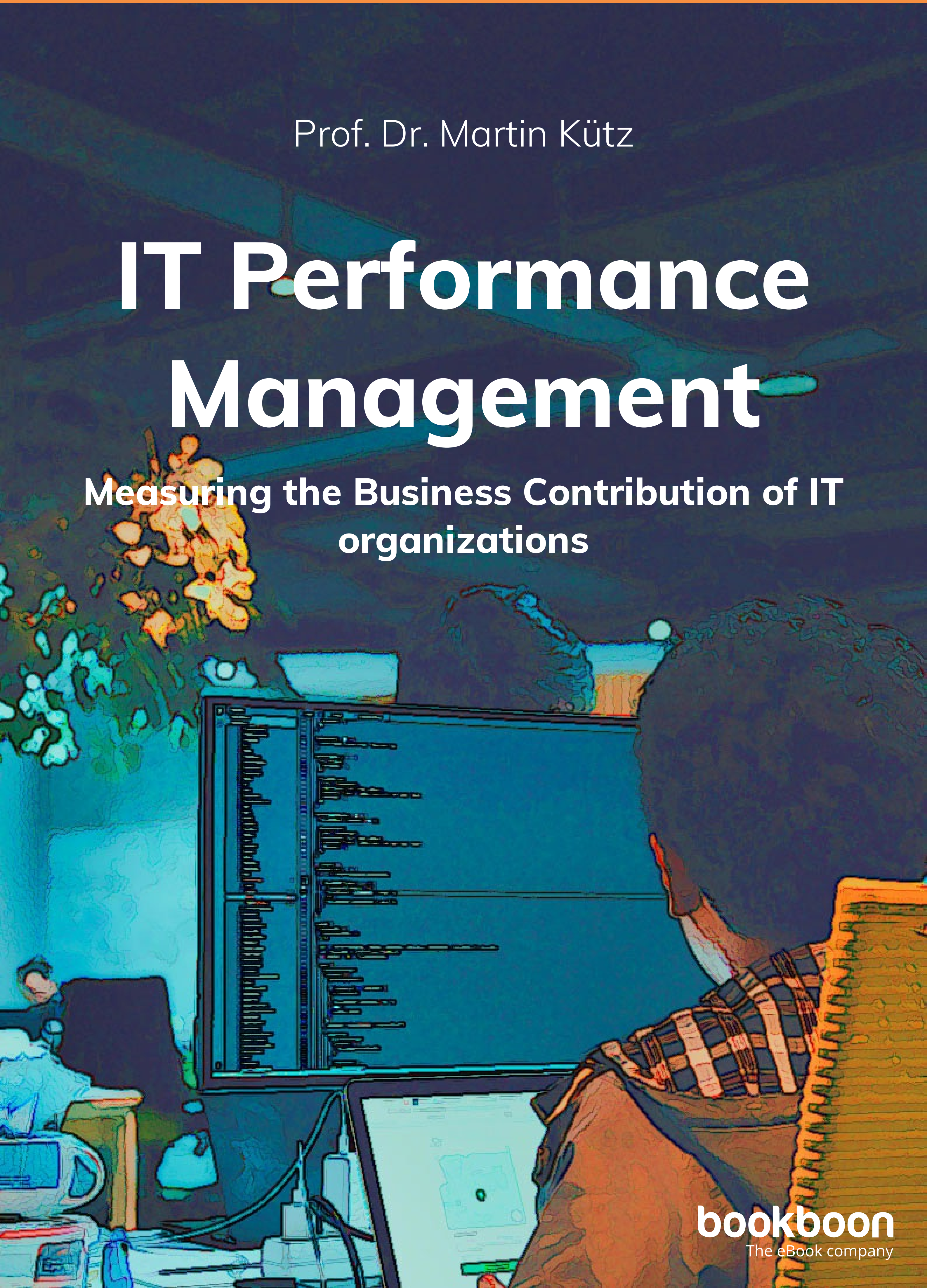
Prof. Dr. Martin Kütz

IT Performance Management

Measuring the Business Contribution of IT organizations



bookboon
The eBook company

PROF. DR. MARTIN KÜTZ

IT PERFORMANCE MANAGEMENT

MEASURING THE BUSINESS CONTRIBUTION OF IT ORGANIZATIONS

IT Performance Management: Measuring the Business Contribution of IT organizations
1st edition
© 2018 Prof. Dr. Martin Kütz & bookboon.com
ISBN 978-87-403-2603-1
Peer review by Prof. Dr. Can Adam Albayrak, Hochschule Harz,
Prof. Dr. Andreas Gadatsch, Hochschule Boon-Rhein-Sieg

CONTENTS


I joined MITAS because
I wanted **real responsibility**
The Graduate Programme
for Engineers and Geoscientists
www.discovermitas.com
Month 16
I was a construction supervisor in the North Sea advising and helping foremen solve problems
Real work
International opportunities
Three work placements
MAERSK

TABLE OF FIGURES

TABLE OF ABBREVIATIONS

AHP	Analytical Hierarchy Process
c.u.	currency units
CAD	Computer aided design
CDS	Cost distribution sheet
CEO	Chief Executive Officer
CI	Configuration Item
CIO	Chief Information Officer
CMDB	Configuration Management Data Base
COBIT	Control objectives for IT and related technologies
CPI	Cost Performance Indicator
ETC	Estimate to completion
GRC	Governance – Risk – Compliance
HR	Human Resources
IPO	Input – Process – Output
IT	Information Technology
ITIL	IT Infrastructure Library
KISS	Keep it stupid and simple
KPI	Key Performance Indicator
MAC	Moves – Adds – Changes
PDCA	Plan – Do – Check – Act
POS	Point of Service
ROI	Return on investment

ROSI	Return on Security Investment
SaaS	Software as a Service
SPI	Schedule Performance Indicator
TBP	Ticket-Based Process
TCI	Telephone computer integration
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
USP	Unique selling proposition
WBS	Work breakdown structure

1 MANAGEMENT, PERFORMANCE AND GOVERNANCE

Learning objectives

In this chapter you will learn,

- what management is and what managers do,
- how management and governance are related,
- why management has to focus on performance,
- what the specific challenges for IT management are,
- why we need a good IT performance management.

Recommended pre-reading

- Wiggers 2004, pp. 8 – 34

1.1 THE BASIC MANAGEMENT MODEL

Why do we need managers? Let us start with this simple question. Obviously a **manager**, be it a man or a woman, is responsible for something. However, this raises two questions. From where does he or she get this responsibility? And what does "responsibility" mean?

If a manager is responsible for something, then he or she has to manage some kind of a **management object**. With respect to IT (Information Technology) this could be an IT system, an IT project or even a complete IT organization. This management object has an **owner** and this owner puts the manager in charge of that specific management object. Manager and owner may be the same person or organization but they are different roles.

In business environments **owner** and **manager** are usually different persons or organizations. This is because the owner is not able to manage his management object himself, e.g. due to missing expertise, or he just wants to save his time and focus on other issues. Owners and managers have a specific relationship and the **principal-agent theory** is dealing with it (see Demartini 2014, pp. 12 – 13). In this theory the owner is denominated as the principal and the manager is denominated as the agent.

To do his or her job properly the manager needs to know what the owner expects him to do. Both have to negotiate on **targets** for the management object and have to reach a target agreement. An important part of this target agreement is the **time span** in which the negotiated targets have to be reached.

Thus our **basic management model** consists of five elements: Owner, manager, management object, targets and time span. The relationships between those five elements are shown in figure 1. Later we will extend this management model by a sixth element (see figure 4).

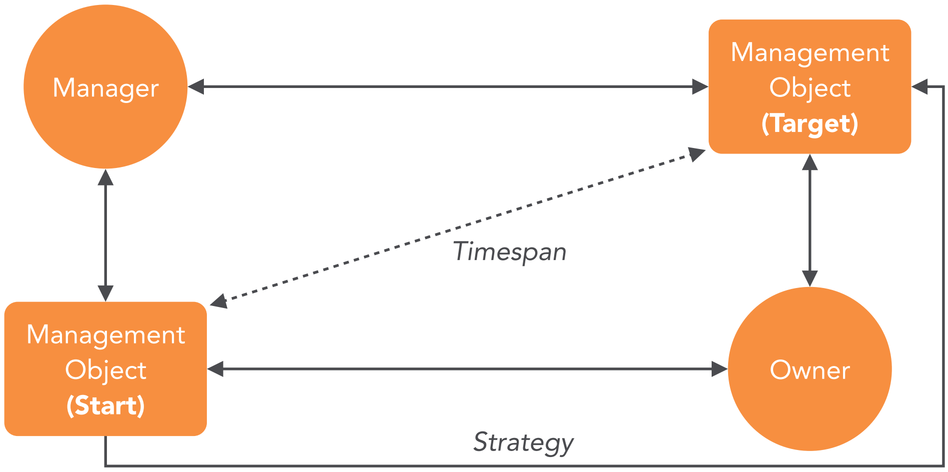


Fig. 1: The basic management model

1.2 THE MANAGEMENT CONTROL CYCLE

Now we will describe what a manager does after having made the target agreement with the owner of the management object (see figure 2).

First he makes a **plan** and sets up a strategy how and in which steps he is going to move the management object into the target position. After he has finished the plan and has got the allowance from the owner to proceed as planned the manager has to **transform** his plan into reality. He has to make the management object work as he planned it and he as the manager has to support this work. Parallel to this work the management object has to deliver data about its status and how it moves towards the given and agreed targets. The transformation must be accompanied by **monitoring**. We will consider monitoring in chapter 2 of this book deeply.

The data, which have been captured in the monitoring step, will now be prepared and documented appropriately, then be **reported** to the manager. Subsequently the manager executes one of the most fascinating management activities, the **deviation analysis**. Here he compares the actual status of the management object with the plan he has made initially. He also has to check whether he has to change the targets and / or his plan, e.g. due to recent business changes or actual technological developments.

If the manager comes to the result that there is a **gap** between the actual status of the management object and the intermediate target which was planned for the actual point of time or that the management object will not meet the final target at the end of the given time span because the expected performance of the management object differs from the planned performance or targets have changed in the meantime then he has to make a **decision** how to proceed now. He also has to **communicate** his findings to the owner of the management object and sometimes has to re-negotiate the target agreement. The communication activities should also include the various **stakeholders** of the management object, e.g. the people working in or with the management object, external business partners, concerned state authorities, other parts of the organization or the public around.

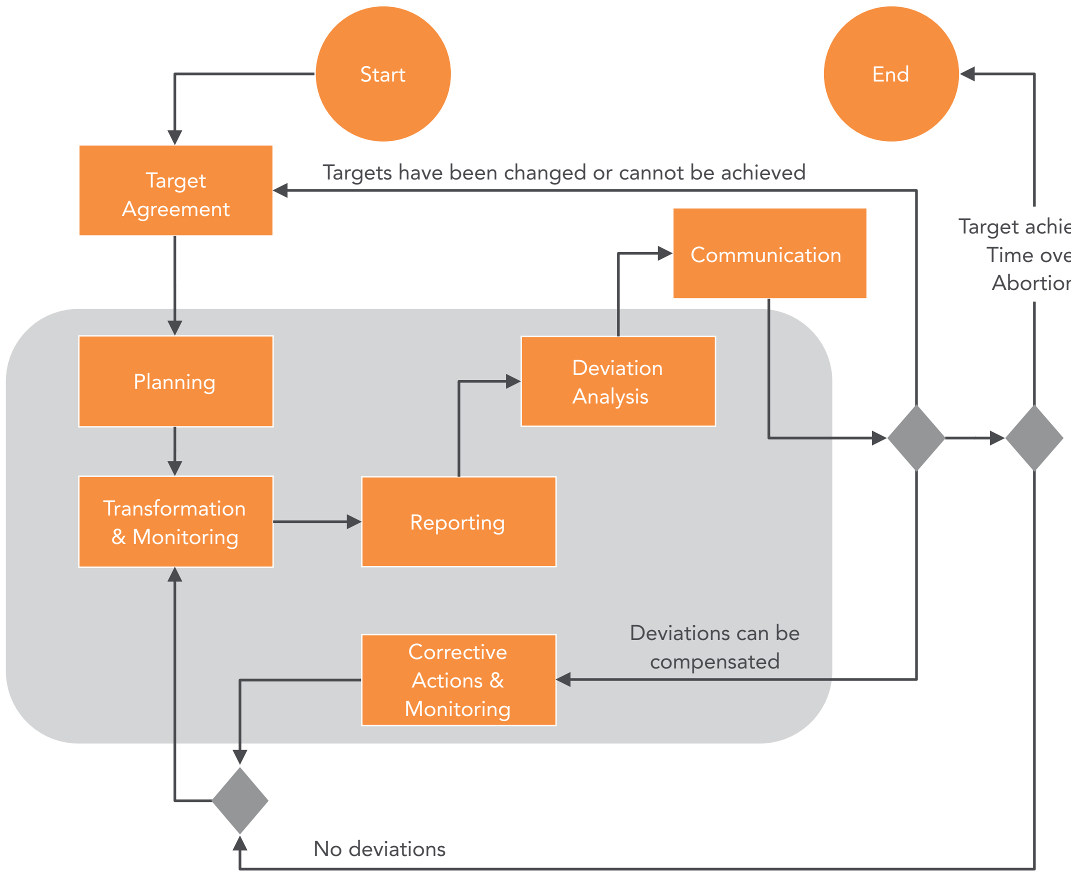


Fig. 2: The management control cycle

The decision, which must be made by the manager, has four basic alternatives:

- All is running and proceeding as planned. Nothing has changed unexpectedly. The work can be continued due to the original plan.
- Gaps between original plan and actual status respectively between (modified) plan and forecast have been found. However, the manager feels capable to reach the agreed targets. He will initiate specific activities in addition to or instead of the running process.
- Gaps between original plan and actual status respectively between (modified) plan and forecast have been found. But the manager does not feel able to meet the agreed targets and thus asks the owner to modify the target agreement or replace it by a totally new target agreement.
- The process can or must be stopped either because the targets have been met completely or the allowed time span is over or the consumed budgets has been consumed completely or it does not make sense to continue for some specific reasons.

These activities are connected by a specific logic and together with those relationships lead to the **management control cycle**, which is presented in figure 2. This management control cycle is derived from well-known PDCA cycle (PDCA = Plan / Do / Check / Act; see Kuster 2015, p391).

The manager has to run through all steps of the management control cycle again and again. It looks like a Sisyphean labor. Of course, management is hard work. However, we should consider the management control cycle as a logical order and not as an operational sequence. Starting and ending the different steps often is not as clear as we have discussed it here. Usually the different activities will run in parallel.

1.3 MANAGEMENT OBJECTS

As we have already seen it each management needs a management object, and each management object must have a responsible manager. Our subject in this book is IT performance management, and thus we have to look for typical management objects in the IT area.

Let us start at the end. The customer gets (a unit of) an IT **service**. To create and deliver a service we need **processes** and (technical) **systems**, and to realize the various systems we run **projects**. Thus we have identified the four basic management objects in the IT. IT performance management must consider each project, system, process, or service.

However, each IT organization runs a variety of projects, has a lot of technical systems, runs a complex network of processes, and offers different services to their customers. The entities of these four basic management objects are themselves management objects. Thus we have – on a higher level – **project portfolios**, **system portfolios** (or system architectures), **process portfolios** (or process landscapes), and **service portfolios** (or service catalogues) as management objects.

What else can we consider to be a management object in an IT environment? The most obvious management objects are the **resources** resp. the conglomerate of all resources, we need. Each IT organization has a resource management, and accordingly it needs a resource performance management.

Another candidate for IT performance management is **GRC** (Governance – Risk – Compliance), and information or IT **security**. Finally we come to the conclusion, that all items, which are considered as management items, are management objects, and IT performance management has to investigate them.

In addition, **organizational units** or **total organizations**, which are focused on IT, are management objects, and an appropriate performance management has to support the management. We can close this investigation by identifying **firms**, which run an IT or an IT focused business. They are management objects, too. Here IT performance management and general performance management converge.

Figure 3 shows the hierarchy of the discussed IT management objects.

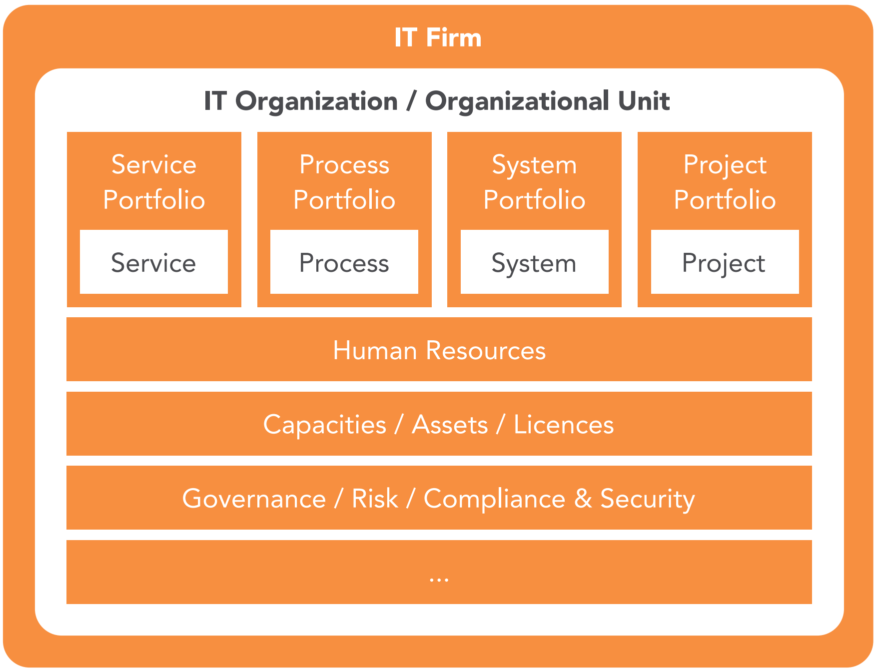


Fig. 3: The hierarchy of IT management objects

1.4 PERFORMANCE MANAGEMENT

Now we are ready to discuss the term performance management. Is it a specific category of management? Is it an integral part of every management? Or is it something else?

To begin the discussion we refer to Wiggers 2004, p15: "Performance management concerns the way, in which the performance of an organization or organizational unit is managed. Therefore it is an important part of the management control cycle." This simply means that the **management control cycle**, which we have introduced in chapter 1.2, is a metaphor of performance management. But what is performance?

We will say that our management object shows a good **performance** if it moves towards the targets as planned. However, this (constantly improving) nearness to the targets is only one side of the coin, which is called **effectiveness**. Management has to reach those targets in a given **time span** as we already discussed as one of the elements of our basic management model. In addition management is not allowed to consume arbitrarily many resources and resource quantities. The given targets have to be approached efficiently and this technical and economic **efficiency** is the other side of the performance coin.

We can summarize this in the following way.

Management aims at reaching targets. The management object has to generate some output, which is beneficial for the owner. Performance means that those targets are met in a minimal (or accepted) time span and with minimal (or accepted) resource consumption. Performance management is an integral part of management; management without performance management would be a contradiction in itself.

Performance management is the aspect of management, which is dedicated to **quantities** and **measurability**: "The question of what to measure and how to measure are fundamental to performance management." (see Wiggers 2004, p. 15). Hence in this book we always will ask for the measurement of **output**, **input**, and **time**, because measurement is not given. It must be designed and built.

1.5 IMPLEMENTATION OF PERFORMANCE MANAGEMENT

If we consider big organizations we will find specific people being denominated as **performance managers**. We will even find organizational units concerning themselves with performance management or performance analysis. Does this make sense due to our finding that performance management is an integral part of every management?

To understand why we have **performance management experts** or even **performance management units** let us think of a person who has a good business idea and starts a small business. The management object now is the new and small firm (the garage company). The founder is as well owner as manager of this organization and employer and employee similarly. Of course does he run through all steps of the management control cycle and has to conduct all relevant activities like planning, monitoring, deviation analysis and decision making.

Now let us assume that the small company is successful. Revenues go up, more and more customers are asking for the very good products or services. More and more resources including human resources are needed to run the business. At the same time the management task increases, too. More and more planning, monitoring, deviation analysis and decision-making activities have to be conducted.

But every **management decision** takes time. It is not enough to say “yes” or “no” or “Choose alternative 3!” but rather decisions must be defined and prepared and then the best alternative has to be identified. Then the decision, which was made, has to be implemented. People must be informed and convinced. And last but not least it must be checked whether things were done as management wanted it to be done.

So, if the business grows more and more time is needed for management activities. But the time of the founder is limited. What can he do? At the end he has two alternatives.

The first alternative is to find **sub-managers** and delegate parts of his management job to them. He resigns a management object to a sub-manager and enters into a target agreement. So he establishes a management hierarchy in his organization.

The second alternative is to consider his management control cycles and find ways to increase his decision-making capacity. He should delegate all activities, which are **supporting the decision making**, e.g. creating plans (The acceptance of a plan is a decision and cannot be delegated), conducting monitoring activities, analyzing deviations and supervising the implementation of decisions.

We will call the people, who take over supporting activities from managers, performance managers. They will help “their” managers to make **good decisions**. The managers are their “customers”.

1.6 MANAGEMENT SYSTEM

After we have described that performance management is an integral part of every management activity and how it comes to performance management experts in big organisations let us now consider performance management as a specific area of activities and responsibilities. We assume that between the manager and the management object there is a complex **management system**. This management system consists of actors, roles, processes, and of course technical systems. It includes the manager as well as the management object and is not at all restricted to a hardware or software systems though such systems will be part of that management system. Thus we can extend our basic management model as it is shown in figure 4.

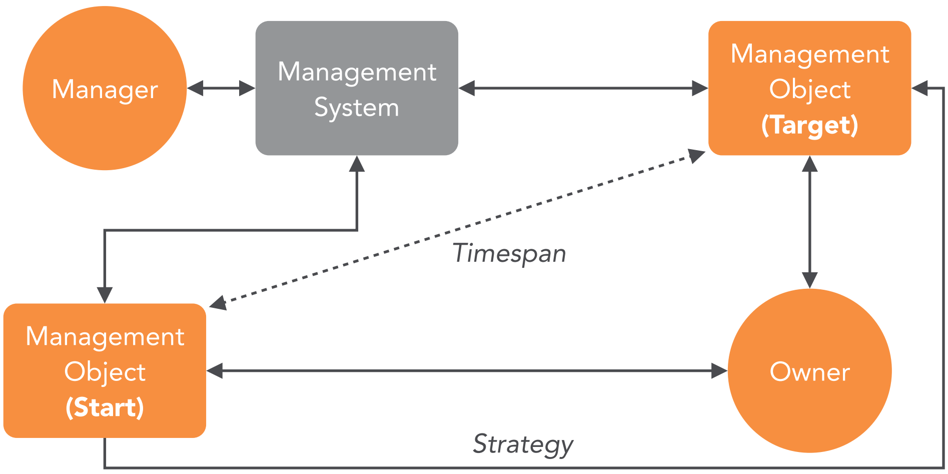


Fig. 4: The extended management model

If we have such a management system to support management then it must be built, administered, and the manager as the primary user needs training and help. Thus we can define the fascinating job of a performance management, as it is shown in figure 5.

