occurs only once.**
10. **All activities are steered using *process* control.**

The following chapters take you through each of these requirements – and why they are critical for a sustainable management system for your service organization.

## 2 THE WHAT

This **first requirement** seems obvious and logical, but in a mature service management context, it is the *most abused* requirement of all: “A process describes what has to happen successively, not the who or how”.

And it's easy to see that this is so. Just look at any of the popular best practice frameworks, including ITIL, COBIT, IT4IT, FITSM, ASL, BSL, etc., and you'll find hundreds of things that are called 'processes':

- ranging from Security Management, Capacity Management, and Financial Management, to Demand Management, Design Coordination and Customer Relationship Management in the discipline of IT Service Management (using ITIL)
- ranging from Application Portfolio Management, Supplier Definition, and Development Strategy, to Software Control and Distribution or Testing, in the discipline of Application Management (using ASL)
- ranging from Information Lifecycle Management, Specify Information Requirements, and Business Data Management, up to Information Coordination or Design Non-Automated Information Systems, for the discipline of Business Information Management (using BiSL/DID)
- ranging from Manage Strategy, Manage Organizational Change Enablement and Manage Human Resources, up to Manage Enterprise Architecture, Manage Configuration and Manage Relationships, in the discipline of IT auditing (using COBIT)
- ranging from a myriad of practicalities that are described in the hundreds of reference frameworks you can find at APQC's website, to any of the 'business processes' that TOGAF talks about

If you add all of these 'processes' together, you'll find many hundreds – maybe even thousands. But if you would test these 'processes' against the first requirement [1] of the 10 presented above, you will immediately see that *they all fail*.

They are *not* limited to just activities: the **what**. They *also* describe the **who** and the **how**, in applied ways. That's because they have their origins in the best *practices*. And practices are always combinations of the what, the who, and the how – applied to some issue of common concern, such as the important question of “How do I make sure that my services are secure?”. That is an important question – no doubt about that – but Security Management is not a process: it's a *practice*. An important one, but still – a *practice*.

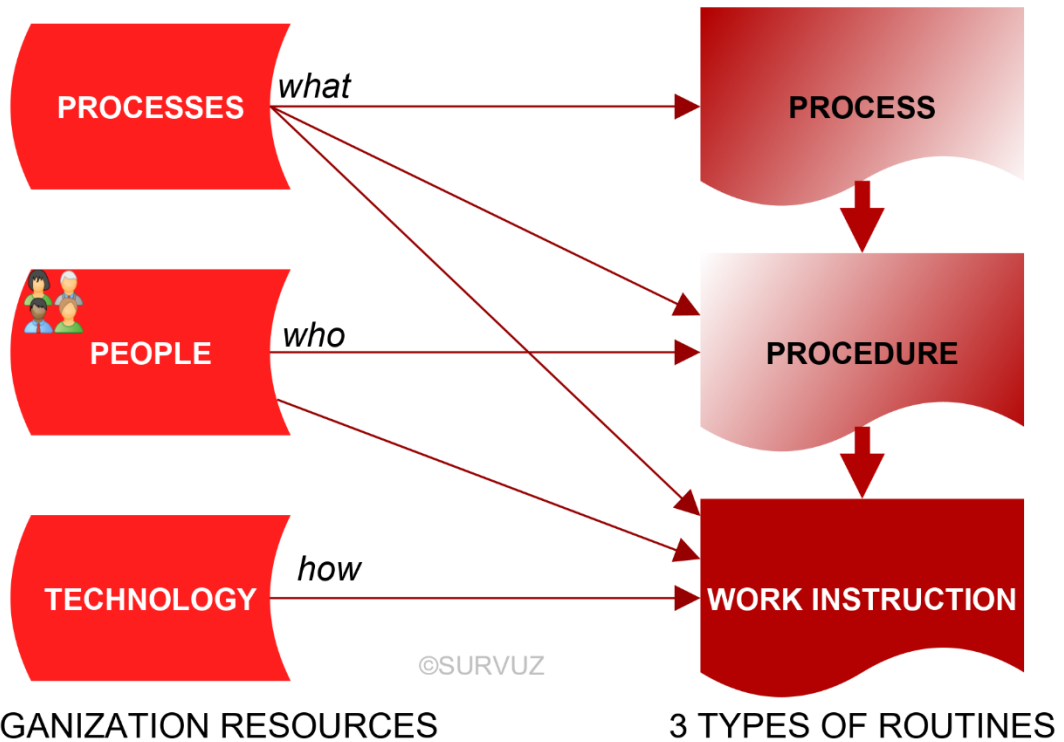


Figure 2. A process only describes the **what**. As soon as the **who** or the **how** is added, we have a procedure or a work instruction: a practice

Figure 3 shows the sum of all 'processes' of ITIL4, ASL, BiSL, and COBIT: far more than 100 'processes'. Just imagine what you'll get when you add the 'processes' of all the other frameworks and standards to it...



Figure 3. Calling everything a process is at best a matter of 'misunderstanding'.



## 2.1 Do the litmus test

Check the things you've been calling 'processes' against requirement [1]. Look at the practical level of detail of the guidance you find. Conclude that this guidance contains details on the **who** and the **how**. And then prepare to foster your new thinking on a sustainable *process* architecture for Enterprise Service Management, using the simple logic of the USM Method.

### 3 A VERB

**Requirement 2** of the 10 requirements says: “A process can be interpreted with a verb”.

Requirement 1 was simple: a process only describes the ‘what’, not the ‘who’ or the ‘how’, otherwise it would have been a procedure or a work instruction, respectively.

When you only describe the ‘what’, you only describe the things you *do*, i.e., the activities.

#### 3.1 “A process can be interpreted with a verb”

---

*“Doing an activity” requires a verb*

---

It’s that simple. That verb can then be applied to *anything*, in terms of *applying* the verb. But it still is only a verb.

But there’s another side to this coin, a very meaningful one: if it is *only* about the what, and not about the how, then there *cannot* be a noun involved in the title of the process... After all, *if* it would involve a noun, the thing would include a specific application of the verb to a practical *object* – the noun. And that would immediately turn the thing into a *work instruction*.

This *object*, the noun, can be anything, ranging from a tangible *good* to a more or less intangible *quality*. As soon as the verb is extended to include that specific tangible or intangible noun, it is no longer a process but it has turned into a practice.

## 4 COUNTABLE

**Requirement 3** of the 10 requirements says: “A process can be counted”.

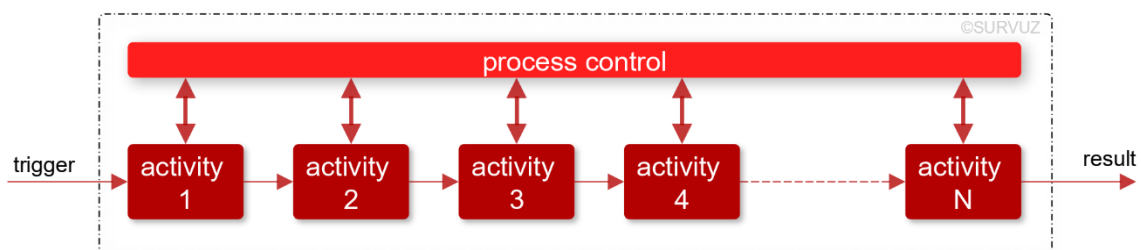


Figure 4. A process has a trigger, a series of activities, and a predefined output (outcome)

This third requirement is also very simple. A process starts with a *trigger*. This trigger can be an activity (someone asks for something), a situation (an event, e.g., something passed a threshold), an output of another process, an alert or signal, or any other conceivable thing that is capable of triggering the sequence of activities of the process. These triggers can occur any *number* of times, and as a consequence, the process occurs the very same *number* or times – even if that process is under a release strategy in terms of bundling its course with one or more other instances of that process. Therefore, the number of times the process occurs can be **counted**.

### 4.1 The litmus test

Now let's hold requirements 2 (verb) and 3 (counting) against the things that have been called 'process' by some many people for such a long time, following up on the best practice guidance of popular frameworks. There's millions of IT experts that call **Security Management** a process. This means I can ask the question “How many *securities* have you done today?”. And I have indeed asked this question many times, from the stage of the conferences where I was invited to speak in the last two decades. And the audience would gaze at me, thinking “What is that crazy Dutchman asking now? That is a silly question!”. Indeed – it was a silly question. Sillier than they perceived – because it couldn't be *answered*. For the simple reason that the thing couldn't be *counted* – and therefore, the thing *wasn't* a process.

I scored the same effect when I asked the audience “How many capacities have you done today?” Or “How many service levels have you done today?”. Or “How many personnel and talents have you done today?”. Or “How many knowledges have you done today?”. Or even “How many Service Desks have you done today?”

All of these considerations contribute to the conclusion that the things that were accepted by that audience as 'processes' (in the examples: Security Management, Capacity Management, Service Level Management, Personnel and Talent Management, Knowledge Management, and even Service Desk), *were actually not processes at all*. They all contained a *noun* that indicated an object (a good or a quality aspect): security, capacity, service level, personnel, talent, knowledge, service desk. And they all are not processes, but *practices*.