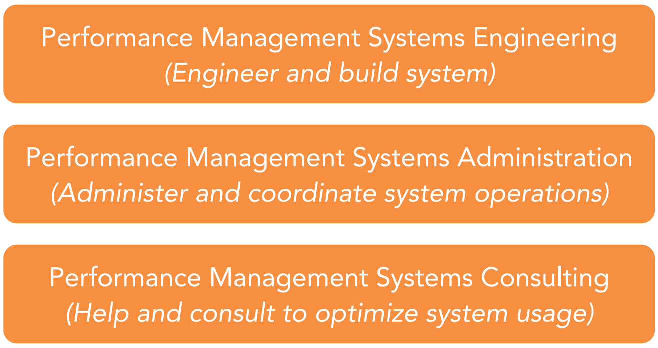


Fig. 5: Areas of performance management

The performance management expert has to care for the management system in total. Because he is strongly involved in all management activities he will sooner or later become a general management expert. In several big organizations it is good practice that top-level managers have to work as performance management experts for some time. This is considered to be an indispensable career step for management offspring.

1.7 PERSPECTIVES OF IT MANAGEMENT

Finally let us have a look on IT management. What are the characteristics of IT management so that we better understand what we should expect from IT performance management?

If we talk about IT, than we have to talk about two sides, the side, which **provides** and **operates** IT systems, and the other side, which **utilizes** those IT systems.

The first side is represented by organizations, which build, operate and maintain IT systems, consisting of various hardware and software sub-systems and produce and provide related services. The other side is represented by organizations, which use those IT systems and the related services to support and improve their business processes.

The first side is called **IT supply**. It may appear as companies whose business is IT, which they sell to external markets. Their customers can be other firms, public authorities or any association. IT supply may also appear as organizational units within firms and other organizations and deliver IT services to other organizational units in the superior organization. The manager of such an internal IT unit usually is denominated as IT manager.

The second side is called **IT demand**. It may appear as a company or any other organization, which is buying IT services in external markets. It may also appear as organizational units in firms, public authorities or any association, which need IT systems and IT services to support their business processes. The manager of such a unit is the regular unit manager. IT demand management is just a (small) part of his total management task.

If IT supply and IT demand meet on external markets then state authorities and laws govern activities and maintain the order on the public markets. If we move into an organization, e.g. a privately owned company, then it may have one or more internal IT supply organizations, and it will have several internal IT demand organizations. It also may have one or more external IT suppliers. To handle that network of IT supply and IT demand organizations it must establish internal **IT governance**, which sets the rules for all IT related activities in that organization. Responsibility for IT governance is often assigned to the CIO (Chief Information Officer) but is also seen as a natural task for the CEO (Chief Executive Officer, see Weill 2004, pp. 117 – 146). The interaction between IT supply, IT demand and IT governance is shown in figure 6.

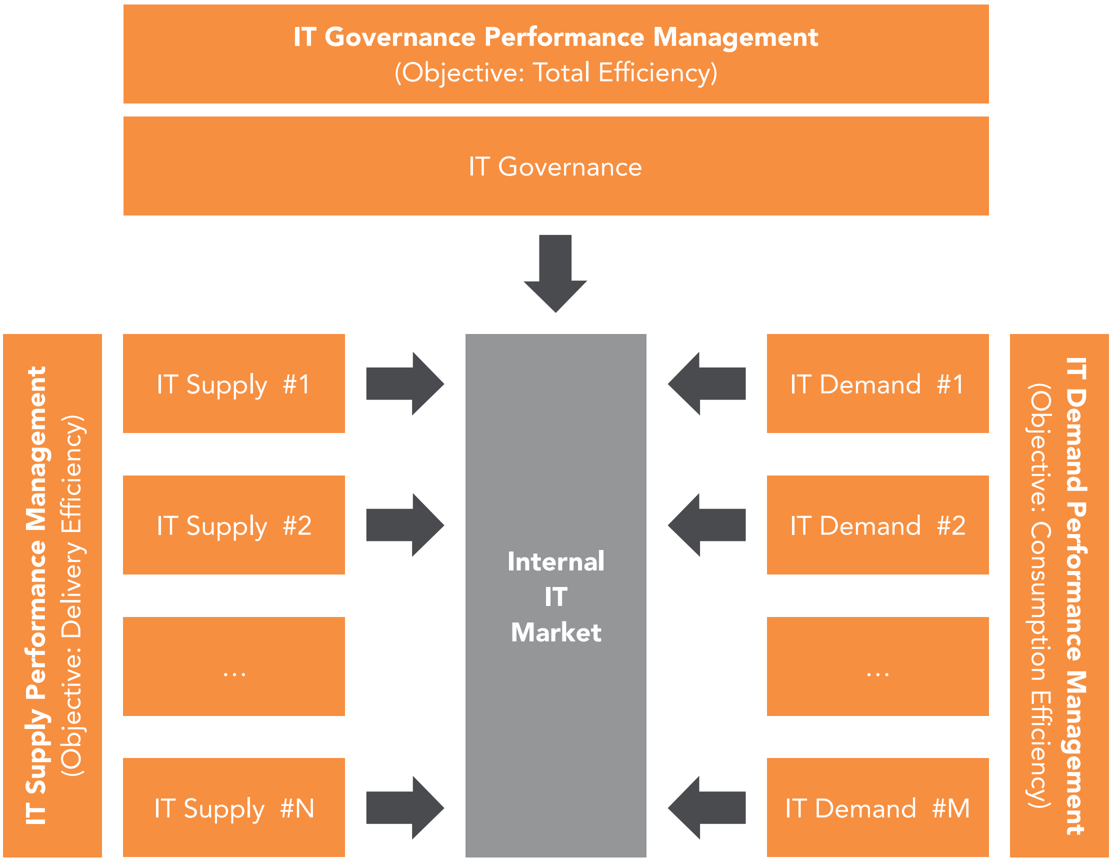


Fig. 6: Internal IT market and IT governance

What does this mean for our subject IT performance management? The conclusion is, that IT supply, IT demand and IT governance are different categories of IT (related) management. Accordingly they all have and need a specific IT performance management. Due to the understanding of management, which is taken as the basis here, we cannot concentrate the total IT performance management in an organizational unit, which is responsible for IT governance.

IT governance is established as principles, policies, standards, instructions, and controls to ensure that IT supply and IT demand are practiced in the optimal way for business and organization. The major task of IT governance can be concentrated into two simple questions:

- **Who is allowed or obliged to deliver which IT systems and IT services?**
- **Who is allowed or obliged to use which IT systems and consume which IT services.**

Finally let us give some examples of related issues, which show that IT governance is active:

- Definition of **application systems**, e.g. ERP or desktop software, which have to be used in the whole organization,
- Definition of authorized internal and external **suppliers**, e.g. for desktop equipment or mobile devices,
- Definition of **IT infrastructure**, e.g. network technology or database management systems,

- Setting of **software strategies**, e.g. self-made software or software packages, usage of open source or proprietary software,
- Setting of **sourcing strategy**, e.g. outsourcing, insourcing, usage of cloud services,
- Setting of **guidelines** for human resource management in the IT area.

1.8 CHALLENGES

It seems that things are changing dramatically today. Every business and public administration area is in a **process of digitalization**. Business processes are more and more automatized, autonomous systems come up and replace human workers. Many IT services become a commodity and the need for in-house data centers and IT experts decreases significantly. We can reasonably forecast that the IT portion in the business processes increases continuously but the traditional IT department is going to disappear.

Hence IT management actually dominated by the management of IT units will change significantly. There will be severe consequences for the related IT performance management. It may be that IT supply management will more or less be changed to a coordinator of the various external IT suppliers or even completely disappear from organization charts. But the demand side will need more IT expertise and the portion of IT management will increase here. Also the importance of IT governance will go up. However, there will be a lot of IT management and correspondingly a lot of IT performance management. The higher the level of digitalization is the more IT and IT performance management will be necessary…

1.9 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.1!

1.9.1 QUESTIONS FOR YOUR SELF-STUDY

Q1.1: Literature often refers to the PDCA cycle as the model of the management control cycle. We have presented a different approach here. Please, find out and comment the differences between our model and the PDCA cycle.

Q1.2: We have presented a hierarchy of IT management objects. Please, give examples for each mentioned IT management object.

Q1.3: Explain the difference between effectiveness and efficiency. What does Wikipedia mean with respect to those two terms?

Q1.4: We have stated two questions to describe the scope of IT governance. Please, find examples, which give specific answers to these questions.

Q1.5: Why do we have to differentiate between delivery efficiency and consumption efficiency?

1.9.2 PREPARATION FOR FINAL EXAMINATION

T1.1: List and describe the elements of the basic management model. How does the extended management model differ from the basic management model?

T1.2: List and describe the steps of the management control cycle.

T1.3: What are the four basic management objects in IT organizations? Please, give two examples for each of the comprehensive management objects in the IT ecosystem.

T1.4: What are the three key areas of (IT) performance management?

T1.5: Please, describe the role of IT governance.

1.9.3 HOMEWORK

H1.1: Find other models of management systems in the literature and other publications. Find out and describe similarities as well as differences to the approach presented in this book.

H1.2: Look for general definitions of performance management resp. performance analysis. Find out similarities and differences to the approach presented in this book.

H1.3: If your organization does not have an own IT department but does use IT systems and IT services, do they need an IT manager and an IT performance manager? Could / Should they combine the IT management task with other management tasks? Which other management tasks would fit best to the IT subject?

2 MEASUREMENT, METRICS AND MONITORING

Learning objectives

In this chapter you will learn,

- how we can manage complexity,
- what are important characteristics of our management objects,
- how we can determine, whether we have already reached our targets,
- what the relationship between the management control cycle and our metrics is,
- how we can get the numbers, which we need for measurement,
- why monitoring is the basis for successful quantitative management.

Recommended pre-reading

- Demartini 2014, pp. 163 – 176

2.1 PERFORMANCE

We have learned that performance management is an integral part of management and that it is the **quantitative side of management**. What does this mean for the manager and the owner of the management object under consideration?

To find this out let us first focus onto the management object. For each such management object there will be some **output**. To produce this output there will be a corresponding consumption of some **input**. And the transformation of those inputs to the expected output will take some **time**. Our management object will show a good **performance**, if much output is produced in a short time while consuming few input (see figure 7).

Fig. 7: A quantitative model of performance

2.2 THE OUTPUT-INPUT MODEL

A first approach to the measurement of performance could be to set up an **output vector** $\underline{s} = (s_1, s_2, \ldots, s_n)$ and a corresponding **input vector** $\underline{r} = (r_1, r_2, \ldots, r_m)$. The number $s_i \geq 0$ is the quantity of the output with the index *i*, and *n* is the number of the different outputs which are created by our management object. Similarly $r_j \geq 0$ is the quantity of the input with the index *j*, and *m* is the number of the different inputs needed by our management object.

If we now define a **timespan** $[t_0, t_k]$, which is defined as the **planning period** for the management task then in the beginning at t_0 we will have $\mathbf{s}(t_0) = (s_1(t_0), s_2(t_0), \ldots, s_n(t_0)) = (0, 0, \ldots, 0)$ and $\mathbf{r}(t_0) = (r_1(t_0), r_2(t_0), \ldots, r_m(t_0)) = (0, 0, \ldots, 0)$, which means, that in the beginning of the planning period there will be neither any output created nor any input consumed. At the end t_k of the planning period we want to have $\underline{s}(t_k) = (s_1(t_k), s_2(t_k), \ldots, s_n(t_k))$ and $\underline{r}(t_k) = (r_1(t_k), r_2(t_k), \ldots, r_m(t_k))$. That means, that the content of our **target agreement** between the owner of the management object and the manager is to create the output quantities $s_i(t_k)$ and to this end consume the input quantities $r_j(t_k)$. Of course, this must be accomplished within the timespan $[t_0, t_k]$.

Often the manager as well as the owner will be happy, if the really produced output quantities $s_{i,a}(t_k)$ are greater than the planned output quantities $s_{i,p}(t_k)$. But as a performance manager we should be careful. If we just need the planned quantities and not more, then the production of more output than needed is **waste**. Thus we have to consider the **target setting** very carefully. Does the output target be: Produce as much as possible, but at least $s_{i,p}(t_k)$? Or is it: Produce exactly the quantity $s_{i,p}(t_k)$, not less, not more?

We also should handle the input side with care. Our naïve assumption will be that we have a good performance if we have consumed less input than planned, which means that the really consumed input quantities $r_{j,a}(t_k)$ are smaller than the planned input quantities $r_{j,p}(t_k)$. But again we have to consider the **target setting** very carefully. Is the input target really: Consume $r_{j,p}(t_k)$ at the most? Or is it: Let the consumed input quantities $r_{j,a}(t_k)$ be in a good balance with the produced output quantities $s_{i,a}(t_k)$? In this latter case we have to define very precisely what a “good balance” is. Hence we learn, that target setting seems to be more complicated and challenging as it looks at first glance.

However, our output-input model raises several **questions**, which we have to resolve in a real-life situation precisely:

- We have assumed that *n* different output quantities and *m* different input quantities are **relevant** for our target achievement. Are we sure that we did not overlook a relevant output or input?
- We have assumed that we consider output and input quantities. But what are the **counting units**? What is the output quantity of an IT service, an IT process or an IT project. The same yields for input quantities. Are they clearly defined? What is a work unit of a human being? And last, but not least, how can we really **measure** what is going on with our management object?
- In a production environment we will often need **stocks**, output stocks because we have produced something which is not taken immediately by the customer, input stocks because we have purchased input quantities, which we will feed to our management object later. How do we have to include those stocks into the target setting?
- We already mentioned that there will be a relation between the output and the input quantities. Theoretically there is a **production function** $r_j = f_j(s_1, s_2, \ldots, s_n)$. But do we really know those production functions? Are we able to find them out precisely in our complex technical and organizational environments?

2.3 AVAILABILITY, RESPONSE TIME, AND CAPACITY

Now we have described the **output-input model**. And it seems to be a reasonable model. But does it fit well to the IT environment? To understand the specific challenges of IT performance management let us consider an IT application system. What are our expectations as users of the system with respect to its good performance? First of all, the system must be up and running so that we will not have to wait until the system is ready for usage. Secondly, if we send an instruction to the system we want to have it executed immediately. This means, that the execution should start as soon as possible and the time of execution should be appropriately small. And third of all, we want to have the same performance of the system every time regardless of the number of people actually using the system at this point of time.

The first characteristic, namely, that the system is up and running, and waiting for our request, is related to the concept of **availability**. The system or any management object is available, if it gets a request from an outside customer and starts to work on it within a given span of time. The customer should not wait for a "long" time.

The second characteristic, a short execution time, is related to the concept of **response time**. If the system is available then the order of the customer should be processed in an appropriately short time.

Finally the third characteristic, that availability and response time do not go down even if the number of system users increases significantly, is related to the concept of **capacity**. Each system be it an application system or a complex socio-technical system like an IT department will have a specific capacity. This means that it will not be able to handle an infinitely large number of customer orders at the same time. There will be an upper limit for this number of simultaneously processed orders. And the performance management will have to find out how big the capacity of the management object really is and how much of this capacity has been utilized or not.

How do availability, response-time and capacity fit into our output-input model?

Availability is related to the output side of our management object. If it is not available then it is not able to produce output. The model assumes that an output is only created if there is a request for output. Later (see chapter 2.8) we will describe strategies to measure the availability of a system. See also chapter 2.6.

Response time is also related to requests for output. It is not a quantity but a property of an output unit. So the target with respect to response time will be that a given portion of output units should show a specific response time. We will also consider this a little more technically later on (see chapter 2.8).

Finally the **capacity** issue belongs to the input side of our management object. If we consider capacity as some kind of space, which we need for creating the output, then the usage of that space can be considered as the consumption of an input.

The conclusion of the previous considerations is that availability, response time and capacity fit well into our output-input model. This leads to a **general assumption**, which will be the basis of all subsequent discussions.

Each characteristic, which is relevant for performance management, can be integrated in the general output-input model.

We cannot proof this like a mathematical theorem, but up to now we have not found any example, which refutes this assumption.

2.4 MEASURING THE TARGET ACHIEVEMENT

With our output-input model we have provided a structure, which allows us to resolve our **measurement issue** with the help of mathematical methods. Let us consider the concatenation of the output and the input vector of our management object $\underline{x} = \underline{s} \oplus \underline{r}$. Now the status of our management object is represented by the vector $\underline{x} = (x_1, x_2, \ldots, x_{m+n})$. And in such a vector space we can define a **function**, which is measuring the **distance** between two vectors $\underline{x}$ and $\underline{y}$, having the following properties:

$$d(\underline{x}; \underline{x}) = 0,$$
$$d(\underline{x}; \underline{y}) \geq 0,$$
$$d(\underline{x}; \underline{y}) = d(\underline{y}; \underline{x}).$$

A well-known distance function in an (n+m)-dimensional vector space is the **Euclidean metric**

$$d\left(\underline{x}, \underline{y}\right) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

With such a distance function we can define the target achievement by $d(\underline{x}_a; \underline{x}_p) = 0$. Here $\underline{x}_a$ denominates the actual status of the management object and $\underline{x}_p$ denominates the planned status of the management object.

However, from a psychological point of view it would be more appropriate if we had a **target achievement level function** with $0 \leq t(\underline{x}; \underline{y}) \leq 1$ and $t(\underline{x}; \underline{x}) = 1$. Can we find such a function and use the distance function d for it?

The answer is: yes. For example we can choose

$$t\left(\underline{x}, \underline{y}\right) = \frac{1}{1 + d(\underline{x}, \underline{y})}$$

$$t\left(\underline{x}, \underline{y}\right) = e^{-d(\underline{x}, \underline{y})}$$

Both functions are used in various sciences to measure **similarities** between complex objects. If $d(\underline{x}; \underline{y}) = 0$ then $t(\underline{x}; \underline{y}) = 1$.

Another approach is based on **geometrical considerations**. We assume to start at $\underline{0}$. Our target point is $\underline{x}$, but we have actually reached $\underline{y}$. If we want to get to the planned point $\underline{x}$, then we still have to cover the distance between $\underline{x}$ and $\underline{y}$ and the total distance for us to be covered will be the distance between $\underline{0}$ and $\underline{y}$ plus the distance between $\underline{y}$ and $\underline{x}$. This leads to

$$t\left(\underline{y}, \underline{x}\right) = \frac{d(\underline{0}, \underline{y})}{d\left(\underline{0}, \underline{y}\right) + d(\underline{y}, \underline{x})}$$

Obviously $t(\underline{0}; \underline{0}) = 0$ due to $d(\underline{0}; \underline{0}) = 0$. And $t(\underline{x}; \underline{x}) = 1$ due to $d(\underline{x}; \underline{x}) = 0$ (see chapter 9.8).

Now let us combine the measurement in the state space of our management object with the management control cycle. What is going on in the management control cycle in terms of measurement?

2.4.1 STEP 1: TARGET AGREEMENT

In this first step the manager and the owner of the management object negotiate and agree on targets. This means an agreement on the relevant output and input quantities. They both assume that the management task starts with $\underline{x}(t_0) = \underline{x}_p(t_0) = \underline{x}_a(t_0) = \underline{0}$. They agree to have $\underline{x}_p(t_k)$ at the end of the timespan $[t_0; t_k]$.

2.4.2 STEP 2: PLANNING

Now the manager prepares a plan. That means, that he subdivides the total timespan $[t_0; t_q]$ by points of time t_k with $t_0 < t_1 < t_2 < \dots < t_k < \dots < t_{q-1} < t_q$. It may be that these points of time have been already set within the target agreement. Usually they will be equidistant on the time axis. If $[t_0; t_k]$ is a year then the intervals could be the 12 months of that year ($q = 12$).

During his planning activities the manager will define intermediate targets $\underline{x}_p(t_k)$. Due to our interpretation the values of the coordinates as output and input quantities of those points will increase if t_k increases. His or her plan will be

$$\underline{x}_{p,0}(t_1),\ \underline{x}_{p,0}(t_2),\ \underline{x}_{p,0}(t_3),\ \dots,\ \underline{x}_{p,0}(t_k),\ \dots,\ \underline{x}_{p,0}(t_{q-1}),\ \underline{x}_{p,0}(t_q)$$

Finally the manager will present his plan to the owner of the management object and ask for his agreement to the plan.

2.4.3 STEP 3: TRANSFORMATION AND MONITORING

Now the management object starts to work. While it is running data must be collected and stored for the subsequent reporting step. The manager has to care for the monitoring.

2.4.4 STEP 4: REPORTING

If the next reporting point is approached the monitoring data must be taken and prepared for the management. The values or coordinates of the different output and input characteristics are calculated from collected raw data. Also the target achievement level indicators will be calculated.

2.4.5 STEP 5: DEVIATION ANALYSIS

At t_1 we have reached $\underline{x}_a(t_1)$. It is probable, that we have $\underline{x}_{p,0}(t_1) \neq \underline{x}_a(t_1)$. If the deviation is small then the manager may come to the conclusion, that he / she will easily get back to the planned status trail $\underline{x}_{p,0}(t_k)$ for $2 \leq k \leq q$. Otherwise he or she will have to do some re-planning and come to

$$\underline{x}_{p,1}(t_2),\ \underline{x}_{p,1}(t_3),\ \underline{x}_{p,1}(t_4),\ \ldots\ ,\ \underline{x}_{p,1}(t_k),\ \ldots\ ,\ \underline{x}_{p,1}(t_{q-1}),\ \underline{x}_{p,1}(t_q)$$

If $\underline{x}_{p,1}(t_q) = \underline{x}_{p,0}(t_q)$ then the manager will proceed but may have to initiate correcting activities in step 7. If $\underline{x}_{p,1}(t_q) \neq \underline{x}_{p,0}(t_q)$ then the manager has to contact the owner and re-negotiate the target agreement in step 6. It may happen that there is no deviation inside but due to changes being initiated from outside the target has be changed to $\underline{x}_{p,1}(t_q) \neq \underline{x}_{p,0}(t_q)$. Then, of course, a similar re-planning is necessary.

At t_2 we have reached $\underline{x}_a(t_2)$. It is probable, that we have $\underline{x}_{p,1}(t_2) \neq \underline{x}_a(t_2)$. If the deviation is small then similar to the activities in t_1 the manager may come to the conclusion, that he / she will easily get back to the planned status trail $\underline{x}_{p,0}(t_k)$ for $3 \leq k \leq q$. Otherwise he or she will have to do some re-planning and come to

$$\underline{x}_{p,2}(t_3),\ \underline{x}_{p,2}(t_4),\ \underline{x}_{p,2}(t_5),\ \ldots\ ,\ \underline{x}_{p,2}(t_k),\ \ldots\ ,\ \underline{x}_{p,2}(t_{q-1}),\ \underline{x}_{p,2}(t_q)$$

This investigation and re-planning will be done for each reporting point until t_{k-1}.

2.4.6 STEP 6: COMMUNICATION

The results of the deviation analysis have to be communicated to the owner as well to other stakeholders. If a re-negotiation of the target is necessary then it must be done now. It may happen that the discussion with the owner leads to another re-setting of the target and thus makes another re-planning necessary. Then another planning step has to be inserted here.

2.4.7 STEP 7: CORRECTIVE ACTIONS AND MONITORING

If deviations from the plan have been identified, and management feels, that the planned targets can still be reached, corrective actions must be defined, initiated, and executed. Those actions themselves have targets, and thus the execution must be combined with an appropriate monitoring to see, whether the corrective actions are successful or not.

What can we learn from these considerations? Firstly: Characteristics and our state space model fit perfectly to the **control cycle model**. The measurement of the selected characteristics allows to find out whether we are "on track" with respect to our target status trail. Secondly: Planning is not a one-time activity and an exceptional part of management work. On the contrary it is the central element of management, and planning or re-planning is an **on-going activity**. Deviations can only be discovered if we have a sound plan.

2.5 PERFORMANCE AND ENVIRONMENTAL CHARACTERISTICS

Let us now come back to the characteristics. We started with the hypothesis that all **sub-targets** of a management object can be described by some **output and input quantities**. However, the real world is a bit more complicated.

In a service or request driven environment it obviously does not make sense to negotiate and plan delivered output quantities because this depends on external forces. We cannot plan the behaviour of our customers. The output, which is under the **responsibility of a manager** here, is availability. Every time when a customer request comes in then the management object has to be able to fulfill the customer request. But can we neglect the quantities of these requests? The answer, of course, is: No.

Obviously there are different categories of characteristics, which we have to consider. First we have the **performance characteristics**. They are part of the target agreement between owner and manager. Secondly there are **environmental characteristics**. External forces drive them, and both, the owner and the manager, have to accept them. However, they are very important for the target setting. The target values, which are set for the performance characteristics, are significantly influenced by those environmental characteristics. They are like weather data. If it is cold and rainy, and you have to reach a target point then you will behave differently from reaching that target point if it is warm and the sun is shining. It also may happen that in case of bad weather you will have to choose another target point.

The situation is similar here. Owner and manager have to make up their mind on the relevant environmental characteristics and they have to define the values, which they **expect** or **forecast**. Based on this information they can negotiate about performance characteristics and related target values.

The example of the **service environment** can demonstrate this. Of course you will make an assumption on the volume of the incoming customer requests, and then set up corresponding targets on availability. In addition this will lead to a target with respect to the needed input quantities often being represented by needed capacities.

The difference between performance characteristics and environmental characteristics can be demonstrated quite easily. The owner will have to pay a salary to the manager, and he may wish to pay him a variable salary so that he gets money according to the target achievement level he or she reaches. Then performance characteristics are those, which could (theoretically) be taken as a basis to calculate the payment to the manager, and the environmental characteristics are those, which cannot be made a parameter for the determination of the manager´s salary.

Thus we have to generalize the status vector model of our management objects and have to differentiate between the **performance status vector** and the **environmental status vector** (see figure 8). The manager has to observe both.

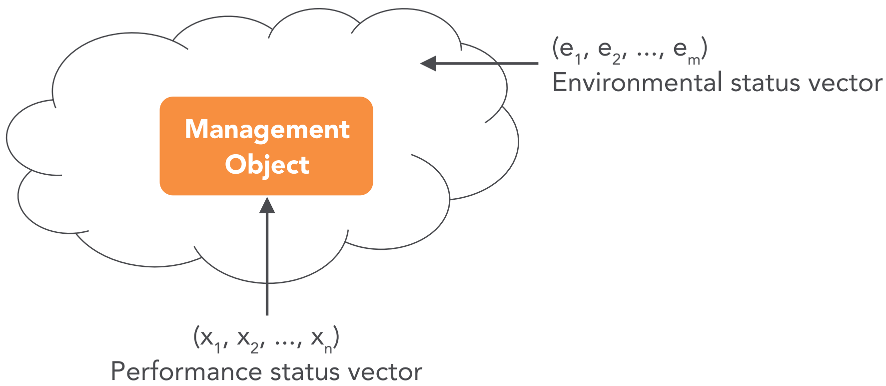


Fig. 8: Performance and environmental characteristics

2.6 TYPICAL EXAMPLES OF PERFORMANCE CHARACTERISTICS

Typical characteristics of management objects are output quantities and input quantities. But often these characteristics are not a primary part of the target. Other characteristics have to be investigated. However, in the real IT world the acting people are usually quite sure which characteristics are performance characteristics and which are environmental characteristics. But this doubtlessness is often only given on the level of the term. And we have to find out how we can interpret such terms with respect to **quantification** and **measurability**.

So let us discuss some typical performance terms and find out what they mean with respect to measurability.

Availability means that the management object takes over a **request for output** and the requestor knows that his or her request has been registered and will be processed in a defined manner. This **registration** could be communicated to the requestor by sending a message. Now availability just means the ratio of the number of requests with a positive registration message and the number of all incoming requests. All readers who sometimes sit before the computer screen and wait for some feedback from the machine know that measuring those two quantities obviously is not as easy as it looks like. The challenge is to implement this measurement technically.

Run time (Duration) means that the request for output is processed in a specific timespan and that the requestor for output does not have to wait for his or her output too long (whatever that exactly means). But what does duration mean from out point of view. Does it mean average duration, which is the mean value of all single duration measurements? Or does it mean the portion of all requests for output, which have been processed faster than a given duration time? And finally we have to ensure that we measure the starting of each single request for output and the ending of this processing as well. If we are not able technically to measure and to document the starting time and the ending time of a request for output then we are not able to make any judgment about duration.

Efficiency in the considered request driven environment means that the needed input is provided according to the requested output, in particular, that not more input is consumed than it is needed to produce the provided output. Hence, efficiency is nothing but a ratio of an output quantity and an input quantity. The problem is that we have to aggregate different output categories and different input categories. We have to find quantitative relations between the units of the different outputs resp. inputs. This will lead to an aggregation by weighted sums where the weights show the relation of the considered output or input unit to a standard output or input unit. If the outputs are sold on external markets then the output weights can be the market prices. Inputs can be considered from a cost point of view, and thus we can take the weighted input quantity as the cost for our input. However, to calculate an efficiency value we have to be able to measure the relevant output quantities and input quantities and we have to be able to determine the weights.

Capacity utilization means that provided input is optimally or completely used. While efficiency considers the consumption of input, capacity utilization considers the providing part of efficiency. Measurement of capacity utilization means that we are able for every capacity unit to find out, whether it is in use at a specific point of time or not.

Output quality means, that the delivered output has all the characteristics, which the output requestor is expecting. If an output unit does not have the expected or agreed quality then the requestor will not accept it and this output unit has to be destroyed and the corresponding input has been wasted. Obviously bad quality leads to bad efficiency. And quality is the difference between delivered and accepted output. However, quality has to be measured. Normally we try to identify missing quality.

Adherence to schedules means that requested output is delivered just in time, not later and not earlier. It depends on the specific business, whether quick response or delivery just in time is preferred by the requestor. It may also depend on the specific output. However, we must be able to measure the time of specific events.

Reliability means, that a request for output, which has been accepted (see availability), is processed and the requested output is really delivered to the requestor. Reliability overlaps in some way with response time and meeting deadlines and quality.

Stakeholder satisfaction means, that the considered stakeholder is satisfied with the work of the manager or management object or with the outputs delivered by the management object. This depends on the considered stakeholder. A specific satisfaction category in the IT environment is the customer or user satisfaction, and employee satisfaction. The customer is a person or role who has the competency to make contracts with the manager. The user is a person or role, who or which asks for output units. And the employees are those people who work in the management object and are parts or elements of that system. Normally employee satisfaction is measured for the members of IT organizations or organizational units in IT organizations.

Every reader of this book will easily find additional examples of performance indicators. In many cases he or she will find out this his or her specific example is a special case of the categories mentioned before.

2.7 TYPICAL EXAMPLES OF ENVIRONMENTAL CHARACTERISTICS

Now let us take a look onto environmental characteristics.

Incidents are defined as unplanned interruptions or reductions in quality of IT services (service interruption). Somehow they are part of quality. However, the manager has to handle incidents whenever they occur. Incidents are always symptoms. The manager has to work on the causes. Sometimes a cause of an incident may lie outside the management object. In case of an IT application system the incident may be that a user complains about missing functionality or malfunction, but it finally turns out that he was not trained properly to work with this application…

Problems are the causes of incidents (see ITIL). Of course, they are performance characteristics if they can be allocated within the management object. But problems may also be allocated outside the management object. So it may turn out that we have performance problems on the one hand and environmental problems on the other side.

Change means, that the management object must be changed or modified due to requirements coming from outside. Requirements may come from outside business needs, but also from technical or technological development, also from the supplier side, e.g. through the closure of a supplier firm or a merger between different suppliers. Of course, there may be changes, which are initiated by the manager himself or by people working in the management object. But there will not be a target to have a specific number of changes.

Risks are occurrences of damage, injury, liability, loss, or any other negative impact that is caused by external or internal vulnerabilities, and that may be avoided through preemptive action. Similar to changes many risks have their cause outside the management object. But there will also be a lot of inner causes. However, risks cannot be planned. And we will see for any target, that there are more or less risks to reach the target.

Size and complexity are properties of the management object. In an output driven business they are mainly determined by the output, which we expect for the management object. The manager is not free to reduce size or complexity because he or she has to be prepared for a specific and expected output volume. But he or she has to consider it to do his or her management job properly. Thus we have to measure size and complexity, which may be a difficult and cumbersome task. And we have to measure it explicitly because there will be non-linear relationships between output and size or complexity.

We think, that those categories of environmental characteristics cover the most situations in the IT environment. However, the reader may in his practice be compelled with additional parameters, e.g. if some IT system is used outdoor, real weather data may be an important information for the responsible manager.

Each manager will have targets for the way, in which he or she manages incidents, problems, changes, and risks, related to his or her management object. Those targets are an integral part of incident management, problem management, change management, and risk management. However, the target will (and should) never be, to increase or decrease the number of incidents, but to handle incoming incidents in an effective and efficient way.

2.8 CHALLENGES OF MEASUREMENT AND MONITORING

Finally let us investigate some indicators and find out, what we must do to determine their actual and planned or forecasted values.

2.8.1 OUTPUT QUANTITIES

The simplest task seems to be the determination of produced or delivered output quantities and the number of provided or consumed input units. However, there always must be a **counting mechanism** and a **documentation of results**. If you want to count the number of transactions of an application system, then you must be able to count them technically.

Counting **transactions**, for example, must be part of the functionality of the system. But was this taken into consideration when the software was designed? Is this functionality provided by your purchased software package? If we are not able or not willing to count transactions, what could be an alternative approach to measure the output of an application system? Example 1: The number of data records having been processed. Example 2: The number of users of the considered system.

But even the latter is not as easy as it seems to be. If we consider an application system, which is only used within an organization, then it does not make much sense to count the number of users registered in the central user directory. It would be more helpful for management to count those users, which have really used the system. Thus we have to make sure that we register whether a user starts and uses this application. But at this point further questions arise. Is every visit of a user one separate output unit independently from time and duration? And is each **visit** independent from what the user really does with the system?

We cannot give a universal answer. But all those questions must be clearly answered in a specific environment.

Those questions must be answered similarly if we consider an application, which is also open to external users, e.g. an online shop. Here we may be able to register visits completely but we may not be able to clearly identify the users behind them. But the situation is quite different if we compare 100 visits of 10 users with 10 visits of 100 users. So the challenge in output measurement is to find a unit, which can be measured and the results can be stored for reporting purposes, and which at the same time makes sense in an economic or managerial perspective.

So the main challenge in **counting output units** is to take an output unit, which makes sense, and which at the same time can be counted and the results be stored for reporting purposes.

If we consider outputs, another problem comes up. Let us again consider an application system, providing *N* different user functions. Does it make sense to register hundreds of different output quantities, which will be easily found in big software packages? Can we define one global output unit somehow aggregating all the *N* user functions? Or can we reduce this task to the consideration of only one or few selected user functions?

To answer the first question we can consider all the *N* different **functions**, compare them, and build weights $w_i \geq 0$ for all functions. Then the output of this application system can be described by the weighted average value

$$S = \sum_{i=1}^{N} w_i \times f_i$$

with the weights w_i and the f_i are the numbers of the calls of the function *i*. The problem here is to determine the **weights** (however, we have good methods to do this, see chapter 9). To answer the second question our problem will be to select the "right" functions.

On a higher level we have a similar problem, namely, if we want to express the total output of our management object with one figure. Again we can do this by building a weighted sum of our different output quantities. If we would sell our products to an external market, the weights could be our **selling prices** and the weighted sum would be the total sales of our organization (neglecting the problem of different prices for different customers, rebates etc.). But if we do not deliver to external customers we will not have prices for our products. We have methods to define the financial value of output, but to investigate these methods is beyond the scope of this book.

However, there is a mathematical problem, too. If we decide to express our total output be the weighted sum

$$S = \sum_{i=1}^{n} w_i \times s_i$$

with the weights $w_i \geq 0$ and the output quantities $s_i \geq 0$, then we have to accept, that the same output value S_0 can be realized by any output vector $\underline{s} = (s_1, s_2, \ldots, s_n)$ having

$$\sum_{i=1}^{n} w_i \times s_i = S_0$$

All these output combinations lie on a (n-1)-dimensional **hyper-plane** in our n-dimensional output space. The performance manager has to find out, whether such a formula gives appropriate information to the responsible manager.

2.8.2 INPUT QUANTITIES

Similar considerations can be conducted for the input. Here the situation is easier for us, and we will be able to find prices for every needed resource, because resources finally have to be purchased from external markets. Weighting the consumed resource quantities with purchasing prices leads to **costs**, which we will consider in more detail in chapter 3.

2.8.3 AVAILABILITY

Let us now proceed to the **measurement of availability**. First remember, how availability was defined in general. A management object is available, if an **output request** can be taken from the requestor. This sounds simple, but in detail it is a quite difficult issue. How does the manager know, that a specific request was accepted for further handling? Hence we have

to find a technical or organizational way to determine this for every specific request. In case of an application system we are finally in a situation to find this out for every transaction, which a user tries to start. Here again we have the problem, whether we are able to get the **availability signal** for every single transaction. If we are not able to proceed in this way, which alternative approaches do we have?

One approach could be to assume that the system, which is under consideration, is available as long as nobody complains about **non-availability**. If this complainant can tell us, how long the system was not available, then we can proceed as follows. If the service window is f and the duration of non-availability is b (like breakdown), then availability could be measured as the ratio $\frac{f-b}{f} \times 100$ [%]. If we take this measurement method, then we would have 100% availability, if no user complains non-availability.

Another approach would be to send small requests ("ping") to the system in regular and short distances, e.g. every 5 or 10 seconds. Then we count the number of those requests, which have been confirmed by the system. If we send N requests within the service window and if C requests are confirmed by the system, then the availability could be expressed by the ratio $\frac{C}{N} \times 100$ [%]. Choosing this approach assumes, that the system normally will be available all the time, if it is available now and some seconds later, too.

Finally we have to consider, that many management objects do have more than one **service point** (like a super market, which has several cashiers). If we are able to measure the availability of every service point (of course, with the same method) what then is the **overall availability** of the system or management object?

Most people tend to say, that if we have m service points and N_i, $1 \le i \le m$, are the numbers of the availability checking requests and C_i, $1 \le i \le m$, are the corresponding **confirmed requests**, then the overall availability is

$$A = \frac{\sum_{i=1}^{m} C_i}{\sum_{i=1}^{m} N_i}$$

This is a weighted mean value of the local availability at the included service points, namely

$$A = \sum_{i=1}^{m} w_i \times a_i \text{ with } w_i = \frac{N_i}{\sum_{j=1}^{m} N_j} \text{ and } a_i = \frac{C_i}{N_i}.$$

Now let us ask, whether this is a "good" method to express overall availability (under the assumption, that the most important target of performance management is transparency).

To demonstrate the problem, let us consider the following **example**: Let the target availability be 95% and let the system have 10 service points, each of which gets 1.000 availability checking requests. Due to our definition the target would have been reached if totally 9.500 requests have been confirmed by the system. This may result from 9 service points having an availability of 100% and one service point having an availability of 50%. It may also result from 8 service points having an availability of 94% with 2 service points having an availability of 99%. However, it could also result from all service points having a local availability of 95%.

But do we have an alternative? Yes, if we assume that we ask for an availability of 95% at every service point. Now we should measure the portion of service points having reached the requested local availability. Looking back to our example the first case would lead to a 90% availability, the second case to a 20% availability, and in the last case we would have a 100% availability. The manager would benefit significantly from this approach, because now he gets the information how many of the service points deliver a good resp. bad service, and he or she now has to look after those service points with low availability. The traditional weighted average approaches would give an **inappropriate signal** to the manager.

Let us now consider a single service point. Let the reporting period there be a month. If we have an availability of 95% reached at the end of the month, then this could result from 10 working days with 90% availability and 10 working days with 100% availability. This leads us to the recommendation, not to consider the average availability per month, but the average availability per day and aggregate it to a metric, which shows the portion of days where we have reached the target availability. So we have to consider, how often we should set our meter back to zero.

Let us shortly get back to the situation with various service points. The approach, which we have recommended here, allows define different target availabilities for different service points. In an IT service environment this may be reasonable, because we have to differentiate the availability target due to the specific (geographical) location of service points.

In total have we found out that the performance management has to think carefully about the **measurement atoms**. Availability per day and service point seems to be better than the availability per 1.000 service points and month. The level of transparency, which we can reach depends on the definition of our measurement atoms.

2.8.4 RESPONSE TIME

Our third candidate for a more detailed investigation of measurement and monitoring is the **response target**. What does a quick response mean precisely? At first sight it will mean that an activity should not take too much time. We are looking for the **duration** of an activity. This may be the duration of processing a change request or the duration of executing a specific transaction of an application system.

The target duration should be d_0 time units. If the transaction is executed n-times, we could measure the start time and the end time of every single execution. This leads to its duration of d_i time units. The duration measure then could be

$$d = \frac{1}{n}\sum_{i=1}^{n} d_i$$

If $d \leq d_0$ then we have reached our target, otherwise we have failed it.

Similarly to the availability measurement d has been defined as a mean value, so that we may have $d \geq d_0$ though a lot of executions have led to $d_i < d_0$. Consequently we recommend to use another approach. Therefore we define an auxiliary parameter b_i and define it as follows: $b_i = 1$, if $d_i \leq d_0$, $b_i = 0$ else. Then we define

$$d = \frac{1}{n}\sum_{i=1}^{n} b_i$$

That is the portion of executions, which have met the duration target. Again we could aggregate transactions with different duration targets, e.g. a process, which runs through different branches. Here we would not need different metrics for each branch.

The measurement of **execution times** seems to be simpler than measuring availability. However, it shows a general challenge, which performance measurement has to master. You can do this either automatically or manually. In the first case you need technical components, which can do this, or you have to write software for that purpose. In the second case you have to define workflows and people have to set the needed time stamps. For both, the automatic and the manual, of course also for hybrid approaches, you need some kind of **structure**, **formality**, and (yes!) **bureaucracy**.

In many cases the transaction or process under consideration is dedicated to the creation and modification of documents. Then those time stamps can be linked to the specific document and the document does not only have its content, but also carries performance management information. In real operations it sometimes takes a lot of time and effort to **make activities ready for time measurement**. You should never underestimate this.

The other characteristics lead to similar questions and challenges. We also have to count, e.g. the identified number of risks. And we should re-count periodically. We often try to value and differentiate between different categories of sizes of our objects, which we have to count. Sizing is always difficult. Be careful to invest too much effort here! It is better to have a rough metric, which is easy to understand (for management) than a very sophisticated metric, which causes much effort and is hard to understand.

2.8.5 USER SATISFACTION

To demonstrate the issue of measurement let us finally consider the **user satisfaction**. How can we measure this? Many organizations do it in this way: Once per year a questionnaire is prepared and sent out to the user community. Usually a quite small portion of the users gives a feedback by filling out this questionnaire (which can be done electronically), so that nothing must be sent back to the distributor of the questionnaire. The questions must be answered by a scoring via given values) often integers from 0 to 5, 0 being the worst and 5 being the best case). For ach question the answers of the different users are put together into the mean value of all given scores. Now these scores of the single questions are put together, by building a weighted sum of all scores. The result is the so-called **user satisfaction index**. If all questions can be scored from 0 to 5 and the sum of the non-negative weights is 1, then the final user satisfaction index also has a value between 0 and 5.

What are the disadvantages of this approach? First we cannot follow this approach on a day-to-day basis due to the relatively high effort. Secondly, the portion of answering users normally is very small, and there is the question, whether the result is a representative one. Or do mainly negatively thinking or mainly positively thinking persons give an answer? Is the average score really helpful information, or would it be more helpful to know the portions of very satisfied or very unsatisfied users?

However, we are interested in the perception of our services through the users. What can we do? If the user community is big enough, we can use **statistical approaches** and take **random samples** from our user community and let them answer to our questionnaire. This could be done on a monthly basis.

Alternatively we could find out the user satisfaction in the following way:

- Count the number of **user complaints**. If there are no complaints our users are obviously satisfied. But it this a permissible assumption?
- We could provide a **good-service ticket** and ask our users to send one if they feel, that we have delivered a good service. But will they send us a ticket even if the service was good?

- We know an organization, which had a single-sign-on for all their application systems. When the user started his local device in the morning, he or she was asked **one single question**: Have you been satisfied with our IT service within the last 24 hours? The only possible answers were "yes" and "no". The user had to give an answer if he or she wanted to continue. So this organization had a user feedback every day.

We see, that it sometimes needs some creativity to find effective ways of measuring effects.

However, there might be one big question, when asking for user satisfaction: Is it really measurement? Couldn´t it happen that users give a negative feedback even if the service was good? And couldn´t it happen that a user does not give the same answer at different points of time even if the service and the service quality had been identical? Yes, this could happen. But if the user feels that our service was not good (enough), then our service was not really good. And if the **expectations** or the **perceptions** of the user change we have to change our service accordingly. The judgment of the user is subjective, of course. But his or her judgment is an important fact for us.