---

*Maybe they are **best practices** – but still: practices.*

---

## 5 NO GATEWAYS

**Requirement 4** of the 10 requirements says: *Processes are not depending upon practical conditions (◇).*

Requirements 1,2, and 3 have laid the foundation for the difference between a **process** (the *what*), a **procedure** (the *what* and the *who*), and a **work instruction** (the *what*, the *who*, and the *how*). See Figure 2. **Practices** are generic descriptions of work, done at the level of a work instruction.

### 5.1 Practical conditions

It is evident that any **practical condition** in a description of work can only be positioned at the level of the work instruction, i.e. the level of these **practices** - and never at the level of the process, for the simple reason that the *how* is not present in that process level. *Quod erat demonstrandum.*

This means that, whenever you see a 'process description' that contains a gateway symbol (◇) that tests a practical condition, you should immediately understand that you are not dealing with a *process* description, but with a *practice* description. On itself, this wouldn't be that bad, but unfortunately, it has far-reaching consequences. The rationale of requirement 4 is very easy to understand, but a simple example will demonstrate the devastating effects of *not* using it.

### 5.2 Far-reaching consequences

The large majority of 'business process analysts' use a drawing technique like BPMN to define 'business processes'. BPMN is the acronym for Business **Process** Model and Notation. BPMN then uses three 'swimlanes' to describe that 'process'. These swimlanes are used to describe work in terms of *who* does *what* and *how*. It may be evident that any result of a swimlane description of work is positioned at the level of the *work instruction*, the *practice*. Again: check Figure 2.

---

*There is no such thing as 'a business process'. As soon as a process is applied to a specific business, it has turned into a practice.*

---

Therefore, the acronym BPMN should not mean Business **Process** Model and Notation, but Business **Practice** Model and Notation.

---

*Business process analysts are not business **process** analysts: they are business **practice** analysts.*

---

And when these so-called business *process* analysts start to specify and document business *practices*, they start at the wrong end of the stick: they clearly suffer from a sincere pandemic: [PTO – Polishing The Outside \(the new Pandemic\)](#). And *that* is where it hurts. At best, their efforts will deliver some improvements of the Lean type: less loss. But they fundamentally miss the sustainable solution that they *could* have found if they had started at the other end of the stick.

### 5.3 The litmus test

Ask any so-called business *process* analyst which **process architecture** they used to *derive* the practices in their BPMN swimlanes (again, according to Figure 2). All you'll get is silence. There will be *no* answer. For the simple reason that *they didn't use* a process architecture. They have simply *started* their thinking at the level of the practices. Most likely, they are even completely *unaware* of the whole concept of process architecture, for the simple reason that they have learned to use the term *process* for what actually is a *practice*. And then you're blind for the consequences. You'll never find a sustainable solution for the organization of work until you start to think from the other end of the stick - from the *beginning*.

---

*It sounds crazy, but the majority of experts has become used to **start at the end instead of at the beginning.***

---

## 6 THE PURPOSE

**Requirement 5** of the 10 requirements says: “Processes have a customer-relevant and unique purpose.”

*In the first chapters, we already demonstrated that many so-called processes are actual **practices**. But even if you do not add People or Technology to the Process dimension, **many so-called processes do not qualify as customer-relevant.***

The service management context of USM is the **customer- and business-driven level of maturity**, as indicated with the green arrow in Figure 5.