Finally every **evaluation** and every **examination** is based on people´s judgment. Every decision is based on personal judgment. If we can document this then it becomes measurable. In complex environments the same measurement if repeated may lead to different results. If those differences are small, then we can interpret this as some kind of white noise. If differences are big this may turn out to be valuable information for the responsible manager.

2.8.6 PERIODICAL MEASUREMENT

This observation leads us to the final remark that measurements are not one-time activities. When used in control cycles they have to be conducted **periodically** according to the reporting periods.

2.9 CHARACTERISTICS AND STRATEGIC MANAGEMENT

Up to now we have discussed more or less metrics or indicators to be a tool in management activities oriented towards control cycles. We have to ask critically, whether this is an exhausting point of view. Do KPI´s (Key Performance Indicators) only make sense with respect to control cycles? And what is the problem?

The problem is that thinking in control cycles leads to a thinking in planning and reporting periods. The time horizon is quite short, normally one year or less. Of course can we set strategic targets within these easy-to-grasp time-spans. But what about **strategic management**? And time horizons of 5 to 10 years? Formally we can apply our control cycle and measurement approach to strategic management. But is this helpful for the responsible management?

Perhaps we can proceed if we understand the **kernel of strategic management**. It basically aims at ensuring the existence of the organization under changing environmental conditions. Of course can we set targets for the organization in total, possibly in terms of size, profitability, market share, reputation, etc. But what can we say if we focus on an IT department? It seems probable and reasonable that within the next 5 years the activities of the IT organization will change dramatically due to high-speed technological development, due to unexpected or unforeseeable technological revolutions. What can we do?

We are not sure that the traditional PDCA approach will really help us, because planning needs a prerequisite – clear and precise targets, quantifiable and measurable. How can performance management contribute to this?

We think that on the strategic level IT (performance) management must be able to understand what is going on outside. Outside means: What is going on in the **organization´s business**? And what is going on in the **IT universe**? And they have to find out, how and how fast the own IT organization can adapt itself to these changes. Performance management can help to master this in

- building and running an **early-warning system**, which as early as possible can give messages to the IT management that something is changing inside or outside the organization. Earth quakes and volcanic eruptions have to be forecasted.
- analyze the past development of the own organization and how fast and how successful it **responded to changes** in the past.

Which metrics and indicators can help to find this out? To answer this question would break up the size of this book. It is one of the most difficult and one of the most fascinating areas of IT performance management. We cannot give a simple answer here. But the reader of this book should be or become sensitive to this issue…

2.10 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.2!

2.10.1 QUESTIONS FOR YOUR SELF-STUDY

Q2.1: Why should performance managers have an eye on capacities? What will be the impact of capacities being too small or capacities being too large?

Q2.2: What can we do to show the total input or output of an IT organization in one figure?

Q2.3: Why should the manager consider environmental characteristics systematically?

Q2.4: What are the most relevant examples of environmental indicators in the area of IT management? Which indicator would you like to add to the list, which is presented in this chapter?

Q2.5: Which categories of target achievement do you know? Which implications do they have for measurement?

2.10.2 PREPARATION FOR FINAL EXAMINATION

T2.1: What are the three elements of our quantitative model of IT performance?

T2.2: What does a distance function aim at, and how can we use it to build specific target level achievement functions? How can mathematical metrics help here?

T2.3: Please, describe the specific properties of performance and environmental indicators.

T2.4: Consider availability and reliability. How are they defined? What are the differences between both terms?

T2.5: What is the meaning of measurement for the management control cycle?

2.10.3 HOMEWORK

H2.1: Find out, what the term benchmarking means. Why is benchmarking an issue for IT performance managers? Which requirements for measurement must be fulfilled to be able to conduct benchmarking?

H2.2: To discuss indicators is strongly related to discussions about scorecards. Find out, what a scorecard is and how it is related to indicators.

H2.3: Conduct a research on indicators, which could be helpful for IT management. What are the precise definitions of the indicators, which you have found? What does this mean for measurement?

3 COST MANAGEMENT

Learning objectives

In this chapter you will learn,

- what costs are and what are the three dimensions of cost management,
- how IT organizations should structure their costs in the most appropriate way,
- what are the typical cost management challenges in IT,
- which targets we usually have due to cost management and whether cost cutting is always the right strategy,
- which methods are most important for IT cost managers.

Recommended pre-reading

- Bhimani 2015

In this chapter we will investigate cost management, which on the one hand is a traditional subject of IT performance management and on the other hand is, of course, very important for IT performance management, not least, because cost cutting or cost reduction are a basic element of target setting for the IT management in many organizations.

3.1 DEFINITION OF COST

What are costs? Let us start with the following definition:

> Costs are the financial value of consumed resources and services as they are needed to produce and to sell the own products or services or to maintain the operational readiness of the own organization. Costs are a term of management accounting.

Costs are measured in currency units. But not every amount of money is cost. What are the differences? In bookkeeping we discuss **expenses**, which reduce an organization´s profit. There are three major examples of the difference between costs and expenses:

- Book **depreciations** are expenses and they are not necessarily identical to calculative depreciations. Normally the amounts of the latter are lower than the first (per period).
- Shareholders provide capital to "their" firm and they expect to get interest from that. **Interest payments** to shareholders are not under consideration of bookkeeping, but they play a role in cost accounting.
- The same is true for **entrepreneurial profit**.

Also **payments** are measured in currency units. If we talk about a payment then there must be a cash flow, either as physical transfer of bank notes and coins or as a transfer of book-money from one bank account to another bank account. If we buy a machine, then there will be a (huge) payment for that machine. But this payment has nothing to do with costs.

The difference between **book depreciations** and **calculative depreciations** may play a role in the calculation of unit costs and (internal) service prices for IT services. The difference between payments and costs will be an essential issue in the preparation of business cases and the determination of **return on investments** (ROI). The reader should have this in mind. We will come back to this topic in chapter 6.

3.2 VIEWS ONTO COSTS

Now we have three major views onto costs:

- cost types (or: cost elements),
- cost centers,
- cost units.

3.2.1 COST TYPES

Let us consider a cost record. Assigning a cost type to this record means to answer the question, which resource (or supporting product or service) has been consumed.

Typical cost types are shown in figure 9 (see Bhimani 2015, pp. 123 – 151).

Cost type	Description
Labor costs	Subdivided into salaries, social charges, or fees for hired labor
Material costs	Subdivided into consumables, raw materials, and supplies
External labor costs	Subdivided into leasing fees, rental fees, maintenance fees, consulting costs, and telecommunication costs
Capital costs	Subdivided into depreciations and interests (for the usage of capital)
Costs of human society	Subdivided into taxes and insurance fees

Fig. 9: Typical cost types

Companies and other big organizations normally work with a given **table of accounts**, and thus IT performance management is not free to define its own cost types. IT performance managers often have a serious problem with this situation, because those table of accounts have been developed at times, when IT did not play any role in production and administration.

This has changed dramatically in the mean time, but usually there is no flexibility to adapt the given table of accounts. The IT performance management expert will not be allowed to define completely new cost types. His chance will be, that he is allowed to define cost sub-types, which help him to create optimal transparency of IT costs.

The identification of a cost type has one big prerequisite. We must know, which **resource** has been consumed. And to find out the amount of costs, or to find out, whether the costs are low or high, we have to know the **quantity** of the consumed resource. What does this mean? It simply means, that we, for the purposes of performance management, have to establish an **accounting of resources** and a **monitoring of resource consumption**. Otherwise we will not be able to build and run a really professional IT cost management.

3.2.2 COST CENTER

The second view onto costs is the cost center. Each cost record must be assigned to a cost center. The cost center gives an answer to the question, who is **responsible** for that cost record. This person has to plan the costs and has to explain it. This means, that every cost center must have a uniquely defined **cost center manager**.

Organizations are free to define cost centers, of course. However, there are typical characteristics of a cost center. It should have a center specific **planning**, and in most cases it has center specific **assets** and / or center specific (human) **resources**.

Each cost record should have a uniquely assigned cost type (only one) and a uniquely assigned cost center (only one). Then the total costs of an IT organization can be distributed into a two-dimensional table, where the rows represent the cost types, and the columns represent the cost centers. The total IT costs are the sum of all cost amounts in the cells of this two-dimensional table (see figure 10).

Cost types	Cost centers								
	1	2	3	4	...	...	...	...	nnn
1									
2									
...									
mmm									

Fig. 10: Cost center cost type table

In most organizations the management is relatively free to define and establish cost centres. So this can be used to evade the restrictions, which may be given for cost types. If you need specific cost sub-types, you can establish two cost centers A and B to differentiate a cost type X. Cost type X in cost center A has another meaning than cost type X in cost center B. Of course, this is a somehow dirty approach, but sometimes the IT performance management has to apply this technique to improve transparency of IT costs.

From a purely cost accounting point of view cost centers can be defined arbitrarily. However, we have made the experience that the boundaries of cost centers, which limit the area of responsibility of the cost center manager, should coincide with **organizational boundaries**. If you take any point in an organization, then there should be a 1:1 relationship between "real" management and responsibility for costs.

Finally we note that we have

- Primary-cost centers,
- Secondary-cost centers.

What does this mean?

Cost centers, of course, should not only consume resources and generate costs. These consumed resources should be used to produce (measurable) output. **Primary-cost centers** deliver products or services to other and outside organizations. **Secondary-cost centers** only deliver output to other cost centers in the own organization. For an IT department this has to be interpreted similarly. Its primary cost centers deliver IT services to other organizational units in the total firm or organization. Its secondary-cost centers deliver their services only to other parts of the IT organization. We will intensively deal with this differentiation when investigating unit costs and (internal) service charges of IT services.

3.2.3 COST UNITS

The third and final view onto costs is the view of cost units. Whereas cost types are focused on resource consumption and cost centers are focused on responsibility for costs, the cost unit has to answer the question, what the reason or objective of the costs was. Why do we have those costs? Do they make sense?

At this point we can come back to one of our starting points and basic terms in this book: the management object. Such a management object is defined by its owner and then given to a manager, so that he as an agent does a job, which the principal is not able or not willing to do. If and only if we can determine a management object to a cost record then this cost record is justifiable and makes sense for the principal, because this management object has to reach given targets. Therefor it consumes resources and generates costs.

Hence a cost unit is a **management object**, considered from the point of view of cost accounting (or resource consumption). For the various management objects in IT Management see figure 3.

However, there is a difference between cost types on the one hand and cost centers on the other hand. Whereas we have recommended that the assignment of cost types and cost centers should be unique, there may be a **multitude of cost units** assigned to the specific cost record. Example: A cost record may be relevant for the data center (cost center) but at the same time for IT security management (cost unit) and a specific software package (cost unit) (see figure 11).

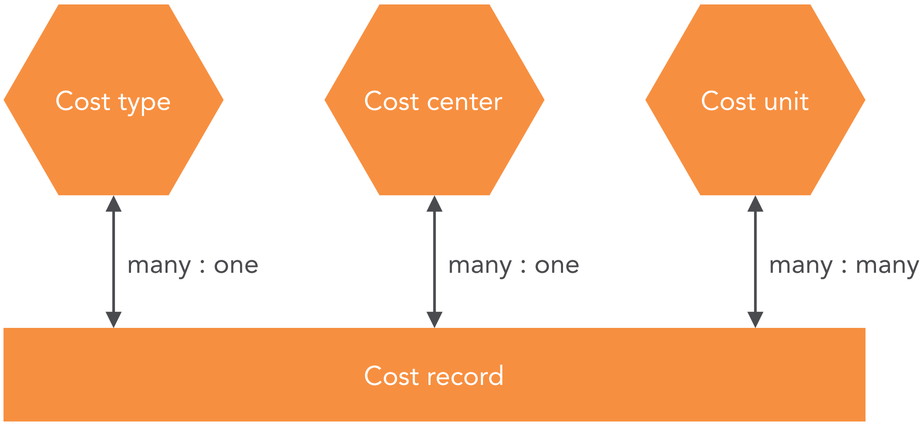


Fig. 11: Cost records and views onto costs

3.3 CHALLENGES FOR IT COSTS

Now that we have set up the cost ecosystem, consisting of cost types, cost centres and cost units, we have to investigate the major challenges for IT cost management.

3.3.1 FIXED AND VARIABLE COSTS

It seems that costs increase, when resource consumption goes up. And resource consumption will go up when the IT delivers more and more IT services to their organization. But will costs go down, when the resource consumption is reduced according to decreasing output quantities? This is not always the case, and hence cost accounting differentiates between variable costs and fixed costs. What does this mean? To understand it let us begin with a definition.

Variable costs increase / decrease automatically according to the increase / decrease of output. No management interaction is necessary.

In the opposite fixed costs do not change if output quantities go up or down. They only change after a specific management interaction.

If we have variable costs we normally assume that the dependency of these costs is **proportional**, which means that costs are doubled if output is doubled. But in IT performance management we should be careful. It may turn out that variable costs change in a non-linear way…

Obviously it would be fine for performance management if all costs would be variable costs. Then the management job would be a very easy job. Because the only thing management could and would have to change, would be the **production technology**. If this decision were made then costs would be a function of the output quantity, and if output would go down to zero also costs would become zero.

But the real world is different. What are the reasons? To understand the basic mechanism let us consider a person who is running a transportation business. This person will sooner or later buy a(nother) truck. Just having this truck will cause costs. If the truck can be fully loaded then the transportation will lead to a specific cost volume. If the truck can only partially be loaded costs will go down, e.g. gas consumption will be reduced. However, taxes and insurance fees will have to be paid whether the truck is working or not. Also the truck will loose value whether it is working or not. The reduction of value is measured as depreciation. The result will be, that the major part of the costs caused by that truck, do not depend on the usage of that truck.

Also in the IT business we have to consider fixed costs. To run our IT business we need some infrastructure. This **infrastructure** must be installed, and it must be ready to be used. And to build and to maintain this **readiness for usage** causes fixed costs. Holding up this readiness consumes a lot of resources. Nearly every thing in an IT organization is dedicated to be prepared for usage. So the portion of fixed costs will be high in IT organizations. And it will correlate to the issue of **capacity management**, which we will discuss in chapter 4. One of the biggest issues of IT performance management will be to find out, which part of the provided capacities is not used, and then to assign the costs of these idle capacities to the management objects resp. cost units.

Whether costs are fixed or variable depends on the **contract**, which we have entered with the supplier of the resource. An example are the salaries. On the one hand we can employ people and then pay them a fixed salary per month. Salaries and associated social charges are fixed costs. On the other hand we could hire freelancers and pay them on an hourly basis. So the costs of labor would become variable costs, because we would only pay for those hours, where the freelancers have really worked for our organization.

Which category of costs should we prefer? Management representatives often tend to prefer variable costs. But does this always make sense economically?

Within the mentioned contract we allocate **risks**. Making an employment contract shifts the **risk of idle times** to the employer. He has to pay a lump sum per month even if he is not able to let the employee work full time. Of course, he can terminate the employment contract, but he will because of corresponding contract regulations not be allowed to stop the payments immediately.

The situation is different if we enter a freelance contract. Then the employment risk is totally shifted to the employed person. And he or she will negotiate a premium to carry that employment risk. So the **unit costs** tend to be higher if we choose the variable payment.

Variable costs are not better per se. In the past years we have seen a big shift from variable costs to fixed costs in the telecommunication business. Only a few years ago we asked for and accepted a payment according to the time we used the telephone lines. We had to pay for every second. These were variable costs in the ideal term. We also had to pay a monthly fee for being allowed to use the telephone system and to get the necessary technical components provided. In the mean time we have completely shifted to **flat rates**. Paying a monthly fee gives us the opportunity to use the telephone system as long as we want or need to. Obviously this makes sense economically for all involved parties.

So the question of fixed versus variable costs cannot be answered in general. It is in the responsibility of IT performance management to find the best cost-strategy. And this has to do a lot with contract negotiation.

INNOVATIVE
LIKE YOU.

If you're hoping for a truly modern education, one where you're encouraged to speak your mind and to think long-term, both when it comes to your own future and the future of the planet. Then the University of Gothenburg is the place for you.

Study a Master's programme in Gothenburg, Sweden I www.gu.se/education

UNIVERSITY OF GOTHENBURG

Let us close this issue with an actual example. At the moment a lot of things are changing in the IT due to **cloud computing**. The propaganda of the cloud service providers actually is "pay as you use". We are sure that this will change over time. It will take not much time until they will ask for flat rates also here.

IT performance management has to care for fixed costs. Behind the fixed cost issue there is always the challenge of **capacity utilization**. IT performance management has to be aware of **idle-capacity costs** and be prepared for reducing them.

3.3.2 DIRECT COSTS AND INDIRECT COSTS

One big issue in IT cost management is the fixed cost issue. But the (IT) world is a complex world, and thus IT performance management has to consider a second big issue, the issue of indirect costs. Let us, as always, start with a definition.

Direct costs can be directly (in a 1:1 manner) assigned to an accounting resp. management object.

Indirect costs can only be assigned to a group of accounting resp. management objects. Sometimes indirect costs are called overhead costs.

However, overhead costs usually will be indirect costs, but there will be indirect costs, which are not at all overhead costs.

First let us make clear, what it means, whether we have direct costs or indirect costs, in terms of cost accounting. As we have mentioned in the definition the **assignment to a management object** is essential for the respective categorization.

So the terms of direct or indirect costs are **related to cost units** primarily. You will also have to investigate direct / indirect costs related to cost centers. But the natural relationship is with cost units resp. management objects.

If we want to discuss direct costs / indirect costs then we first have to identify the management object, whose costs we are going to investigate. If we can assign a cost record uniquely to a cost unit, then we have direct costs with respect to that specific cost unit. If we consider a **hierarchy of cost units** then we will come to the conclusion that a cost record will be direct costs on a specific level in that hierarchy. For cost units on lower hierarchy levels the same cost record will be indirect costs, for cost units in higher hierarchy levels it will, of course, also be direct costs. Thus we cannot not discuss about direct / indirect costs absolutely. It is always a **relative term**, and we have to denominate the cost unit, to which this cost record is assigned.

But what is the problem with direct costs now? If we consider a specific cost unit and a corresponding cost record then there is no problem in allocating costs if that cost record is categorized as direct costs. But there will be a problem for cost accounting if that cost record is categorized as indirect costs. Now we have to find an answer for the question, which portion of that cost record has to be or should be assigned to the considered cost unit.

To give a very clear answer in advance: We cannot say, which portion of the cost record has been exactly caused by the cost unit under consideration, if the cost record is indirect costs for that cost unit. We have to set a **distribution rule** or algorithm to assign that cost record to the concerned cost units. The best result will be, if everybody involved says, that he / she doesn´t know a better way of distributing the costs. And the challenge for IT performance management is to find such "best" distribution rules.

Example: Let us consider an IT department. The costs of the IT manager and his secretary are direct costs with respect to the total IT department. They are indirect costs for all management object resp. cost units within this IT organization. How should we distribute these costs internally – if this is necessary? One approach is, that people mainly determine management efforts. So management costs could be distributed within the IT organization according to the distribution of manpower. Alternatively one could come to the conclusion, e.g. in an environment with a high volume of outsourcing, that people are not so important, and distribute management costs according to operational costs (budget volumes).

We cannot decide in general, which method fits best. This may be different according to different organizations.

Actual IT organizations have a **high portion of indirect costs**. Traditional examples of indirect costs are **infrastructure costs**, e.g. network or data center. During the recent years the volume of indirect costs has increased due to the exchange of departmental applications as we had them even in the 1980´s by so-called **corporate applications**, e.g. ERP systems. Also the **convergence of technologies** (e.g. telecommunications) leads to an increasing volume of indirect costs.

We will have to come back to the issue of indirect costs when discussing unit costs and charges for IT services in chapter 5. The reader should be aware that the categories of fixed / variables costs and direct / indirect costs overlap (see figure 12).

	Fixed costs	Variable costs
Direct costs	*License fees for an application software*	*Paper consumption for printing*
Indirect costs	*Management salaries*	*Consumption of electric power*

Fig. 12: Overlapping Cost Categories (examples)

3.3.3 PRIMARY AND SECONDARY COSTS

IT performance management has to pay attention to a third pair of cost categories, the differentiation between primary costs and secondary costs. This categorization makes sense on the **background of cost centers**.

A cost record is denominated as primary costs, if the cost center manager is responsible for planning and managing the resources behind that cost record, and the resources are purchased from outside the organization on external markets.

A cost record is denominated as secondary costs, if another cost centre is responsible for planning and managing the resources behind this cost record. These costs are indirect costs with respect to the considered cost center, and they are distributed among the other cost centers.

The concerned cost center manager is not allowed to refuse those costs. He has to accept them, and he is not able to plan them. From the point of view of an IT department secondary costs are more or less restricted to costs, which are charged to the cost centers of the IT organization from outside the IT organization, especially from non-IT cost centers. Examples are

- charges for office space,
- general management fees,
- apportionments of overhead costs.

3.4 METHODS OF COST ACCOUNTING

Having now investigated the basic views onto costs and the different manifestations of costs, mainly the very challenging fixed costs and indirect costs, we now want to have a look into the cost accounting toolbox. The most relevant methods of cost accounting for IT performance management are:

- standard costing,
- activity-based costing,
- cost-distribution sheet.

Historically those methods have been developed in the environment of industrial mass production. So we should be a little bit careful in applying those methods to the IT environment.

3.4.1 STANDARD COSTING

Standard costing is one of the most important tools of cost accounting and at the same time one of the most powerful tools of cost management. The method **combines costs and quantities**, and this combination improves transparency and helps to explain deviations between planned and actual costs.

Planned costs are those, which we have planned on the basis of planned (input) quantities (Cost planning without quantity planning does not make any sense).

Actual costs are those, which we actually identify. They are correlated to the actual (input) quantities or consumed resources.

Now the standard costing method introduces a third term – the target costs. **Target costs** are the costs, which we would have planned on the basis of the actual (input) quantities.

Example: Assume, that you have to buy paper for your printers, and you expect to need 10.000 packages with 500 sheets at a package price of 4,00 EUR. Planned costs would be 40.000 EUR. At the end of the year you find out that actual paper costs are 50.000 EUR, and actual quantity is 14.000 packages. Of course, actual costs are 25% higher than planned costs. This is a negative information. But what would you have planned, when you had foreseen the quantity of 14.000 packages, which is 40% more than the planned volume? Let us assume that a planned volume of 14.000 packages would have led to a price of 3,50 EUR per package. Then target costs are 49.000 EUR. So you have found out, that you can explain 49.000 EUR of the actual paper costs, and there is only the open question, why the actual paper costs exceed the target costs by 1.000 EUR. Something has happened during the planning period, because without any management action the actual paper cost would be 56.000 EUR. May be, that there has been some management activity.

Of course, performance management has to investigate not only the financial aspects, but also the changes of the real world. We also should ask, why the actual quantities have exceeded the planned quantities by 40%. Was this significant change completely surprising? Or did we make some mistakes in the planning process? Didn´t we anticipate business developments early enough?

Usually standard costing is done under the assumption of completely variable costs. But the method also can be applied to **fixed cost situations** and **mixed fixed and variable costs**.

Example: You plan 5.000 GB of storage space, and your supplier offers this volume for a fixed price of 24.000 EUR for the whole planning period. During the planning period it turns out, that you need another 1.000 GB. Because there is an urgent need the contract is made after a very short negotiation and you pay another 6.000 EUR. So the actual costs are 30.000 EUR finally. Of course, the increased costs can be explained with the higher need for storage space. But now it may turn out that negotiating about 6.000 GB in the beginning would have led to a price of 27.000 EUR. So storage costs are 3.000 EUR too high, compared to target costs. Again the question is, why we have not foreseen the additional need of 20% storage space compared to the assumed 5.000 GB. How can we improve our planning quality?

3.4.2 ACTIVITY BASED COSTING (ABC)

This method aims at a better investigation of processes by cost management. Starting point, however, is the problem of fixed costs caused by providing capacities, e.g. human resources. ABC (in German: Prozesskostenrechnung (PKR)) tries to resolve fixed costs by correlating them to processes or specific activities.

A central term in this method is the **cost driver**. That is the resource, which mainly determines the costs of an activity, or which is of relevance for the costs of our management object under consideration. This means that it could make sense to consider more than one cost driver simultaneously.

How does ABC work? Starting point is a resource, which is provided with a specific capacity. This provided capacity leads to fixed costs for your organization. E.g. consider labor capacity provided by your employees. This capacity has been provided to be able to execute a specific activity or process.

Now you find out the number of planned or actual process or activity executions and determine, how much of the capacity is used (on average) by one execution. Multiplying the capacity volume, which is needed for one process execution, with the number of the executions leads to the total capacity consumption and correlated **used-capacity costs**. The difference between total capacity costs and the used-capacity costs lead to the idle-capacity costs.

Now it may happen that you expect a specific volume of process executions and due to the technical bundling of capacity you have to install more capacity than you really need. This is the **planned idle capacity**. If there is a lesser utilization of capacity than you have planned it, then this portion of the idle capacity and the corresponding costs are called **residual capacity** resp. residual costs.

Finally you have to analyse what the reason for idle capacity has been, whether it had been avoidable, and who has to carry the idle capacity costs.

Example: To better understand the power, but also the restrictions of ABC, let us consider an organizational unit conducting three processes with the following cost characteristics:

- P1: 1.200 executions and variable unit costs of 20 c.u. (currency units),
- P2: 300 executions and variable unit costs of 50 c.u.,
- P3: 100 executions and variable unit costs of 60 c.u.

Let us further assume that the (only) cost driver are personnel costs and the total personnel costs of this organizational unit are 450.000 c.u. We further assume that the total personnel cost per working hour are 50 c.u. So our organizational unit has a capacity of 9.000 working hours. The total variable costs are $VC = 1.200 \times 20 + 300 \times 50 + 100 \times 60 = 45.000$ c.u.

Now "traditional" cost accounting would conduct an **overhead calculation**. With this approach the block of fixed costs is distributed among the three processes in the same portions as the percentage of the process specific variable costs related to the total variable costs. This would lead to the result, presented in figure 13 (see OH entries in figure 14).

Process	Total variable costs	Portion	Total fixed costs	Total process costs	Quantity	Process unit costs
P1	24.000	24/45	240.000	264.000	1.200	220
P2	15.000	15/45	150.000	165.000	300	550
P3	6.000	6/45	60.000	66.000	100	660
	45.000		**450.000**	**495.000**		

Fig. 13: Overhead calculation

To consider the ABC approach let us assume that we have found out that each process execution needs a specific amount of labor capacity:

- P1: 4 hours → 200 c.u.
- P2: 2 hours → 100 c.u.
- P3: 4 hours → 200 c.u.

Thus the totally used volume of working hours is H = 1.200 × 4 + 300 × 2 + 100 × 4 = 5.800, which means that 290.000 c.u. of personnel costs are needed for process execution. We have an idle capacity of 3.200 working hours and thus idle capacity costs of 160.000 c.u. The net unit costs of the process executions are:

- P1: 200 + 20 = 220 c.u.
- P2: 100 + 50 = 150 c.u.
- P3: 200 + 60 = 260 c.u.

But now we have a problem. How shall we distribute the **idle-capacity costs** among those three processes? After all the idle-capacity costs are more than 35% of the total personnel costs of the considered organizational unit. In a real situation the responsible manager must have very good reasons to explain and justify such huge amount of idle-capacity costs.

If we assume, that those costs of 160.000 c.u. are necessary overhead for those three processes, then performance management has to establish a rule to distribute those costs among the three processes. In terms of cost accounting we have indirect costs of 160.000 c.u. There are at least three different ways to distribute them:

- According to the portion of costs of the single process, related to the total process costs (version IC-1): Total process costs are TC = 220 × 1.200 + 150 × 300 + 260 × 100 = 335.000 c.u. Process costs of P1 / P2 / P3 are 264.000 c.u. / 45.000 c.u. / 26.000 c.u., which means a portion of 78,8% / 13,4% / 7,8% (approximately). This means that the considered process has to carry additional 126.090 c.u. / 21.490 c.u. / 12.420 c.u. Finally the total process costs including idle capacity costs are 390.090 c.u. / 66.490 c.u. / 38.420 c.u., and this finally leads to process unit costs of 325 c.u. / 222 c.u. / 384 c.u.
- According to the portion of executions of the single process, related to the total number of 1.600 process executions (version IC-2): This leads to idle capacity costs of 120.000 c.u. / 30.000 c.u. / 10.000 c.u. Finally the total process costs including idle capacity costs are 384.000 c.u. / 75.000 c.u. / 36.000 c.u. and this leads to process unit costs of 320 c.u. / 250 c.u. / 360 c.u.
- Every process has to carry one third of the total overhead costs (version IC-3): This leads to process costs of 317.333 c.u. / 98.333 c.u. / 79.333 c.u. and process unit costs of 264 c.u. / 328 c.u. / 793 c.u.

Hence we have four different distributions of idle-capacity costs (see figure 14).

	Methods of distribution of idle-capacity costs			
Process	**OH**	**IC-1**	**IC-2**	**IC-3**
P1	220	325	320	264
P2	550	222	250	328
P3	660	384	360	793

Fig. 14: Distribution of idle-capacity costs

The final question is: Which method is the right one? Which method is the best one? The overhead method (OH) seems to be the worst method, because the other three approaches, based on ABC, seem to be more reasonable. This is true, of course. However, we will only be able to use ABC, if we know the **capacity usage of each process execution**. This is the basic prerequisite to apply ABC. If we are not able to deliver this information, then we cannot apply ABC. Better methods need more data. That is the price, which we have to pay for the more reasonable results.

If we would think that our three processes had two different cost drivers then we had also to provide the capacity consumption quantities with respect to the second cost driver.

3.4.3 COST DISTRIBUTION SHEET

The third cost accounting method, which we will need, is a tool to calculate unit costs for IT services. We will present and discuss the **cost distribution sheet** (in German: Betriebsabrechnungsbogen) in chapter 5.4 of this book.

3.5 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.3!

3.5.1 QUESTIONS FOR YOUR SELF-STUDY

Q3.1: In many situations the occurrence of fixed costs and variable costs depends on the nature of contracts between your organization as a customer and your supplier being a service provider for your organization. What are the pro´s and con´s of fixed costs (and corresponding payment rules in the contracts) for you, for the service provider? What could be the reasons for different points of view?

Q3.2: Make a research on the term “standard costing”. Where are your results similar to the statements in this book? Where are the differences?

Q3.3: Denominate specific areas in the IT organization where an application of ABC costing makes sense.

Q3.4: Find examples for cost center structures in IT organizations. Do you see improvement or optimization potential?

Q3.5: List typical cost types, which are relevant for IT organizations.

3.5.2 PREPARATION FOR FINAL EXAMINATION

T3.1: Determine, whether the following terms represent cost types, cost centers or cost units: rental fees for software, hourly rates for external experts, data center, workplace service, security management, SAP competence center.

T3.2: You plan to cooperate with external consultants at a volume of 1.000 person-days. The expected average daily rate will be 800 EUR. At the end of the year you see that your budget for external consulting has an overrun of 50%. You find out that daily rates for consultants have increased by 25% and that you have needed 200 person-days more than planned. Make a judgment of this situation. Which cost accounting method do you apply to run your analysis?

T3.3: Due to your planning you will need 3.000 GB of storage. However, the offering of your supplier is 2.000 GB. The annual rental fee for a storage unit will be 180.000 EUR. What will be the unit costs for 1 GB of storage per month?

T3.4: If you want to consider the costs of a specific business process, which view onto costs should you have here? In your organization there is a team of 5 employees working in several business processes. Describe a method to assign the personnel costs to the business processes.

T3.5: Consider an IT organization as a service provider. Explain the terms “primary-cost center” and “secondary-cost center” with respect to service delivery.

3.5.3 HOMEWORK

H3.1: Make a research for charters of accounts. Do you find specific charters for IT organizations? What do you find? Is it appropriate for your IT organization? What is your opinion?

H3.2: Is there a relationship between cost categories (fix / variable, direct / indirect) and the efficiency of an IT organization?

H3.3: It is often stated, that IT costs are too high. Is this a reasonable statement? How should IT management / IT performance management handle this criticism?

4 RESOURCE AND CAPACITY MANAGEMENT

Learning objectives

In this chapter you will learn,

- what are the most important resources of IT organizations,
- how to manage internal and external resources appropriately,
- how to plan resources and capacities,
- how to monitor capacity utilization,
- how to conduct make-or-buy decisions
- how to manage external suppliers.

Recommended pre-reading

- Damij 2014, pp. 77 – 82

In this chapter we will investigate the second pillar of input management in the IT, namely the management of resources and capacities. In chapter 3 we discussed the financial value of the consumed input, now we are going to consider the "real" and technical quantities.

First let us discuss, what has to be considered in detail.

4.1 GOODS AND SERVICES

We buy and provide goods and services from **external partners** to be used in our value creating processes. Those quantities can be used for one time. After they have be used or consumed the quantity resp. the included units of that resource or that service cannot be used for a second time.

If we need resources in the form of **physical goods**, we can buy specific quantities, can store them physically in a warehouse for some time and take it from the warehouse, when we need a unit for our IT production.

This is quite different for services. **A service unit cannot be stored**. It must be exactly provided in that second, when it is needed in our production. So we cannot buy a bundle of service units and store it in our warehouse. We only can make a contract with our service provider to have the right to get a service unit delivered, whenever we need it.

4.2 CAPACITY MANAGEMENT

Capacity is defined as the total amount, that can be contained or produced, or the ability to do a particular thing. So the term "capacity" is related to limitations or restrictions. IT management has to ensure, that **enough capacity** is provided. If there is insufficient capacity, then IT will not be able to fulfill customer´s requirements. And if there is too much capacity provided, then a part of this capacity will be unused. In either case the provision of capacity will be strongly related to fixed costs. Hence performance management should consider capacity management with priority.

4.2.1 STATIC CAPACITIES

We buy or rent static capacity from an external provider. Static means, that we can use the units of that capacity by putting something there or connecting something to these units permanently. If this is done, then the **occupied unit** cannot be used in another way. If we take the connected subject away, then the occupied unit becomes free and can be used for other purposes.

A typical example in the IT is storage capacity. If we have 1.000 GB storage space installed, then we can store files and data on it. If all the storage space is occupied with files, then no storage space resp. capacity is available.

Static capacity always is defined as some kind of **space**.

4.2.2 DYNAMIC CAPACITIES

With the term of dynamic capacity we denominate **throughput**. This finally means, that it combines static capacity with time.

To understand it better let us consider a parking lot for cars with 100 parking places. Then this parking place has a static capacity of 100 cars. If we know or assume, that a parking car leaves its parking place after 2 hours (on average), then the dynamic capacity of that parking lot would be 1.200 cars per day or 50 cars per hour.

Usually **static and dynamic capacity are combined** and cannot fully be separated. If a specific throughput is limited then we have either to reduce the length of the stay of the thing, which needs the space, or we have to increase the static capacity.

The challenge for IT performance management is that in a request driven business as we have to manage it in the IT, we cannot assume, that our "orders" come in constantly and evenly distributed. And we also cannot assume that each order needs the same space for the same time.

If we have a lot of static capacity installed and only a few orders come in, distributed in time, then each order has enough space available. We also may find out that a lot of space is never used. We have too much static capacity installed.

On the other hand, if the number of orders increases and their space need or occupation time goes up then we will experience that we do not have enough capacity to fulfil orders in time. We will have **waiting queues** or requesting customers will cancel order requests.

The problem and challenge in the IT environment is that we will have both (extreme) situations within short time intervals.

Let us consider a local area network (LAN). For a long time every day the network will be more or less idle. But vice versa there will be specific busy hours, where an extreme volume of requests for network usage comes up. It may also happen, that we have a bottleneck only in a small area of our network and the other parts are more or less underused.

4.3 PLANNING

Now the question is what IT performance management has to do to properly manage resources and capacity. It all starts with **planning** (as always). And if we have to plan the input part of a business, we first have to plan the output part of that business.

For an IT organization this means that we have to base our **input planning** on a sound **output planning**. If we accept the service-oriented thinking of ITIL, then we need a plan, which documents, which services are needed in which quantities at which time intervals. Then we should know the production function for every IT service (see chapter 5), and after applying those functions to the planned output quantities, we would know, which input is needed, and how much, and when. This sounds easy, but it isn´t, of course.

First of all we have to consider the on-going operation (**Run the business**) as well as the changes through projects (**Change the business**)! The latter will also need resources and capacities.

It is not enough to determine the quantities and their distribution in time. We also have to provide the needed quantities in time.

Let us now consider specific technical equipment, which provides some space (static capacity). If all units of that space are occupied by orders, which are currently processed, then no further order can be processed. Assume that for the time being we are not able to reduce the (average) occupation time of the actually processed orders. Then we have to add further space (static capacity). This will take some time due to **procurement** and **installation** activities. After some time the added capacity volume will increase, and the waiting orders can enter the processing. If the order volume decreases then sooner or later we will find out that capacity units are idle, and then we can decide to cut back capacity.

In the era of cloud computing it seems reasonable to get needed static capacity in arbitrary quantities and immediately, if needed. Then we do not need any capacity management, because capacities can be adapted to actual needs without any (significant) time delay. From a technical point of view. But now we have to **couple technical capacity management with cost accounting**.

It may be economically more efficient to have a fixed static capacity installed, even if it is not always completely used. It may be that our organization has to pay less for a fixed capacity bundle, which then has a specific percentage of **idle capacity**, than to get only and exactly that capacity, which is actually needed. For the first alternative we talk about fixed costs, for the second alternative we talk about variable costs (see chapter 3).

Also the time, which is needed to buy and to install additional static capacity, may be too long from a business point of view, and so it could also make sense economically to have some **overcapacity** installed.

On the other hand we have to discuss, whether it makes sense economically to work with limited (static) capacity. If all installed capacity units are occupied then orders have to wait until capacity units are freed from preceding orders. Of course, time is money, and hence there will be some negative effects. But it may turn out that service levels (duration times) decreasing do not hurt the business essentially and can be tolerated by the business within some threshold values.

However, it depends on the specific situation. There may be **business critical functions**, which do not allow any delay. Then due to the dynamic capacity of the system the static capacity must be chosen so that no order has to wait. Static capacity must be provided for **peak situations**, for the times, when a maximum volume of orders has to be processed.

So the **capacity issue** is strongly correlated with the **fixed cost issue**. If we provided capacity (and we cannot get it from third parties with that flexibility as our business needs it), then we have (a lot of) fixed costs for installing and providing capacity (which may not be fully used and / or all the time). We also have to accept that capacity usually is **bundled** and we cannot buy or rent just the optimal capacity. We have to buy or rent more according to the packages, which are offered by our external producers or providers.

On the other hand let us consider the **resources**, which we need in the form of **goods** or **services**. Theoretically we can buy any quantity at any time. But again we have to learn that ordering, delivering, and warehousing takes some time and this time may fluctuate. So we need some **buffers**. Also our suppliers may **bundle the resources** so that we are not able to buy specific quantities of a specific good but have to buy given quantities as offered by the supplier. Are we always able to use the complete ordered quantity in our production processes afterwards?

A similar situation may occur in the provision of **services**. Of course, each service unit has just to be provided in the moment, when it is really needed. But we have to make a contract with the service provider. And it may occur that he sells his services (or the right to ask for a service unit) in bundles also. An example is the procurement of software licenses. It may be that we need 47 licenses, but the software company only provides a package with 50.

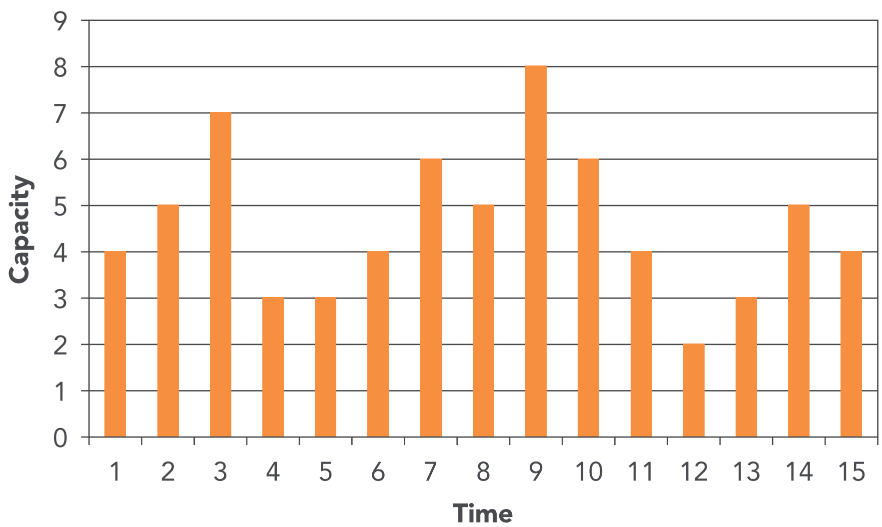


Fig. 15: Requirements for capacity over time

In figure 15 we see a simple example of the changing capacity requirements over time. The highest static capacity is 8. If unit costs are 10 c.u. per time unit then we have to pay 1.200 c.u. for the capacity 8 over 15 time units. Now we can reduce the installed capacity and buy the additionally needed capacity on the market. We assume that an external capacity unit costs 16 c.u. per time unit. This leads to the results presented in figure 16.

Installed capacity	Costs for installed capacity	Costs for extra capacity	Total costs
8	8 × 10 × 15 = 1.200	0 × 16 = 0	1.200
7	7 × 10 × 15 = 1.050	1 × 16 = 16	1.066
6	6 × 10 × 15 = 900	3 × 16 = 48	948
5	5 × 10 × 15 = 750	7 × 16 = 112	862
4	4 × 10 × 15 = 600	14 × 16 = 224	824
3	3 × 10 × 15 = 450	25 × 16 = 400	850
2	2 × 10 × 15 = 300	39 × 16 = 624	924
1	1 × 10 × 15 = 150	54 × 16 = 864	1.015
0	0 × 10 × 15 = 0	69 × 16 = 1.104	1.104

Fig. 16: Costs for installed and extra capacity

STUDY FOR YOUR MASTER'S DEGREE
IN THE CRADLE OF SWEDISH ENGINEERING

Chalmers University of Technology conducts research and education in engineering and natural sciences, architecture, technology-related mathematical sciences and nautical sciences. Behind all that Chalmers accomplishes, the aim persists for contributing to a sustainable future – both nationally and globally.

Visit us on **Chalmers.se** or **Next Stop Chalmers** on facebook.

So, if we could provide external space immediately, when the need comes up, then it would be the cost minimal solution to install a capacity of 4 and buy the rest on demand from an external supplier.

However, if you have to order this external space two time units, before the need occurs, then you have to make **forecasts**, and those forecasts may be wrong. Our forecasts will be based on actual data. If the changes in the utilization of the static capacity follow a specific pattern then we can make good forecasts. If the capacity-need changes by chance the risk of making wrong forecasts will be significantly high.

4.4 MONITORING

An essential component of our resource and capacity management is monitoring. If we stay to the model of a space, which is temporarily occupied by orders, then monitoring has to deliver the following information:

- for any point of time the actually installed static capacity,
- for any point of time the actually occupied space (which in combination with the totally installed space gives an information about the idle or unused space),
- for every order, which occupies space, the point of time, when the usage of space starts, and the point of time, when the usage ends (which gives the duration of the space usage by this specific order),
- for every order the space, which has been occupied by this order.

These demands for measurement are easy to put, but the implementation can be quite difficult and complicated.

First we have to find out and to decide, how often we would like to measure **space utilization**. Every second, minute, hour, day? And we have to consider the fact that a measurement might have an **impact on the availability** of that capacity (by blocking it during the measurement for other usage).

Secondly, we have to take the two time stamps for every order and at the same time to measure the amount of space occupation of each specific order. Are we technically able to do this? If we consider a storage system, the internal logging of the system will give us this information. But what about a system, where we are not so clear, what the occupied space really is.

Let us for example consider a quite complex system, a service desk. The capacity of this service desk is the throughput of incidents and service requests, summarized by **tickets**. If performance management finds out that this service desk is able to resolve 100 tickets per day and that the average duration of a ticket process is 10 days, then the static capacity of this service desk is 1.000 tickets in 10 days. But we will not be able to measure this directly.

So, in many cases, especially in cases of (very) complex systems or organizations we know that theoretically there is a static capacity behind the system, but we are not able to identify a technical component as the carrier of that static capacity. So we are not able to measure the capacity utilization of the system or organization, and we also may not be able to measure the capacity consumption of a specific order.

What can we do here? There is a general strategy in (IT) performance management. If we are not able to measure directly we should try to **measure indirectly**.

We can come to the conclusion, that our system has enough capacity, if all orders / requests are processed within the expected time interval. If the (virtual) static capacity is completely occupied then the "next" order has to wait, until the capacity, which this order needs, is freed.

So, if the processing times of orders increase, then this might indicate that there is not enough capacity. If more and more requests are sent to the system a **waiting queue** will result. Orders have to wait for some time until the system´s capacity is freed to accept the next order. In technical systems (but also in organizational systems) it may happen that the system runs into a **deadlock situation**. That means that no order can be processed. Even the orders, which have already entered the system, do not proceed.