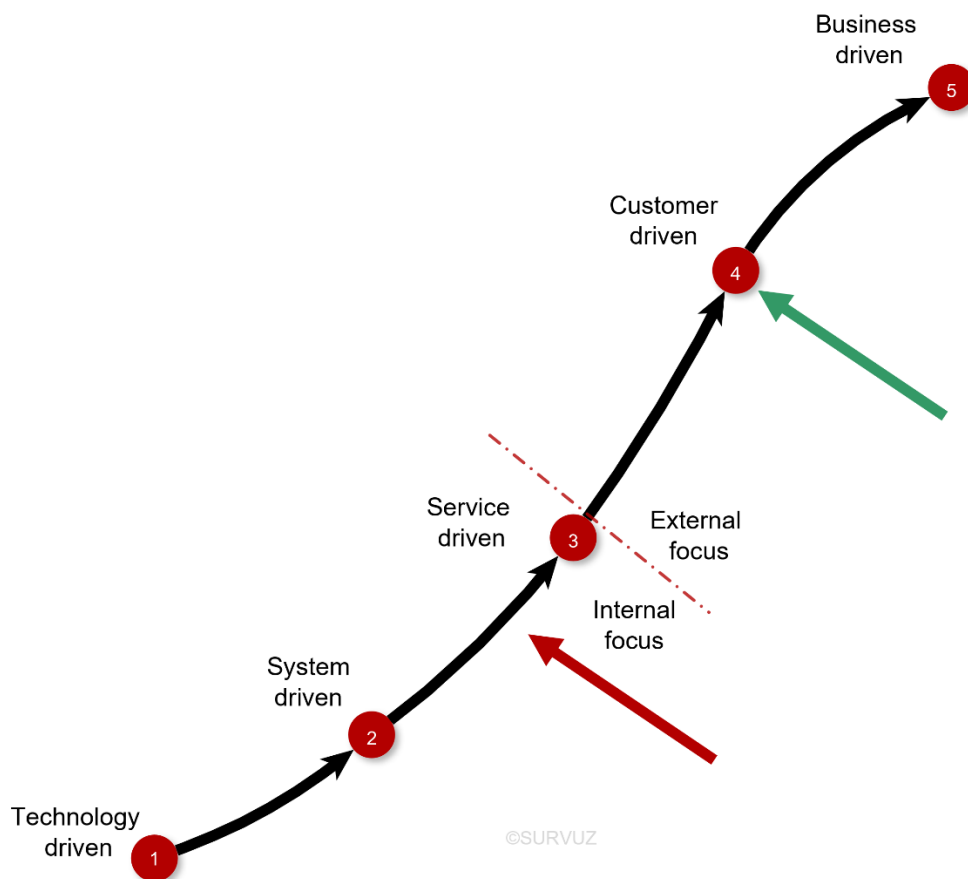


Figure 5. The USM Value Maturity Model

Value creation is the ultimate goal of any service provider, and therefore *maturity* should be measured in terms of the *value* that the provider creates for the customer and not in terms of the traditional *capabilities*. Although providers talk about value creation all the time, most of them are still struggling somewhere between level 2 and 3, and many don't even reach that

level – just like their customers. This is mainly due to the dominating technology focus of these providers and the fact that for the last 3 decades, they have mostly followed a practice-driven approach, which has not provided a sustainable service delivery strategy.

---

*“If you do what you did, you get what you got” [Einstein]*

---

As Einstein said, they will have to change that approach, or else his famous statement will keep these providers stuck at the position of the red arrow in Figure 5. And maybe they can make a lot of *money* there, but they will not create a lot of *value*.

### **6.1 A customer-relevant process doesn't deliver intermediate results**

The providers at levels 1, 2 or 3 tend to call *anything* a process, as long as it's a sequence of activities – whatever the **output** is. However, if the output of that process is not (yet) relevant to the customer, it doesn't qualify *as a process* for the customer, because its output is at most a semi-finished, intermediate result that doesn't influence the customer's business (yet).

---

*From the perspective of the customer (levels 4 and 5), only a process that delivers an actual customer-relevant result is a process, the rest is at best an intermediate step of a process.*

---

For a customer who seeks *value* from a provider, this is not enough. Moreover, this has been the main reason for the Business-IT Gap of the last decades: providers with an immature, mostly technology-focused proposition for customers that seek business improvements.

### **6.2 Example**

Dell is a well-known provider of desktops, laptops, and other computing equipment. They may be scored at a level 5 of the traditional Capability Maturity Model, and they can be expected to do very well in terms of Operational Excellence. But in the USM Value Maturity Model, Dell will score a level 1, 2, or at most a level 3, because Dell has no idea to what end the customer is using this equipment in their business. Dell may be a very profitable company, but they do not act on level 4 or 5.



### 6.3 Intermediate results fail to qualify as processes

A provider that calls 'Software Development' a process doesn't relate to the customer's business, as software is merely a piece of technology (level 1), not even a system (level 2).

'IT Asset Management' is not a customer-relevant process, because this is an internal activity of the provider and the output is only relevant to that provider. Besides, IT Asset management is mostly described *as a practice*.

Supplier Management, Knowledge Management, Service Design, Service Validation & Testing, Deployment Management, Strategy Management and many more processes are not *customer-relevant* processes, because they are internal activities of the provider and their output is only relevant *to that provider*.

---

*Providers that offer IAAS, PAAS, HAAS, or cloud-based SaaS are merely acting at the system level (2) or at most at the service level (3).*

---

### 6.4 Customer-relevant processes

Processes that *are* customer-relevant will need to have a customer-relevant *input* (a triggering customer request) as well as a customer-relevant *output* (the response to that request). Requirement 5 of the 10 USM requirements will highly limit the number of processes that can be called processes *at the customer-driven or business-driven levels* (4 and 5) of the Value Maturity Model.

## 7 THE PIECES

**Requirement 6** of the 10 requirements says: “A process can be divided into sub-processes, but that does not change the process.”

In the previous chapters we’ve learned that a **customer-driven strategy** can only recognize processes that have a customer-relevant output. Any intermediate result is only of interest to the *provider*, not to the customer. For the customer, only the end-result counts.