On the other hand one of the major challenges of IT performance management is to identify and quantify **idle capacity**. Sometimes this capacity is not idle in a technical sense. The capacity is used to run activities, but it may turn out in a deeper look that those activities do not contribute to the business. Especially human capacity is at risk, to be busy but do not really contribute to the target achievement of the organization.

4.4.1 ZERO-BASE BUDGETING

A method, which is dedicated to this problem, is the **zero-base budgeting method**. This method has been developed to analyze those organizational areas, which are often described by the term administration or cross-section activities. In terms of process management the so-called **supporting processes** are considered here.

The zero-base budgeting comes from the financial budget planning, and its basic question always is: Do we really need this activity? What will be the impact for our business, if we completely eliminate this activity? If we really need it, how much can we reduce it in quantity or in quality without any negative impact for our business? (see Pyhrr 1977). The objective of zero-base budgeting is the **lean organization**.

4.5 HUMAN RESOURCE MANAGEMENT

One of our most relevant resources is **human resources**, because people make the business. This is especially true for IT, because, though IT more and more being automated, it finally is people business. However, the human resource is a challenging resource. People do not work like machines, their productivity is going up and down over time, even during one day. The **man-day**, a unit of human work, is a fuzzy unit. You will realize it when you try to measure the output of programmers. The result is not only a matter of quantity. It is as well a matter of quality, a matter of problem solving capability, a matter of creativity. This is true not only for programmers but similarly for all people working in an IT organization. Management of human resources includes motivation because motivated people will produce more and better output. It also includes training and coaching, because skilled people will be more productive and deliver better quality.

On the other hand people are a wonderful resource because of their **extreme flexibility** and their **will to contribute**. They will work over time, more than is allowed by law, and they will do things, which they have never done before, because they are members of a team, and they want to make this team successful. You will see this in your next project, where you are responsible for project management or project performance management.

We should, from a performance management point of view, consider our employees as **valuable machines**. It is expensive to get a new employee hired or to prepare a person for a new task. And also the running costs of human beings are high (salaries). It seems to be inhuman to consider employees as machines, but we are convinced that this view is much more appropriate than to consider human labour like some kind of bulk material. A machine needs an **investment**, it generates **operating costs**, it must be **maintained** and **reviewed**, and sometime it has to be **updated** and **modernized**. Obviously we have to fulfill similar tasks when managing employees.

Of course, the reader may ask: Why should I have my own machines. Couldn´t I rent them on an hourly basis and work with freelancers or temporary workers. As performance management experts we should consider this very carefully. From a purely economical point of view the question can be answered with capacity considerations.

Let capacity needs over time be represented by *f(t)*. Then the total capacity need is $\int_a^b f(t)dt$. We can have own capacity on level *c*. So we provide $\int_a^b cdt$. Now we need temporary additional capacity, represented by $\max(0; f(t) - c)$. If labor price is p_i for own employees and p_e for temporary workers, whereas usually $p_e > p_i$ can be assumed, then total labor cost will be

$$L = p_i \times \int_a^b cdt + p_e \times \int_a^b \max(0; f(t) - c)\, dt$$

Now we have to choose *c*, such that *L* becomes minimal.

However, reality is not as simple as that. If we work with a high amount of **temporary workers**, we will have the challenge of **know-how preservation**. We have to care for knowledge transfer to our organization, when temporary workers or freelancer leave our organization. And independently from their category we have to manage all our human resources. Probably it will cause higher efforts to manage external personnel than to manage own staff. In cost management this effect is described by the term of **transaction costs**.

There are two main arguments, why we should always consider to work with external people:

- if we need a special know-how or skill **temporarily** and it would not make sense to train own staff,
- if we need personnel capacity to cover **peak workload**, e.g. for a big project.

A lot of IT managers have a rule based on painful experience: They have a limit for the usage of external workforce. You will often hear the value of 30%. That means, that the portion of external workforce is limited to a maximum of 30% of the total workforce.

4.6 HARDWARE MANAGEMENT

Every organization, which supports its business processes by means of IT, will need a lot of hardware:

- servers for computing activities and processing data,
- storage systems to store data,
- network equipment to transfer data,
- devices for receiving data,
- devices for delivering data.

Due to standardization this equipment (in many, but not in all cases) can be **purchased or rented** from different providers. During the last years cloud computing has come up, and this technology enables organizations to use servers and storage machines without having or seeing any piece of hardware. Computing and storage power have become characteristics of the network (Internet). This was meant or addressed by the term **IT doesn´t matter** (Carr 2003). Hardware (in most cases, especially in the commercial world) has become a commodity and can be managed as such.

We have to define the needs and then find out the quantities and finally decide, whether to purchase or to rent the necessary equipment or to buy appropriate services without seeing the technical components behind it. Such decisions are driven by two factors, **price and risk**. Usually both factors are linked. Low prices often mean high risk, and lower risk leads to higher prices. Performance management has to find out the optimal (technical and economic) solution.

Be very careful with general statements like "Leasing is better than buying" or "cloud is always the best way to proceed". Each single **situation** has to be analyzed carefully, and each **decision** has to be made specifically. Things are also **time-dependent**. A good decision today can be a bad decision tomorrow, e.g. due to technological or economic changes. Performance management experts should be sensitive. Decision-making is a hard job and (usually) there are no simple solutions.

The major problem with purchased hardware is, that you can buy additional units quite quickly, when your capacity needs increase. But if your capacity needs go down, nobody will buy the used equipment or only at a lousy price…

4.7 SOFTWARE MANAGEMENT

What is software? If we consider an executable piece of software then it simply is a **machine**, which needs another machine to be executed. But it is a machine and could be considered as such. However, there are some differences to typical hardware components like servers, desktop computers, or mobile devices. To build a copy of a piece of hardware is not easy, and it causes significant costs. To make a copy of a piece of software is just to copy a file, and the costs of the copy are (near) zero.

You need software to **support** your business processes or to **run** your business processes, if they have been automated completely and do not need human interaction. Now you have three alternatives:

- You can develop your own **individual software**,
- you can buy or rent a **software package** and install it on your computers, or
- you can use a **"software as a service"** (SaaS), which is provided in the cloud, and thus you will never see a piece of software, but you will be able to use it.

What are the advantages and disadvantages? Developing and then maintaining **individual software** will be a cumbersome and time-consuming task. You will need a team of good software developers. However, the own software may improve your **competitiveness** in your organization´s business. It may allow you to deliver an **added value** to your customers.

On the other hand there are a lot of good and powerful **software packages** on the market. Why should you develop your own E-Mail system, a text processing system, a bookkeeping system, or a human resource management system? Why do we think that developing own software does not make sense at all?

One argument is the **price**. Consider what you have to pay for commodity software like a text processing package. You would never be able to develop it cheaper by your own, even if your organization had thousands of employees.

The second argument is **time**. If you buy ready-to-run software, you can install it today on your computers and use it tomorrow. It does make an economic difference, whether you can work with software tomorrow or have to wait months until you can start to work. The earlier you can start to work with a software the earlier you can earn the **benefits** from it.

Now there is a third argument for using software packages. This is relevant for big packages like bookkeeping systems. Because the software company has developed it for various customers, the software (if it is a good and well established software package) contains a **big variety of functionality** to satisfy the variety of customer needs and customer requirements. So, if you buy a good software package, you have a lot of functionality available. Even if you do not need or want to use it actually. But you can switch it on, when you need it (It may be that you have to pay an additional fee to your software provider then…). The advantage of such a software package is that it provides much more functionality than you are able to use today but you have it available every time you need it. And it is already integrated into the system.

Alongside with this stock of functionality there is a fourth argument to decide for a software package. It is the argument of **risk**. The mentioned examples of a bookkeeping or HR software are dedicated to the area of **supporting processes**. Here a lot of **legal requirements** have to be considered. And legal requirements change. Sometimes our feeling is that they change on a daily basis. If you run a self-developed software in those areas, then you have to care for adapting your software to the ever changing legal requirements. If you use a software package then the software provider is on duty to prepare his software accordingly.

There are a lot of areas, where it is at least questionable to develop own software. Consider, for example,

- workflow management systems,
- document management systems,
- data analysis systems,
- online shop systems,
- materials management systems, etc.

The general rule may be: **Standard software for standard functions**. But again: be careful with such pre-judgments! It could make sense in your specific environment to not follow the main stream because this could lead to a USP (Unique Selling Proposition) in your specific business. And finally it is not the question of Yes or No, self-developed software or software packages. You will use both. And big packages have interfaces, so that you can add own software to them or combine the package with your own "apps". Some packages even provide an **integrated development workbench**, so that you can enrich the package with your specific functionality. As ever, the challenge for performance management is to find the **optimal mixture**.

The third alternative is using **software as a service** (SaaS). Here you do not get a copy of the software. The software is running on a server somewhere in the world, and you only get **access** to the software. At the moment those offers are usually combined with the **pay-as-you-use** philosophy. But we are sure that very soon also this market will follow the **flat-rate** principle in the telecommunications area.

The advantages of those SaaS offerings are that it takes only minutes if not seconds for the **registration**, and then you can use the software. It is as easy as shopping in an online shop. The disadvantage is that you have to take the software as it is. There are no **customizing options**. May be that this will come up with increasing maturity of this software business. The second major disadvantage of SaaS is that your **data are stored outside** your organization, and you do not know, where, in which country of the world, your data assets are stored. **Industrial espionage**, done by your competitors and foreign secret services, is a fact, and hence, storing data outside and not knowing, where they are stored, could be a **serious risk for your organization**. However, we are sure that things will change in the future. The providers of SaaS have to become trustworthy. They have to protect their **customer´s data** better than their own data. And they will tell you, where they store your data. Of course, it may be that you then have to pay a higher price for those services. Performance management experts should attentively follow the development in this area.

Figure 17 summarizes the investigations about the different software categories.

	Individual software	Software Package	Software as a service
Advantages	*fits perfectly to needs, software under full control*	*ready to use, includes experience of all customers, no time for development needed, provider ensures compliance with legal regulations*	*ready to use, no technical requirements except internet access and browser*
Disadvantages	*long development time, depends on quality of requirements, lock-in effect with own employee*	*lock-in effect with supplier organization, cash out for licenses, software available to all competitors*	*lock-in effect with supplier organization, software available to all competitors*

Fig. 17: Comparison of software categories

Now let us finish the investigation on software with two questions. First: What will happen if the **software provider disappears from the market** either by being purchased and taken over by another (software) company or by shutting down his business? Secondly: Should we **buy or rent** software? And should we pay a single or a monthly lump sum, or should we pay per use?

To answer the first question: This will not affect you, if you only use self-developed software. If you have any software in use, which is provided to you by an external organization, there will be some **lock-in effect**. That means that you cannot easily change your software provider. This is, because your data are formatted according to that specific software. And most interfaces will somehow be **proprietary**.

If your software provider is taken over by another software company this may happen for different reasons. Either because of this acquisition the purchasing company can **expand its software portfolio**. In this case there is a very good chance that you can use your actual software for a long time (However, you should try to re-negotiate your contract…). Or the purchasing company wanted to **eliminate a competitor** from the market. Then they will sooner or later stop to extend and to maintain your software. Perhaps they can make you a very **attractive offer to migrate** from your actual software to their equivalent package.

To answer the second question, let us first consider self-developed software. It is developed by own employees, and they must get their salaries while working. You must be able – as an organization – to pay these salaries. If you are not able to pay the salaries, then you will have to raise a credit. But will you get it, and will you later on be able to pay the monthly installments for this credit? If you use packaged software you could either purchase it, and pay the purchase price (which is similar to the self-development of the software), or you could rent the software, and pay monthly or annual fees for it.

Usually the totally paid amount will be higher for renting then for purchase. This is a disadvantage of a rental solution. But the advantage is that you do not have to pay a huge amount of money. This reduces the need for capital, and **saves liquidity**. The liquidity, which you need for paying the fees, can be generated from the ongoing business and is financed by the financial benefits through the usage of the software.

But also annual fees are **fixed costs**. If you pay for your software as you use it, then you will pay more if you use it more often, and you will pay less if usage goes down. And you will pay nothing if you stop usage. Whether this is economically better for you than **monthly or annual flat-rates**, depends on the specific circumstances. The problem or challenge of every pay-as-you-use solution is the **monitoring of the consumption**. There must be a metric designed to measure the software usage, and this must be provided in the software functionality. But what is the right metric to **measure software usage**? The number of key strokes, the number of transactions, the number of data records created, modified, or deleted, the number of screen switches, etc.? All together? Obviously it is not as easy as it looks like. The well-known software companies today usually have pay models like a **fixed fee per month and registered user**. Sometimes they offer a payment by a **fixed percentage of the company´s sales** (e.g. in the market of ERP solutions). Both models seem to be correlated with the usage of the software, but there is not at all a direct relationship. For example, if the number of orders has doubled, but the sales volume does not change then the usage of the software will strongly increase but you will not pay more for the system if you have chosen the sales-based payment.

We forecast that also in the SaaS world the pay-as-you-use models will be eliminated and be replaced by flat-rate models, based on the number of registered users.

From a **legal point of view** you will never buy software. You will only buy or rent the right to use it. This is the background of the **floating-license model**. You buy a fixed number of licenses from the software provider, and arbitrarily many people are allowed to use the software, but only the contracted number in parallel. If you choose this model, then the number of contracted licenses is a typical **static capacity**…

4.8 SERVICE MANAGEMENT

Now let us investigate the fourth and maybe the most important resource, which is consumed in IT organizations, the external services. You may call it service provision, out-tasking or outsourcing, it is always the same strategy. Your organization has decided to do a specific subject not in-house but buy it **ready-to-use** from the external market. You have to make a **contract** with the selected external supplier, this supplier will deliver the contracted services, and you will have to pay for it according to the invoices, which the supplier will send to you.

What does this mean for performance management? First of all it must be clear, what the supplier has to deliver, which **quantities** at which **service levels**. Your organization must be able or must be enabled to **monitor the delivery**. You must be able to check, whether the sent invoices are correct or not, and whether you have to pay them or not.

May be that the monitoring of the service delivery is taken over by the supplier. Then you have to trust in his figures, and you should be able and be allowed to review his monitoring activities.

You have to **link your in-house systems** with the systems, which are run by your supplier. This is not always easy and sometimes very complex and challenging. You also have to run these **technical and organizational interfaces**. Independently from the degree of outsourcing you will have to master those interfaces between your and your supplier´s organization. This causes **transaction costs**.

For each task, activity, process, system, etc., which you do not do yourself but let it do for you by an external business partner you have to build up some **retained IT organization**. You must not under-estimate the necessary effects.

From an economical point of view the price for the external service plus the costs for the retained organization on your side must be lower than the costs, which occur if you decide to produce this service in-house.

Organizations, which practice outsourcing, usually cooperate with several supplier organizations. Hence they have to establish a **supplier management**. The activities of the different suppliers must be coordinated. From your customer´s point of view your total service delivery network of your own IT organization and the various suppliers must **operate as one seamless organization**. This is a real challenge.

4.8.1 OUTSOURCING AND INSOURCING

Outsourcing is a standard strategy for performance management. There is a potential that you can get needed services significantly cheaper than producing them in-house. But this effect must be measurable. On the other hand the costs for managing each supplier and coordinating the different suppliers may have a significant volume. Hence there can be a situation that insourcing reduces costs. As a performance manager you have to see both directions simultaneously, the **out-sourcing** and the **in-sourcing**. You have to find the **balance** of both strategies for your organization. This balance will **change over time** and has to be re-adjusted due to volume changes, changing and emerging technologies, changes in the supplier markets, changes in your business and your organization. The equilibrium is dynamic. And this is one of the every-day challenges in IT performance management.

4.9 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.4!

4.9.1 QUESTIONS FOR YOUR SELF-STUDY

Q4.1: Why should capacity management be a central task and have high priority for IT performance managers?

Q4.2: How can you find out, whether you have installed too much or too few capacity?

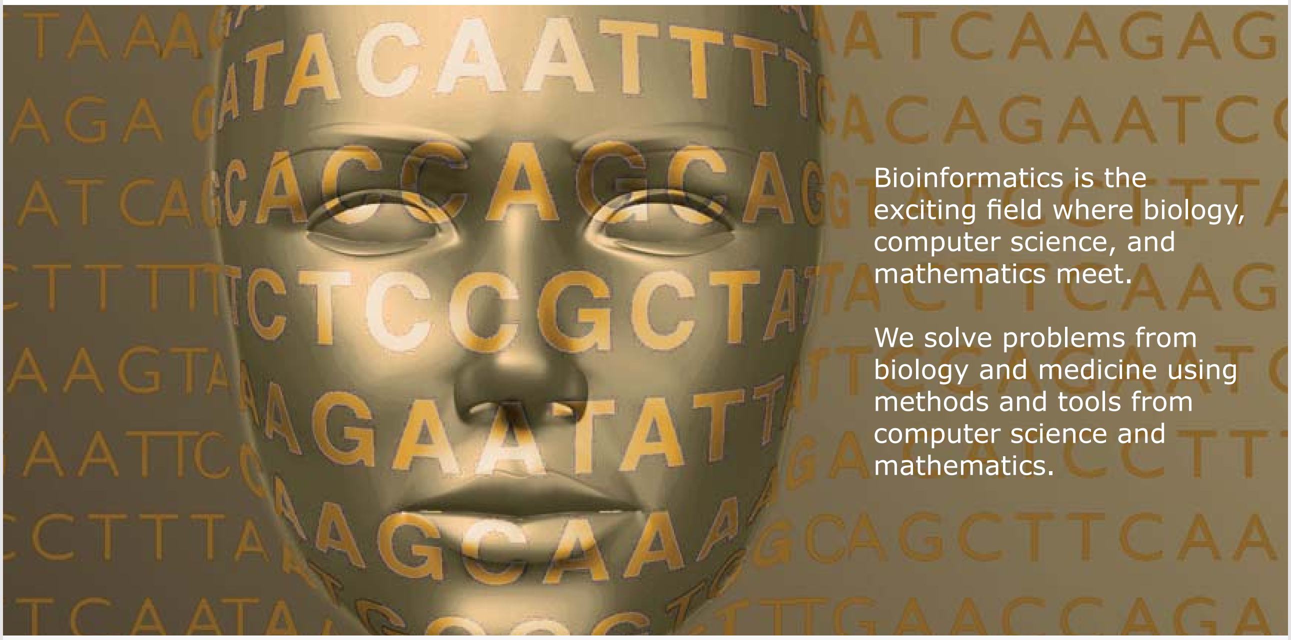
Q4.3: Comment the statement “IT doesn´t matter”.

Q4.4: What is the kernel of the zero-based budgeting approach?

Q4.5: Describe the advantages and disadvantages of outsourcing and insourcing for an IT organization.

UPPSALA UNIVERSITET

Develop the tools we need for Life Science
Masters Degree in Bioinformatics



Read more about this and our other international masters degree programmes at www.uu.se/master

4.9.2 PREPARATION FOR FINAL EXAMINATION

T4.1: Describe the term “capacity”. What are the similarities respectively differences between static and dynamic capacity?

T4.2: What is the impact of over-capacity? What is the impact of too small capacity?

T4.3: What are the pro´s and con´s of buying or renting hardware? Which impact does cloud computing have on hardware management?

T4.4: What are the advantages and disadvantages of individual software, purchased software packages and software-as-a-service offerings from the cloud?

T4.5: What are the reasons to buy IT services from external suppliers?

4.9.3 HOMEWORK

H4.1: Conduct a research on software license management. What are your results?

H4.2: Find out more about the outsourcing subject. What are the most important statements on IT outsourcing?

H4.3: Write a recommendation with respect to the usage of external experts in your IT organization. What are the opportunities and the risks for your organization?

5 SERVICE MANAGEMENT AND CHARGING

Learning Objectives

In this chapter you will learn,

- what service levels are and why we need service level agreements and a service catalogue,
- how we can measure the service performance and service levels,
- what cross-charging is and how it should be organised,
- which legal requirements must be fulfilled to practice cross-charging,
- what the relationship is between change management and service management.

Recommended pre-reading

- van Looy 2013

5.1 FOUNDATIONS OF SERVICE MANAGEMENT

Let us now investigate the output side of IT and the respective IT performance management. The outputs of IT organizations are services and project results. In this chapter we will consider services, projects will be investigated in chapter 6.

Before we start our investigation we have to define what we are talking about. Due to ITIL, the leading framework for IT service management (see van Looy 2013, pp 3 – 22) we define a service as follows:

A service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.

5.1.1 THE MEANING OF THE TERM SERVICE

First the service, whatever that is, facilitates **outcomes** in the customer´s organization. The customer organization wants to achieve something, and the service helps or supports to achieve this. The customer´s organization has a need for the service, and without that service it would not be able to facilitate the considered outcome.

The service provider takes over the responsibility of the **costs** of the service and the **risk**, that something might go wrong, when the service is produced or provided.

Finally the service must have a **value**, and ITIL is sharpening this by asking for the business value of every IT service. This **business value** is sub-divided into two categories, namely the service utility and the service warranty.

To get it to the point, the service utility is described by the fitness for purpose, and the service warranty is described by the fitness for use. The **service utility** covers the functionality from the customer´s perspective, and the **service warranty** covers the assurance, that the IT service will meet agreed requirements.

5.1.2 CONSEQUENCES

What are the consequences for an IT organization if it is going to deliver services to a customer´s organization? First the service must fulfill a **customer´s need**. It must **add value** (finally a financial value) to the customer´s business. Thus we talk about **economic efficiency**. The customer will ask for a service of an IT organization if it is cheaper for him to let the IT organization deliver that service than to produce it himself in its organization. Here we find ourselves at an interface between the IT performance management and the performance management of the customer´s organization. IT will have to determine the (unit) costs of the service. The customer has to find out what are the **savings** for his organization by sourcing that service from the IT organization.

There is another aspect of service management. The customer gets a **service** from the IT organization. He does not get pieces of hardware or software. He has transferred costs and risks of owning hardware and software to the service provider. And this also includes that the customer does not own hardware and software. This is included in the aspect of **risk transfer**. Hardware may be damaged or destroyed. The customer has transferred that risk to his service provider. And the service provider has to care for faultlessly working machines. Also software may be deleted, may have bugs and must be maintained. The customer expects that he is able to use the needed software, whenever he needs it to run his business.

In conjunction with this **ownership issue** we see that services have a very specific nature. A service cannot be produced, then stored in a warehouse, and later on be delivered to the customer. It must be **produced** just when the customer needs it, and it must be produced within a running business process in the customer´s organization. Things are done **on-site** and **real-time**. However, the service provider has to be able to produce the service, whenever the customer needs it. The IT organization must always **be ready to deliver** the requested quantity of service units. The fixed cost issue, which we have discussed in chapter 3, comes from this.

5.1.3 SERVICE CATEGORIES

Now let us consider IT services in more detail. Of course, they all will somehow be linked with hardware and software. Hence **systems** will play a big role in IT service management. But what are **typical services** of IT organizations? Based on the initial service definition IT customers want to use more or less complex systems, which are maintained and operated by the IT organization, to store, create, modify, and deliver information. From a business point of view we are talking about **application systems**. So, if we are going to write down a list of our IT services we will have a first and sound **service catalogue**, if we just list all our application systems, which our customers want to use and have to use due to their business needs. Our service is, that we hold the hardware and software assets, and that we guarantee, that the customer and his employees can work with those application systems, whenever they have to apply it to their business. Usually we differentiate between **business specific applications** and **general applications**.

Then we have to provide our services on-site at the customer´s location. It has turned out to define some general-purpose services for this. We are talking about **workplace services**. It makes sense to define this specific category of services, because their usage is (normally) not restricted to a single application. With a stationary desktop or a notebook or a modern mobile device the customer´s employee is able to get access to all application services, which he needs for his specific sets of tasks. This is similar for output devices like workplace or departmental printers or other output-generating devices.

And finally all those services will need some consulting and training, and this leads to a set of **complementary services**. All other areas should not appear as elements of the service catalogue. This is technical infrastructure, and the customer wishes, that this is managed by the service provider (see figure 18).

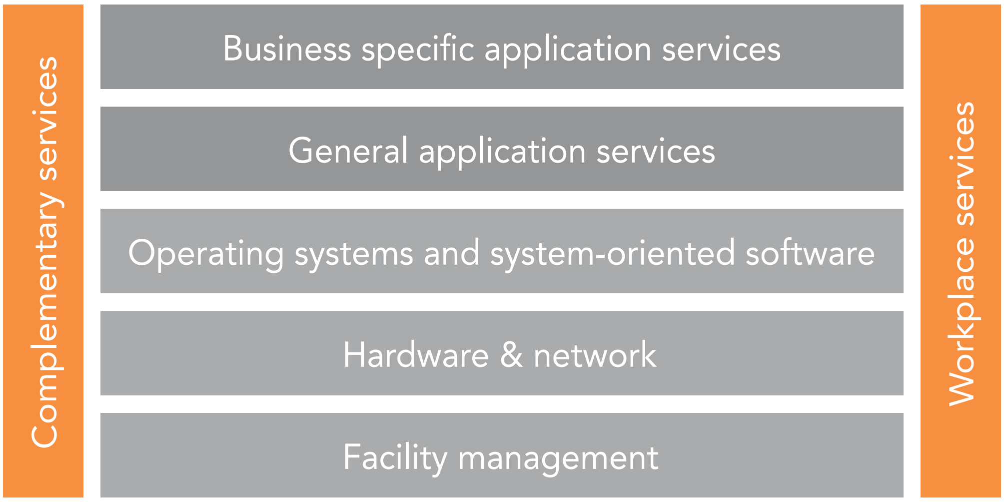


Fig. 18: Service categories

5.1.4 SERVICE CATALOGUE

What will be the size of our service catalogue? How many services shall we offer? Is there an optimal number? How many services an IT department has to offer to its internal customers depends on the **granularity** of the functionality, which is the basis of the service utility… However, software packages and hardware devices being installed in the customer´s environment may be a good indication for the number of services you have to provide.

Thus you will have a more or less long list of **devices** and related services installed in the customer´s environment. You will need at least

- a **desktop service**,
- a **notebook service** (which can be combined with the desktop service, if you provide a docking station notebook combination),
- a **mobile device service** (smartphone),
- a **workplace printer service** (possibly with an integrated scanning, fax, and copier functionality),
- a **department printer service** (possibly with an integrated scanning, fax, and copier functionality).

Usually you will offer more services or service variants in this area due to specific needs of your customers, e.g. a CAD workstation service, or a TCI workplace service, or a home office workplace service. It may be that you offer a (limited!) number of different desktops, notebooks, mobile devices, printers – if there is a business need. If not, you should restrict the equipment variants as much as possible due to the achievable level of economic efficiency. Norming and **standardization** will reduce the unit costs.

The **customer interface services** will include those software components, which every user needs, e.g. so-called **office packages**, **mail** and/or **communication functionality**, nowadays usually **internet access**. So you will bundle those micro-services to a catalogued service because users need the whole functionality, the total package.

Now let us consider the **application services**. Here the focal point is a specific software package and all other components and infrastructure elements, which are needed to facilitate, that the user can use the functionality in his daily business. How much application services an IT organization is going to offer depends on the structure of its application landscape. If it is dominated by **departmental applications** with a low level of integration and decentralized data, then you will have a high number of application services. If you have highly **integrated data** and **corporate applications** then you will only have a limited number of application services. So this depends on the architecture of your application landscape.

Last, but not least, you should offer a set of **complementary services**. This includes some standardized **on-request services** like the MAC service (MAC = moves / adds / changes), establishing a user account, re-setting of passwords, etc. It also includes in-house **trainings** and **seminars** as well as **consulting services**.

5.1.5 SERVICE BUNDLING

As an IT performance manager you have a dilemma. Having only a few services makes life easier for you with respect to cost accounting – also with respect to cross-charging (see chapter 5.4). But big **service bundles** have consequences. There will be a lot of users, who do not need or use the full functionality, which is provided by the "big" services. If you offer small and focused services, this will, from a customer s point of view, look fair and in accordance with the cause of costs. But cost accounting and cross-charging efforts will increase. If you have too many services in your service catalogue then transparency will get worse. This is, of course, also true, if you have too little services in your catalogue.

There will be, theoretically, an **optimal number of services** for each organization. But you will not be able to evaluate it precisely. Experience also shows, that there are some cycles or waves. At one time customers want to have a filigree catalogue of services, some years later they prefer the bundling of services, and another some years later the pendulum swings back…

5.2 SERVICE UNITS AND SERVICE QUANTITIES

We are investigating the tasks of IT performance management. And according to this we have to quantify and to measure. When talking about services we have to find out not only, which services have been produced or consumed, but also, how much of each defined service has been produced or consumed. If you want to count service quantities you have to define a **service unit**. But what is one piece of a service?

5.2.1 WORKPLACE SERVICES

Let us first consider the workplace services. If we deploy a desktop or a notebook, etc. at the customer´s site, then the customer would accept to pay for it as long as his team members really use it. If the device is not used, because the user has to do some work, which does not need IT supported functionality, then the customer would not like to pay for it or be responsible for the costs. The customer will usually follow the **pay-as-you-use paradigm**.

And here we are in the middle of practical IT performance management. If we follow this philosophy, then we have to **measure the usage** of the device. But what is usage? When the device is up and running? When the user has signed on? When he presses some keys, etc.? First you have to define, what usage is. And then you have to find out, how to measure it and store the data. This may be an (actually) unsolvable issue. And finally it is an issue of economic efficiency, too. The **effort** of exact measurement and processing the measurement results could be too high.

However, there is another question. If a user does not work with the device at his workplace, what is happening if he/she does not need access to any application service? Can somebody else use the device service then? But what is going on, if two people need this device at the same time? The business has to decide, whether one of both persons (and his/her business) has to wait until the device is free again. Or, whether every user has his/her own device, which is reserved only for him/her. This, of course, will cause costs of idle time of the device. Actually the **costs of device idle time** will be lower than the **costs of waiting business processes**. However, this might be different for very expensive devices, as it was in the early days of data processing with the mainframe computers.

But if the business requires **device services** and it is cheaper to assign a device permanently to a user or workplace then the customers have to pay or be responsible for the **total device costs**. And the corresponding service unit will be **one device for one time unit**. The time unit usually will be a month, but it could be different due to economic reasons. What does this mean? If the customer wants a device service for four people and for one year then he needs 48 device service units. Choosing a month means, that he can terminate the service contract on a monthly basis. So the paradigm in this device service area will not be pay-as-you-use, but **pay-for-the-potential-usage** (whenever you need it). The service does not mean to have a beautiful device in the office, but having the right to use the device for some time.

This right of use also applies to workplace printers. Considering departmental printers the situation looks different. The service here is to get **printed pages of paper**. And if two users send a printing service request to that device, then one user has to wait a few minutes until his/her request is fulfilled. Here the costs of user idle time can be neglected. In this environment the cost allocation could be based on the **number of printed pages**. Counting of printed pages is a standard functionality of modern printing devices. However, from a performance management point of view we should ask, whether it is necessary to allocate printed pages to single users. If such a printer is only used by the people in one specific cost center, is there a need to really know, which person in this cost center has printed how many pages? There may be good business reasons to know that. If not, we should avoid the effort even if it is a very low effort.

5.2.2 APPLICATION SERVICES

Let us now consider the application services. Here we have a similar situation as for the device services. Initially the customer will follow a **pay-as-you-use** thinking. And we as performance management experts should start with this, too. The theoretical approach will be to consider elementary functionalities of the underlying software. But a modern software package will offer hundreds or even thousands of those functionalities. If we follow this approach then we have to **monitor** the usage of each of those functionalities. If the software has not been designed according to measuring its usage on a functionality level, then we will not be able to conduct this measurement. But let us assume for a moment, that we would be able to measure the software usage on that micro-level. Then another question would arise: How do we **measure the software usage** altogether? Because used software functions may vary extremely due to run time, computing time, storage usage, etc.

Again we will come to the conclusion that the customer should not pay or be responsible for the real usage of the system, but for the **right to use** the application whenever he/she has a need to use it for supporting his/her business. This right will again be restricted to a time-unit, usually this right will be given for a month. Most software suppliers follow this philosophy. Their **licensing policy** is to assign such a license, namely the right to use the software, to **specific persons**, and they close their contracts with software "buyers" accordingly.

However, there may be a different licensing policy, the system of **floating licenses**. What does this mean? If you buy such software, then the supplier sells a specific bundle of usage rights, e.g. a bundle of 10. Now everybody in your organization is allowed to use the software as long as the number of concurrent users does not exceed the contracted number of licenses / rights. Each single right floats between the members of your organization.

The advantage of this approach for your organization could be, that much more than 10 people could use this software. The disadvantage would be, that you will have more or less often the situation, that people who want to use the software have to wait until one of the right to use the software is free again (see chapter 4.2 according to capacity management). Whether this is acceptable for your business depends on your specific situation. Whether you have this alternative to select between different usage models depends on the **sales strategies** of your software supplier. If your software supplier does not offer floating licenses, you cannot run this model in your organization. Of course you could realize this model for self-created software, but we have never seen this in real businesses.

5.2.3 COMPLEMENTARY SERVICES

Finally we have to consider the remaining services, which include training and consulting services on the one hand but also service like MAC (Moves-Adds-Changes) of devices, e.g. installing a device for a new employee, moving a workplace and the assigned devices, etc. This will lead to a sub-portfolio of services, which includes **standardized activities** as well as **individual activities**.

There will be a fixed unit cost assigned to each standardized service and these services will be initiated by a corresponding **service request**. The service unit will be a service execution. Usually an organization will have not more than 10 services of this type.

The second set of services is based on expert activities and the service unit will be an **expert hour**. Hence the service catalogue will contain different types of experts with different fixed costs per expert hour. These services are also initiated by orders or service requests. The consumed service units / expert hours depend on the task to be considered. Each request will lead to specific costs. Usually an IT organization will offer up to 10 services of this type.

Figure 19 gives an overview on the most used service units.

	Workplace services	Business application services	General application services	Complementary services
Typical service units	equipment month, printed page	user month (total business application or partially)	user month, business object month	activity, person-hour

Fig. 19: Typical service units for service categories Note: Service management does not consider projects. There are some good reasons for it. See chapter 6 for the respective discussion.

5.3 SERVICE LEVEL AGREEMENTS

Now let us discuss the single service. How can we precisely document, what such a service does and what it does not. The basic term for this documentation is the **service level agreement**. This is a document, which contains all relevant information about a specific service. What is meant with the term "level", and what is meant with the term "agreement"?

Let us first discuss the **agreement** part. What is agreed, and who are the partners, who are involved in such an agreement?

If we are on a public or external market, then an agreement is reached between a customer (of a service) and the provider (of a service). Within the boundaries of laws, economics and technology they are free in entering agreements. So, if the IT organization is an IT service company, they may have specific agreements with all their customers.

If the IT organization is an internal IT department or IT service center, things may be different. Of course could it be, that the IT organization has to enter into specific agreements with each customer, represented by a location, user department or cost center manager. But with respect to the economic effects of **standardization** (see economies of scale) the IT customers will not be allowed to make arbitrary and individual agreements with the internal IT service supplier. They will be obliged by **IT governance** and thus top management to accept **generalized agreements**.

There will be no local negotiations between the IT organization and the various other organizational units of that company or association or administration. The agreement will be made between IT organization and IT governance, possibly dictated by the IT governance to the IT organization. Hence within firms and other organizations a SLA will more or less be a **service level description** or documentation. However, the term SLA has been commonly settled and we will not introduce another term in this book.

Now let us consider the levels of our services. What does it mean? After a service has been defined and documented, it must be provided. And this can be done better or worse. Because we are in IT performance management we have to measure the level of the **service fulfillment**. How can this be done?

Because we are not able to measure a service in total, we have to select characteristics and in conducting a service those characteristics can reach different levels. We will define target levels for each selected (and hopefully important or relevant) characteristic, and we will measure the degree of the target level achievement. The service level agreement documents the **selected service characteristics** and the **corresponding target levels** to be reached. This usually will include a description of the selected **measurement method**.

So the service level agreement will document the **functionality** and **scope** of the service. It makes sense, not only to describe what the service will deliver to the customer, but also to describe, what the service will not include. E.g. for a desktop service it could make sense to explicitly document that the installment of a personal printer is not part of the desktop service.

Then the service level agreement will document, which characteristics have been selected for the service warranty and the corresponding target levels and measurement methods (see figure 20).

What it includes.	**Service unit and service windows.**
What it does not include.	**Characteristics / Service levels.**
Formal regulations.	**Responsible persons / duties.**

Fig. 20: Service Level Agreement

Now the reader might fear, that there is a great variety of relevant service characteristics. However, most of the characteristics, which we will see in real operations, will turn out to be variants of just a few general service characteristics. Let us try to identify the most important service characteristics (see also chapter 2.7). We see:

- availability,
- reliability,
- run times,
- adherence to schedules.

5.3.1 AVAILABILITY

Most of our services are **request-driven**. The service production starts, when the user initiates it. Thus availability is the most important characteristic of every service. Most readers will intuitively combine the availability term with percentages. We talk about 99% availability. But as we are considering performance management we have to ask, what this precisely means. From the customer perspective a service is **available**, whenever a user sends a **service request** and the **production of the service** starts immediately. Of course, there will be, due to technical or organizational constraints, some delay between sending a service request and starting the service delivery. Hence availability means, that the service starts within a defined time span after sending (or receiving) the service request. It may be, that the service (or the system behind the service) sends a **signal** to the user informing him that the service request has been **registered** and service delivery will start as soon as possible. Availability could mean to get a **service request confirmation** within a defined time span after sending the request (see chapter 2.8).

This very abstract exercise shows, that we have to define carefully what we understand by availability. And then, of course, how to measure it?

If we now would like to put availability into an indicator, we could it define as the ratio of the number of service requests which have got the proper service request confirmation, and the number of all service requests (within the considered period of time). Of course, the considered time interval must be identical in numerator and denominator of the ratio. The prerequisite to this approach is, that we are able to **count all incoming service requests**.

Sometimes (in most cases?) availability is measured a little bit differently, especially for the measurement of the availability of software systems or hardware components. A technical agent is installed, which sends requests (ping) to the system and then waits for the confirmation. These requests are sent **periodically** in a high frequency, e.g. every second. Availability then is a ratio again, namely the ratio of requests with a confirmation and the number of all requests. This approach assumes that the system will also be available between two subsequent requests, if it has confirmed both subsequent requests.

Now a second dimension of the availability issue comes up. (Application) Services cannot be requested from everywhere, but only from given points of service (POS). Hence the measurement of service requests and their confirmation must ensure, that all service points are included. We are looking for the **end-to-end availability** of the considered service. Technically does it mean, that we are able to measure it in each POS. This means, that every device must be able to collect relevant data. If we are not able to do this end-to-end measurement, then we have to define availability in a different way, and we must tell this our customers accordingly.

However, if we have different points of service we have to deal with the question, how to proceed, when the availability varies between different service points. Most publications and contributions tend to aggregate the service point availabilities into one (key performance) indicator by building the **mean value of all local availabilities**. We recommend to proceed differently, namely to measure the local availabilities and then to build the ratio of the number of service points, where the (local) **availability targets have been reached**, and the number of all included service points.

5.3.2 RELIABILITY

The second service characteristic, which is often mentioned, is reliability. What is this? We think, that reliability of a service means, that, if the **service production** has been started, it **reaches the forecasted or expected end**. We expect, that a running service does not break down, but is finished successfully. Considering the model, that the service execution is started by an order, which is confirmed to the customer, means, that the service breaks down after being started and does not come to its defined end. This service instance will never send a result or an end-of-operation confirmation to the customer.

We think, that due to the high reliability of technical components service reliability is not so important for IT performance management. Because we are allowed to expect, that with highly reliable technical infrastructure, software systems will always deliver the expected results, because software machines operate deterministic and not as randomized systems. But this may change in the future…

5.3.3 RUN TIMES

The next very important characteristics of a service are time spans, basically the total time span to execute a specific action, which is part of the functionality of the service. Starting again with our model of confirmations for accepting a service request and finishing a service activity, the duration of that activity is the time span from confirming the **acceptance** of the order to confirming the **finish** of the order execution. We only have to take the time for both confirmations. And this must be done for a huge number of service requests.

Now it is clear, what the duration of a single transaction / order execution is. But can we concentrate into one indicator the duration of thousands or even millions of transactions? Again we have the **mean-value issue**. And similarly to the service points with different target availabilities we will have different target durations for different functionalities.

Again the portion of transactions, having a duration, which does not exceed the expected duration, related to the total number of transactions. Ideally we now define specific target durations for each functionality. Is this a realistic approach? Usually organizations expect that 80% of all software transactions do not exceed 2 seconds in duration. We have implicitly considered software transactions, which have a very low duration (seconds or minutes for long running transactions).

Our thoughts can be applied similarly to **business processes**. Here we have durations of days or weeks, sometimes months. However, the structures are the same. To measure software transactions we can use the monitoring functionality, which is often an integral part of the software itself or some layer of middleware or system software. For business processes the measurement of starting and ending times must be realized. In case of **automated workflows** this is done by the underlying workflow management system. If we consider a more or less manual process, people have to **make time stamps**. We have to ensure, that this is done. Therefor we will need forms. And to evaluate the data afterwards we finally need software… So we have to establish some workflow and assign a **ticket** to each process execution. However, a **ticket system** will be available somehow in every IT organization. We can use it by defining an appropriate (new) category of tickets for every specific workflow.

Specific examples of run times are response times (see chapter 2.8).

5.3.4 ADHERENCE TO SCHEDULES

Let us now consider another important service characteristic. If there are some just-in-time deliveries then the adherence to schedules has a significant meaning for IT management (see chapter 2.6). This is relevant for totally automated workflows or transactions, which have **to start or/and to finish at a specific point of time**. Examples of those categories of services are to be found in the area of IT reporting (reports have to be delivered / printed until 12 o´clock of the third working day every month), closure of bookkeeping, daily planning of truck loading, etc. Here every transaction has to be checked, whether the target time was met or not. The natural indicator is the portion of transactions, which met the target time, related to the total number of transactions. You can calculate this value, if respective **time stamps** are set and documented.

It may be not only interesting to know, whether a target time was met or not, but also see the difference between the target time and the actually met point of time. Performance management should not only be interested in those transactions, which have been **too late**, but also consider those, which have been **early**. It may be reasonable to install two indicators, one, which measures the delays, and one, which measures the too early birds. Please, note, that we again have to master the mean-value issue here.

5.3.5 OTHER SERVICE CHARACTERISTICS

Now we have considered availability, reliability, duration of activities and adherence to schedules. What also could be a general service characteristic?

One characteristic could be the **effort of each transaction**. Usually a variety of resources or capacities is consumed by any transaction. There may be a dominating or cost driving resource. So IT management could be interested in **measuring the consumption** of this specific resource. However, the transactions must be enabled to measure this specific resource or capacity consumption! If we are interested in the consumption of human labor, then it must either be documented automatically (by taking time stamps for beginning and ending with a specific piece of work) or manually in the way, that people write down, what they have done and when. There will always be some effort for monitoring or capacity consumption.

If we were interested in the resource consumption globally we could measure this by determining the **unit costs of a transaction**. However, in this case we should be able to assign costs to the respective cost unit, e.g. an application system. If we have assigned all relevant costs to a specific transaction or service, then the unit costs are the ratio of all documented service specific costs and the number of service units being delivered in the considered time interval.

The other service characteristic, which is of interest, is the **satisfaction of the customer or the user with the service**. First we have to differentiate between the customer and the user perspective. The customer is a manager, who decides, whether he takes a specific service for his area of responsibility. The user is a person, who needs service units for his daily work. The user is the person who sits before the screen of a desktop, notebook, or other end devices and suffers from bad usability of (application) services.

Now we have to answer the question, how we can or should measure the satisfaction of a group of people with a specific subject.

The first and most intuitive approach is to ask the people, how they value the delivered service. You do some "market research". The method to be used here is the **benefit analysis** or **scoring**. The advantage of this approach is, that you get direct feedback from the polled people. The big disadvantage is, that this interview-like approach causes a lot of efforts. So it does not seem reasonable to do it on a monthly basis. There are other critical aspects of scoring, which will be discussed in chapter 9.

Do we have alternatives to measure customer or user satisfaction? The traditional approach would be, not to interview all customers or users but only take a **random sample**. However, there would be some effort to do it on a monthly or even weekly basis. If you feel it the best way to let people answer questions, then you could reduce this interview and scoring approach to just one question: Have you been satisfied with our service during the last 24 hours? If you restrict the answer to YES / NO, then the portion of all answers being YES would be an indicator for customer or user satisfaction. This polling could be integrated into a single-sign-on procedure or you install a small application, which wakes up every 24 hours…

Are there further alternatives? Of course, but now we have to change our view and look for **indirect measurements** of satisfaction. Sometimes it is easier to look for the absence of satisfaction instead of looking for the presence of satisfaction. The idea is, that the users can be sub-divided into a subset of satisfied users and another subset of unsatisfied users. Thus, if we know the portion of α unsatisfied users, then we also know the portion $1 - \alpha$ of satisfied users. But how can we **measure the level of dissatisfaction**? We can just give a few examples of approaches, which we have found in real organizations.