### 7.1 Software development is not a customer-relevant process

The consequence of the 6th requirement is that most of the things experts focus on (especially in IT), are not relevant to the customer. E.g., software development is an internal activity. The customer couldn’t care less *how* the functionality is generated, as long as it *is*. Software on itself is an empty shell. It can only be of value if it cooperates with hardware, network, middleware, housing, hosting, or any other component that is required to create a fully operational *facility* for a customer. And all of that can only be of *real value* to the customer, if its functionality is guaranteed by the required *support* when that facility is used in practice in the customer’s business.

---

*A service is a supported facility*

---

### 7.2 Testing is not a customer-relevant process

The same applies to many other things we’ve learned to call ‘processes’ for the last couple of decades, including testing, monitoring & event management, service design, knowledge management, etc. All of these only deliver a direct output for the provider, and at best an indirect output for the customer. Therefore, at best, they can only be considered to be activities, steps, or *provider-centric* sub-processes of any customer-relevant process.

### 7.3 Fragmentation rules the show

The fact that – especially in IT – so many providers focus on only very small portions of the customer-relevant facilities and processes, pushes providers to lower levels of the USM Value Maturity Model. We outsource as much as we can, and the number of outsourcing partners is growing exponentially, creating a **fragmented society** and a myriad of [Devil’s triangles](#).

As a consequence, the relationships between components and parties also grow exponentially, creating a **connected society**.

And because of all of this, the dependency between parties leads to a **dependent society**.

Whatever approach we follow to make sure the whole thing with all its growing complexity keeps doing what we need it to do, we at least need a *control* approach at an end-to-end level: an Enterprise Service Management strategy that focuses on **complexity reduction**. That is where [the USM method](#) kicks in: it provides the concept of the universal link that can be used to build endless supply chains and service ecosystems, based on a simple [service management architecture](#). With only customer-relevant processes.

## 8 THE PROCESS MODEL

**Requirement 7** of the 10 requirements says: “A process model organizes the processes.”

---

*Enterprise Service Management requires one captain on the ship*

---

In the previous chapters we've learned that **practices** are not processes, and that a customer focus requires that processes should be defined from a customer-relevant perspective. Now let's presume we have determined what these processes are, so we can take the next step and put these processes together, in *one* model. The 7<sup>th</sup> requirement for *one* process model is easy to understand: an efficient management system must be *singular* – by definition. As the management system is based on the process model, we simply can't have two or more of either of these.

---

*There must be one captain on the ship, one steering wheel in the car*

---

### 8.1 Frogs in a bucket

Many organizations suffer from having *more* than one management system. This is often caused by the island perspectives that the involved teams take: these teams reside on a lower level of maturity in terms of the [USM Value Maturity Model](#), as they perceive their contribution to the enterprise from an isolated perspective, an *island*. They then manage that contribution in a rather independent way, as if they were king on their own island. They act like frogs in a bucket – jumping in any direction they think is best, from *their* perspective – not from the singular *enterprise* perspective.

It is relatively easy to determine their maturity position, as their service agreements (if they have any at all) are stuffed with internal KPIs, emphasizing their internal focus. This is what we see in DevOps: a tilted silo with lots of new IT shops that focus on change, each using similar techniques, but often failing to act as a coherent set.

### 8.2 Collaboration requires integration

If the involved teams want to contribute to the enterprise's achievement, they will need to cooperate in an effective and efficient way.

---

*In a complex environment, interoperability is key*

---

And this is exactly where so many organizations fail: they allow their teams to act as islands, each having their own, local interpretation of a management system. But effective and efficient collaboration requires **normalization** of the way the teams contribute to the enterprise goal. And normalization requires some level of **standardization**, especially on the aspect of each team's management system.



Figure 6. USM helps create sustainable supply chains with normalized links

The links of the supply chain simply need to fit together. Not just *within* organizations, but also *between* organizations. Figure 6 shows exactly what would happen if you do not normalize the management system of the actors in the supply chain: both images specify a **supply chain**, but if you would be hanging from the edge of a cliff by one of these chains, most of us would prefer the lower one...

### 8.3 The USM concept of the link

The links in these supply chains are the *management systems* of the contributing teams (the 'actors'). It may be obvious that the upper image of Figure 6 is composed of many types of management systems, and the lower image has a *normalized* management system for each team.

Unfortunately, this is easier said than done. Creating supply chains from normalized links has an inherent danger: the links must be **strong** (you wouldn't want to trust your life to a chain made of paper) and the links must limit themselves to your interoperability function: the **outside** of your team domain (nobody wants to be told what *internal* decisions they should take).

This concept of a singular *normalized management system* as an *acceptable link* is the core concept of the [Unified Service Management method](#), and it is based on the concept of an integral and integrated *process* architecture.

## 9 HOLISTIC

**Requirement 8** of the 10 requirements says: “*An integral process model includes all service management activities.*”

In the previous chapters we’ve learned that **practices** are not processes, and that a customer focus requires that processes should be defined from a **customer-relevant perspective**. Now let’s compare your organization with a car, or a ship.

---

*How many steering wheels would you prefer for your organization? How many captains on your ship?*

---

Of course, there should be only ONE captain on the ship, one steering wheel in your car, as chapter 8 stated. This comparison makes very clear that each organization requires a strategic choice for the leading principles, the leading ways to organize the work, the leading ways to deliver value, at least if you prefer an *effective* organization. After all, each inconsistency in your management system leads to an *ineffective* organization – and not only that: inconsistencies can also be destructive for the quality goals of the organization. And exactly *that* is what we see *everywhere*: fundamental inconsistencies in the way organizations organize their performance.

---

*The number one fallacy? Project management.*

---

### 9.1 Projects follow processes

Whenever I see an organizational chart including a Project Management Team, or worse – a Project Management Office, a Project Portfolio Team or even a Program Team, I know where to look for the cause of underperforming organizations. Not only is the success rate of projects horrifying low, the thing is, that many project *managers* have a goal that is *conflicting* with the interest of the organization – even though they *think* they contribute to that.

These project managers got an assignment from a C-level manager, they got a budget, and they got the authority to take staff out of their regular work to contribute to the project. And last but not least – they think they can determine *their own way of organizing the work*, using one of the many project management techniques. They are the champions of the organization! They make it happen! And as long as popular frameworks like ITIL or COBIT say that Project Management *is a process*, or even that Program Management *is a*

*process*, they will continue to *undermine* the strategic opportunities for the organization to ever become an integrated, holistically working, value-delivering organization.

---

*Project management practices undermine the strategic goals of the organization*

---

This way of doing Project Management is in direct conflict with rules 1, 2, 4, 5, and 7 of the 10 rules we started with, and it is one of the most fundamental errors you can make in setting up a service organization.

As with DevOps, Project Management focuses on *change* – within a very limited scope. It has lost contact with the *continuous* value creation promise of the modern service provider, and it thinks that when the project result is delivered, the job is done – while everyone with an ounce of brain matter would understand that it then only *begins*.

---

*Project management is often the second captain on the ship*

---

Therefore, projects should always *follow the processes* of the organization. And these processes should comply with all 10 of the rules of chapter 1, to ever enable the organization to be an efficient creator of continuous value, becoming a successful organization that is fit for a customer-driven future.

When project managers would learn to understand that, they would immediately be more successful, as they would *use* the organization's singular management system, and they would *embed* their results in this holistic system, assuring the value creation that *must* have been the goal of their project.

---

*Unfortunately, most project managers think they live on an island*

---

This of course doesn't only apply to projects, it applies to *all activities* of the organization: they should all be part of **one holistic management system**, for the simple reason that the organization needs *one* captain on the ship, *one* steering wheel in the car, *one* strategy for continuously delivering value to their customers.

Unfortunately, many teams and disciplines in the average organization see the world from their own, isolated perspective, **as islands in the organization's see**, focusing on their own span of control instead of collaborating as components in a holistic system.

---

*We should learn to see our organization through the eyes of one integrated management system*

---

If you want to learn how to achieve this, you should start with the *foundation* of your organization. And that will most likely require that you UNLEARN a lot of what you have learned in the past – at least if you want your organization to ever achieve level 4 of the USM Value Maturity Model.



## 10 NON-REDUNDANT

**Requirement 9** of the 10 requirements says: “In an integrated process model, each activity occurs only once.”

In the previous chapters we’ve learned that **practices** are not processes, that a customer-focus requires that processes should be defined holistically from a **customer-relevant perspective** for maximum effectiveness, and that a process model should **include all relevant activities** to prevent ending up with two or more captains on the ship. Two more requirements to go!

### 10.1 Why Best Practices’ redundancy kills your efficiency

The 9<sup>th</sup> requirement deals with the *efficiency* of the process model. This requirement finds its origin in the remarkable observation that (service) organizations have an incredible amount of *redundancy* in their daily routines. Now, *everyone* knows that redundancy causes *inefficiency* – and not only that: redundancies can be destructive for the quality goals of the organization. Yet, all organizations that ever allowed me to look in their kitchen, had this huge inefficiency embedded in everything they do on a daily base! And the line of defense was always the same: *this is best practice!*

### 10.2 Non-redundancy for maximum efficiency

Now suppose you want to create maximum *efficiency* in an approach that was already designed for maximum *effectiveness*. Would you then really want to start with adopting best practices that are the result of a *presumed* strategy – hoping that they would *also* fit your own organization, your routines, your tooling, without knowing how and why these practices were created in the first place? Or would you consider starting at the other end of the spectrum, at the *beginning* – not at the *end*...?

---

*Starting at the beginning and not at the end seems to be the most logical approach – but it is rarely followed*

---

Starting from principles and building blocks, in a well-designed Service Management Architecture, might just deliver the consistent results you had *hoped* to find in the examples of others (the *best practices*). Taking that architectural perspective would most likely allow you to also benefit from all these practices in a way you had never suspected.

### 10.3 Start with the essential question: what do you *do*?

So, how do you **design** for maximum efficiency through maximum non-redundancy? First of all, this should not start in your local *organizational* structure ('people first'), or in your local selection of *tools*: these two dimensions are always the result of ever-changing conditions, opinions, history, legal constraints, cultural considerations and technological requirements. You should start with *the things you do* to realize your organization's mission and goals – and that is what you find in the *process* dimension. Note that processes are not *practices*, and that all the before-mentioned requirements for *processes* apply here.

### 10.4 The root of non-redundancy lies with the activities

We can now focus on the stuff processes are made of: activities. We want to make sure that we build a process architecture with maximum non-redundancy for maximum efficiency. This means that – according to the concept of *building blocks* - for every activity that is used more than once, there is only *one specimen* in the process architecture.

Let's take Change Management as a process. If Risk Management then requires a change to deploy some improvement, it will simply invoke Change Management and let that process do what it always does: change something. The same would apply to practices and functions like Security Management, Continuity Management, Service Desk, Financial Management, Portfolio Management, or any of the other popular practices and functions: they would all re-use Change Management for the purpose of *changing* something in a controlled manner.

And if you execute an operational activity in your live environment, it doesn't matter whether that was *triggered* by a change or an incident – as long as you do what is required in a responsible way. At level 4 of the Value maturity Model, this should *always* be determined by the (agreed) interest of the customer. Therefore, Operations Management should be triggered by Change Management *and* by Incident Management, as well as by any other component of the process model that requires an operations activity in the live environment. One of the biggest inefficiencies can be found when organizations apply multiple 'deployment' activities in the very same – one and only - *live* environment. Get rid of them! There is only *one* Operations Management process.

### 10.5 The ultimate re-use of activities

All of this comes down to the ultimate re-use of *activities* that are defined in constructions that we handle as *processes* – according to the earlier requirements in the list of chapter 1. Once you get the hang of that, the only thing you need to be watching is that there is a limit to the *uniqueness* of these activities. Some are similar, others are *very* similar, and others are *identical*.

The requirements in the list will help you find the ultimate process model; *true* processes instead of practices, with customer-relevant purposes, in an integral and integrated process model, and non-redundant for maximum efficiency.

You can try and find out for yourself what the resulting process architecture is, but you may also read up on what the USM Method has specified. It will save you 25 years of hard work – and the result (Figure 7) is easily understood if you accept the requirements of chapter 1.

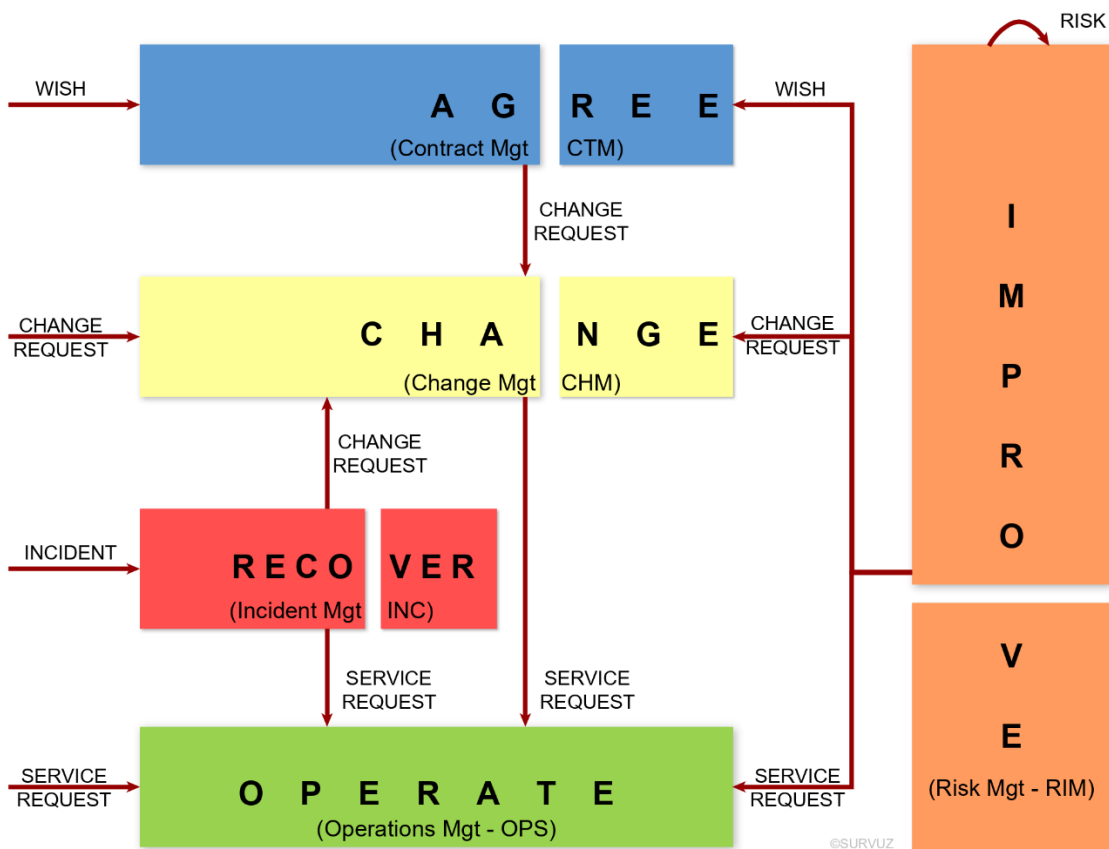


Figure 7. The USM Process Model

This integrated process model then quickly demonstrates which ‘path’ a customer interaction can follow through the integrated process model, which results in the **8 unique workflows of the USM Process Model**, covering **all activities** you’ll ever need do for your service management.



contribute to, together with the other two organizational resources: People and Technology.

### 11.1 Process control through steering

Processes can be steered from two perspectives:

- the **process manager** perspective – focused on the *logic* of the process
- the **coordinator** perspective – focused on the *execution* of the process

In the USM method, the process manager only steers on the logic of the process that is laid down in the agreements about the way the process is executed. These agreements are made with the (internal and external) parties that operate the process: coordinators and operators. This means that the process *manager* doesn't *coordinate* the actual process: the manager only directs (steers) the way this is done.

The coordination can then be organized from – again – two different perspectives (see Figure 8):

- the **logic** of the process: this concerns the profile of the *process coordinator*
- the **power of the hierarchy**: this concerns the profile of the *team coordinator*

The outcome of this steering is the *adjustment of the process execution*.

### 11.2 Process control through informing

For improvement *and* accountability purposes, we need data on the execution of the actual calls in the process. This can again be approached from two perspectives: the *process manager* and the *coordinator*.

The process manager will again agree with the coordinators *what data* will be gathered, and the coordinator will then gather the data and make it available for the process manager, as well as for internal purposes.

Process data serves two purposes:

- **improvement** of the way the process is set up and executed
- **accountability** for the way the calls in the process have been executed

Improvement is the ultimate responsibility of the process manager. The data is used to analyze the process, look for opportunities to further improve it, and agree with the coordinators on those improvements. The process manager can

use the USM process Risk Management to handle all improvement initiatives, using [the agile USM deployment model](#).

Accountability is expressed in terms of reporting, through two channels:

- **service reporting** to the customer (covering the reporting on the delivered facilities and the delivered support in the form of the 8 USM workflows)
- and **process reporting** to the internal management

The outcome of this informing is the *improvement of the service delivery* (effectiveness as well as efficiency) and of customer satisfaction.

## 12 CLOSING REMARKS

This concludes the rant on the topic of "Demystifying the term *process*". Applying these 10 requirements will inevitably lead you to the USM Process Model, its 5 processes and 8 workflows (Figure 7).

Hopefully, this has demonstrated the power of the methodical approach specified in USM, the Unified Service Management method. If you spend some effort on this, you'll soon find out how USM enables you in a more effective and - most of all - more *efficient* way of applying the popular guidance from best practice frameworks such as ITIL, FITSM, or COBIT, and to find a much simpler, cheaper, and more sustainable route towards any certification against popular standards like ISO27001 or ISO9001.

Tons of free information are available at the [USM Portal](#). Compliments of the [SURVUZ Foundation](#).

The new thinking of USM - the Unified Service Management method - can help you **unlearn**. Read [the USM book](#) or any of the [free downloadable e-books](#), or take a [USM Foundation course](#) or a [free online USM workshop](#) - whatever you prefer, and *learn* what you haven't learned at school.

If you're a professional, supporting others to improve their business, you can also become a [USM Professional](#): a coach, a trainer, a product expert, or an auditor. The SURVUZ Foundation supports you with lots of free instruments to be used in your assignments – but only after you've demonstrated that you can use these instruments and explain USM to others..



©2022. This free e-book is a publication of the SURVUZ Foundation