The **number of incidents** is an indicator for dissatisfaction. If this number goes up, we assume, that the level of satisfaction goes down. Some organizations have implemented some kind of **complaints management**. Similarly to the volume of incidents we assume, that a high number of complaints shows a low level of satisfaction. Of course, there will not be a 1:1 relation between incidents / tickets / complaints and the level of satisfaction. But it is better to measure indirectly or imprecisely than to measure nothing. And IT performance management should be very careful in interpreting the results of measurements. **Indicators show symptoms**. To find the **real causes** needs professional diagnosis, not only in medical science but also in IT management.

These last considerations show, that finding the "right" measurement approach needs some **creativity**. You always have to answer the question: Does my metric really measure, what I am looking for? And we have to find the **balance between precision and cost of measurement**. Of course, also IT performance measurement has to follow the principle of economic efficiency.

5.4 CROSS-CHARGING

Now in the final part of this chapter we will look for the **financial value of services**. We need to know this to answer the question: Is it better – in the sense of economic efficiency

– to produce a service or a part of a service in our own organization, or should we buy it from an external IT service provider? The service provider will offer his service(s) finally by telling his **price**. And we should be able to know our internal “price”. That means, that we should know the **unit costs** of the respective service, if we produce it completely within our own organization.

If we ourselves are an IT service provider, which sells services to **external customers**, then we have to tell our customers, which prices they have to pay, if we deliver our services to them. In addition, if our organization is a **group of legal entities**, then we have to make contracts with the group members and write invoices if we deliver our services to other group members. If we deliver our services only to **internal sub-organizations** within the same legal entity, then there is no legal need to write invoices. But management could decide, that the IT organization has to behave like an external service provider.

Considering the financial value of a service resp. a service unit we have to **differentiate between unit costs and price**. The unit costs can be calculated, the price is set by a **management decision**. If I sell my products on a market, the price must be higher than the unit costs, because my firm has to make profit. The profit is the interest for the capital, which somebody, e.g. shareholders, has invested in my company. If my company cannot pay the expected cash dividend, the investor will take his money and leave my company. If the market does not accept my price, I have to stop selling the respective goods or services or to reduce the (unit) costs to increase the profit margin, namely the difference between price and unit costs.

5.4.1 COST DISTRIBUTION SHEET

However, unit costs are the basis, and here we can and have to apply cost accounting methods. The tool, which we have to use here, is a **cost distribution sheet**. This CDS can be considered as a complex algorithm to take the costs we have and being structured by **cost types** and **cost centers**, and **re-allocate** them according to our **service catalogue** representing the **cost units**. We must find a way to distribute each currency unit of costs among all our services. The simplest approach would be to do this transfer without any intermediate step by building a table, where the rows represent all (used) cost-center cost-type combinations, and the columns represent the services we offer. There are few organizations, which practice this somehow brutal approach. Every cost center manager has to "explain" his costs by directly assigning them to specific services. His / her costs increase the service costs accordingly. Figure 21 shows this straight cost distribution.

However, this is a very simple model of the internal **IT value chain**. We know, that IT organizations have a lot of indirect costs and several functions, which are finally needed to deliver the services, but there is no direct relationship between the function and any service. For example, consider IT management, IT performance management, IT security management, technology and architecture management, etc. The IT world seems to be very complex.

What can we do? Let us change our perspective.

Cost-center cost-type combinations	Budget	Services				
		S1	S2	S3	...	S99
CC1 / CT1	999	111	111	111	...	111
CC1 / CT2	999	111	111	111	...	111
...	...	...	...	...	...	...
CC1 / CT99	999	111	111	111	...	111
CC2 / CT1	999	111	111	111	...	111
CC2 / CT2	999	111	111	111	...	111
...	...	...	...	...	...	...
CC2 / CT99	999	111	111	111	...	111

… …	… …	… …	… …	… …	… …	… …
CC99 / CT1	999	111	111	111	…	111
CC99 / CT2	999	111	111	111	…	111
…	…	…	…	…	…	…
CC99 / CT99	999	111	111	111	…	111
Service costs		999	999	999	…	999
Service units		333	333	333	…	333
Service unit costs		3	3	3	…	3

Fig. 21: Cost distribution sheet

5.4.2 PERFORMANCE CENTER

If we do not consider our cost centers as sources of costs but as **sources of output**, then we can build a model of our IT value chain. For each cost center, which now is considered as a **performance center** (see figure 22), we can identify, which (internal) service they deliver to other IT cost centers. So each IT cost center manager has to answer the following questions:

- Which services do I deliver?
- Who is my (internal) customer?
- Which quantity of service units do I deliver to my customer?
- Which costs do my services cause?
- Which amount of costs can I transfer to the cost centers of my customer?

Under this philosophy the IT manager could come to the conclusion, that he provides management services, according to the financial volume in the IT department. If he, together with his secretary, causes costs of 150.000 EUR p.a. and the total IT costs will be 6.000.000 EUR p.a., then the unit costs of his comprehensive management service is 25.000 EUR per 1.000.000 EUR of IT costs (of course not including the 150.000 EUR of his own cost center). The technology manager may define his service unit as a working hour of him or his team members, and so he has to find out, how many hours he will work or has worked for other sub-organizations of the IT organization.

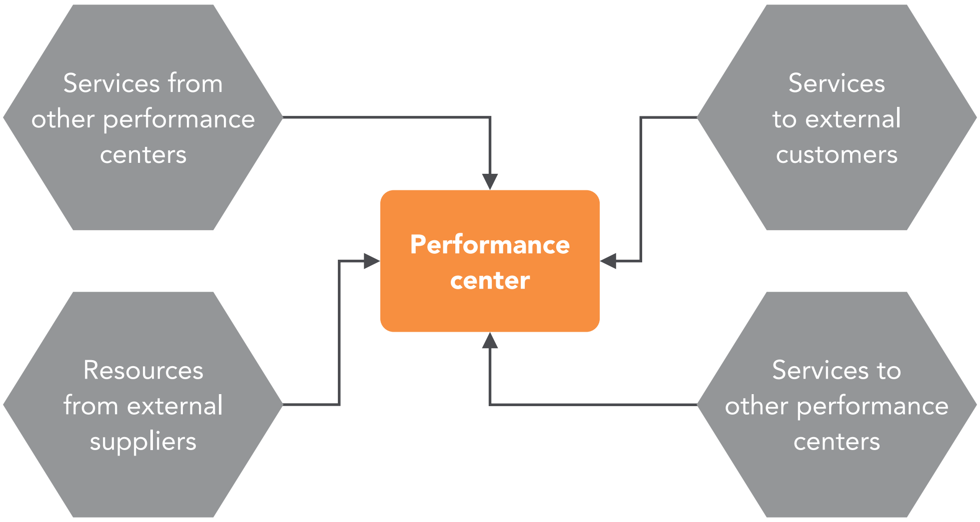


Fig. 22: Cost centers as performance centers

The reader may criticize those choices of service units. Indeed, in his / her specific environment it could make sense to choose other service units. You have to find out on-site, **which approach leads to best transparency**.

Now let us investigate the performance-center concept and re-fine the cost distribution sheet. For reasons of simplicity we assume, that the value chain in our IT organization has a structure that each performance center within any sequence of performance centers subdivides this sequence in a way that all other performance centers belong either to the suppliers or the customers of this performance center. This means that the network of performance centers does not have cycles. Figure 23 shows such a network of performance centers without cycles. We will later investigate how to proceed without performance center networks including cycles.

What does this mean for cost planning and distribution?

Ideally we would start with the primary cost centers, delivering services to external customers. They all have to determine the services and the related quantities, which they deliver to external IT customers. Then they have to determine the **needed quantities of resources and internal IT services** on the basis of the **bills of materials** for the external services.

If a service center delivers services to external and internal customers, it has to wait with the planning of internal services until its internal customers have notified their demands.

Now we can go backwards through the whole network until we have reached those service centers, which do not need internal services but only resources from external suppliers.

At the end of this planning step we have a total plan of all quantities of external services, internal services, and resources.

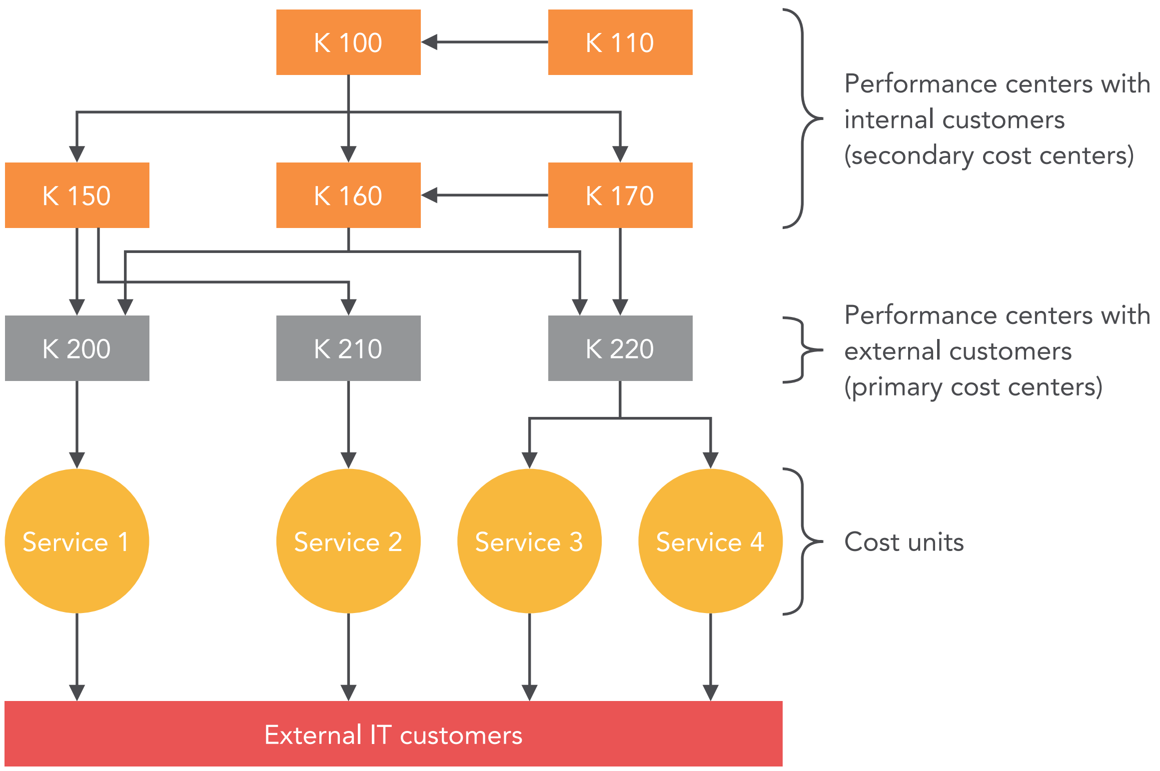


Fig. 23: Value chains without cycles

Now the second part of the planning starts. Each service center has to plan its needs of resources and internal services financially. We do this by weighting the resource quantities with (expected) purchase prices. Applying the bill of materials first to those performance centers, which do not need internal services, allows us to calculate the costs and unit costs of the internal and external services delivered to internal and external customers. Now we go through the whole network forward and step-by-step calculate the costs and unit costs of internal and finally external services.

What is the conclusion? The cost distribution sheet is nothing but the model of our **IT value chain** resp. the **network of service centers** in our IT organization. Furthermore we have to know the bills of materials for every internal and external service, which is produced in our organization. Do we have this information in real organizations? Do they even know their internal services exactly?

How can we proceed in real business environments, where most IT organizations do not have a precise bill of material for their services, not even for their external services?

The situation seems worse than it really is. Usually we have the actual values of the last planning period(s). We know, which quantities of services we have really produced, which resource and capacity quantities really were consumed, and how much costs we really had. Hence we should be able to create a **rough model** of our value chain, where secondary cost centers at least have to describe, that they produce a big **service cake** and how much of this cake they deliver to other (internal) service centers.

From a theoretical point of view this is not an approach, which we can recommend. But in real life it is better to have a coarse model of our organization than to have nothing.

However, there is always the danger, that costs are distributed as is, and also unnecessary costs, e.g. costs of idle capacity, are distributed. How can IT performance management protect the various IT sub-organizations against such a fogging of (idle) costs?

Well, as we have already mentioned it, we have the actual values. Each service center knows (or should know), how much it has produced and delivered and how much costs it had. Now performance management has to initiate a **sensitivity analysis**, asking, what would happen, if the service center had to increase its service delivery. There will be a percentage c, such that the output could be increased from 1 to $1 + c$ without an increase of costs. If we find out that the value of c is small, e.g. lower than 5%, then the over-capacity may be acceptable due to technical or risk reasons. But if the value of c is higher, then we should ask for the reasons, and who has to pay for this over-capacity.

Let us come back to the IT manager and his approach that a management unit is managing 1 million EUR of IT costs. If this approach is realistic, then this would mean, that he will have to work 10% more, if the IT budget increases by 10%. If this does not happen, then he should look for a better definition of his management performance unit.

The unit cost calculation is not easy, of course. The reason is not, that it is challenging from a mathematical point of view – not at all. The reason is the **complexity** of the real world and to find **simple but appropriate models** to describe it. And he will have to master the challenges of fixed costs and indirect costs, which often do not occur isolated, but come up in combination. And the problem and real challenge for many IT experts is, that they are not aware and not trained to describe their **work output in terms and quantities**. They often live due to the motto: I do, what must be done. IT performance management has to open out this old-fashioned way of thinking.

5.4.3 VALUE CHAIN CYCLES

Let us now finish the cost rate considerations with investigating a real-life problem, when building cost distribution sheets. We will often find, that a cost center A delivers something to a cost center B, then cost center B delivers something to cost center C, and finally cost center C delivers some services to cost center A. Obviously we have a cycle in our value chain. How should we proceed with this? An extremely simple example is shown in figure 24.

This figure must be read as follows:

- CC-1 needs the internal service S5 and the resource R1 to produce the external service S1 and the internal service S3. This leads to the demand functions $s_5 = f_5(s_1, s_3)$ and $r_1 = g_1(s_1, s_3)$.
- CC-2 needs the internal service S4 and the resource R2 to produce the internal service S5. This leads to the demand functions $s_4 = f_4(s_5)$ and $r_2 = g_2(s_5)$.
- CC-3 needs the internal service S3 and the resource R3 to produce the external service S2 and the internal service S4. This leads to the demand functions $s_3 = f_3(s_2, s_4)$ and $r_3 = g_3(s_2, s_4)$.

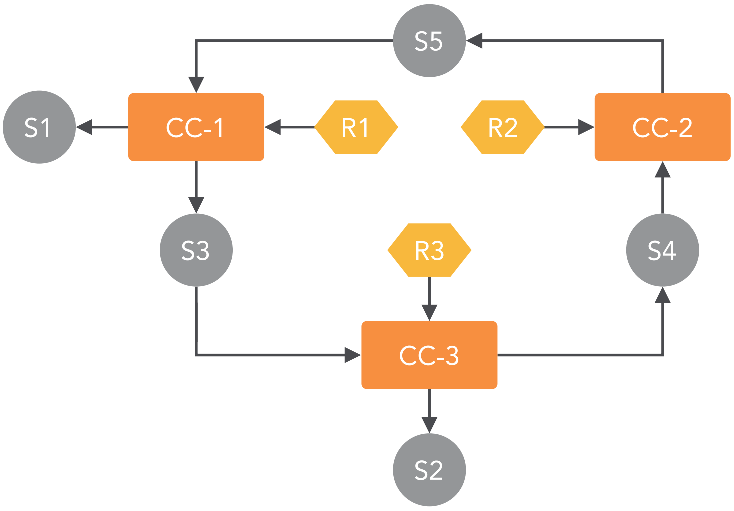


Fig. 24: Cycles in value chains (example)

How can these demand functions help to determine quantities? Starting point are the requested quantities of the external services. Here we have the given quantities $s_{1,0} > 0$ and $s_{2,0} > 0$. All other quantities are set to 0 in the beginning: $s_{3,0} = s_{4,0} = s_{5,0} = r_{1,0} = r_{2,0} = r_{3,0} = 0$. Now we follow the algorithm, which is described by the following equations:

- $s_{1,n+1} = s_{1,n} = s_{1,0} > 0$
- $s_{2,\, n+1} = s_{2,n} = s_{2,0} > 0$
- $s_{3,n+1} = f_3(s_{2,n};\ s_{4,n})$
- $s_{4,n+1} = f_4(s_{5,n})$

- $s_{5,n+1} = f_5(s_{1,n}; s_{3,n})$
- $r_{1,n+1} = g_1(s_{1,n}; s_{3,n})$
- $r_{2,n+1} = g_2(s_{5,n})$
- $r_{3,n+1} = g_3(s_{2,n}; s_{4,n})$

We could start the algorithm and increase n as long as the differences of the quantities between two subsequent values are greater than 1. It may turn out, that another rule to stop the algorithm is more appropriate… At the end we will have all needed quantities. Now we can value the quantities of the resources with the purchase prices and then we have to calculate the costs of the internal and external services. Thus we have to work with the cost functions:

- $cr_1 = cr_{1,0} > 0$
- $cr_2 = cr_{2,0} > 0$
- $cr_3 = cr_{3,0} > 0$
- $cs_1 = cf_1(cr_1; cs_5)$
- $cs_2 = cf_2(cr_2; cs_4)$
- $cs_3 = cf_3(cr_1; cs_5)$
- $cs_4 = cf_4(cr_3; cs_3)$
- $cs_5 = cf_5(cr_2; cs_4)$

We could start the algorithm and increase n as long as the differences of the quantities between two subsequent values are greater than 1 or 0,01 c.u. It may turn out, that another rule to stop the algorithm is more appropriate… However, these calculations will be cumbersome. Even in our very simple example we have to run those calculations for several variables. In real environments we will have to handle hundreds or even thousands of variables.

What can we do to come to pragmatic approaches? It sounds brutal and making things diffuse. But we should **break up the cycles** and model an IT value chain without cycles. Of course, our cost distribution will become more imprecise. But has it been precise from the beginning? Remember our considerations on indirect costs. We are never able to distribute indirect costs in a just way. If we were able to do it, then it would be direct costs.

The heuristic rule is to cut the service exchange relation with the lowest value in such a circle. In our example, shown in figure 24 this could be the delivery of S5 from CC-2 to CC-1. What are the effects? The receiving unit, where the line is cut, has a little less costs. The other receiving units have to carry slightly higher costs. At the end the cost rates of the services will not be totally correct. But does it matter, if calculated cost-rates go up or down by 0,1 to 0,5%? We are convinced, that the economic effects of idle costs or missing capacities are much more influencing the business.

If you are a very careful performance analyst, then you can calculate variants of the cost distribution sheets by cutting the discovered value chain cycles at different supplier-customer-relations.

5.4.4 SETTING PRICES

Now let us assume, that you are ready with the cost rate calculation. How should you make the final step to set prices? If you are selling your services to external markets, then you are totally free in setting your prices. The market including your customers as well as your competitors will tell you, which prices are **acceptable**. If you are delivering your services within a legal entity, then the price setting is the result of your **management culture**. You can make specific services very expensive if you want to eliminate those services from your service portfolio, e.g. services, which are based on a technology to be scrapped. And you could sell services with prices below the unit costs, if you want to promote those services, e.g. services, which are based on a new and emerging technology.

5.4.5 TRANSFER PRICING

However, things are different in an organization, which consists of several legal entities, especially, if those entities are active around the world and are located in foreign countries. You have to sell your IT services then formally and write invoices. Money will be transferred between different legal and tax paying entities. Here you are **not allowed** to set prices arbitrarily. What is the reason?

To understand the mechanism let us consider an example. Let us assume, that the IT organization is a member of a legal entity X in country A. And that it has to deliver IT services to a subsidiary Y, which is a legal entity in country B. If a price of a service is increased, then the revenues and, more important, the profit of company X increases, too. On the opposite side the costs of company Y also increase, and thus profit goes down. Hence the fiscal authorities in country A will love this situation, because they get more taxes from X. The fiscal authorities in country B will not, because they get less taxes from Y.

The problem becomes even bigger, if the fiscal authorities in country B find out, that the service, which Y gets from X, is available on the domestic market and offered there at lower prices, e.g. due to significantly lower salary levels in country B. Then they will possibly not accept the totally invoiced amount from X, but only the lower equivalent from the domestic market. Thus accepted costs of Y will decrease, profit will increase and tax payments in country B will increase accordingly.

If now the IT organization of X would try to avoid that trouble and sell their services at prices, which are accepted by the fiscal authorities in country B, then it is quite probable, that prices will be lower than the unit costs. Now the fiscal authorities in country A will not agree to this proceeding due to reduced profit of X and resulting lower tax payments. It may happen, that they take the services, which X has delivered to Y, and value them according to market prices in country A. This will lead to fictitious revenues of X and subsequently to fictitious profit and finally to real and higher tax payments.

Obviously there is a **dilemma** in delivering services to subsidiaries in foreign countries. To solve the problem the concept of **transfer pricing** has been developed. For enterprises with different legal entities the basic rule for invoicing deliveries within the group is the **arm´s length principle**, which can be summarized into two sub-rules:

- Make a price that you also would make for an independent third party.
- Sell the service to all members of your enterprise group at the same price.

Of course this does not solve all problems, but it gives a sound guideline. We still have the problem of different cost levels in different countries. Though the "theory" of transfer pricing has developed some widely accepted methods of price setting (see Abdallah 2004), the ultimate approach is to come to an agreement with both involved fiscal authorities.

The standard approach for transfer pricing is the so-called **cost-plus method**, which means, that the price is set by calculating the cost rate and then increasing this cost rate by a given percentage of the cost rate. Fiscal authorities will ask you for your **cost rate calculations** and the used **upcharge percentage**. Defining this margin is neither in the responsibility of IT performance management nor in the responsibility of IT management. Top management must do this.

5.5 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.5!

5.5.1 QUESTIONS FOR YOUR SELF-STUDY

Q5.1: How should we define service units? Consider this for the different service categories separately. Find ideas for the pay-as-you-use approach.

Q5.2: Write down the pro´s and con´s of bundling respectively unbundling of IT services.

Q5.3: Give examples of services, where run-time or adherence to schedules are the most important service levels from a business point of view.

Q5.4: How should we distribute network costs and management costs to services?

Q5.5: Write down the pro´s and con´s of charging for IT services.

5.5.2 PREPARATION FOR FINAL EXAMINATION

T5.1: What is an IT service? Give 2 examples of an IT service.

T5.2: What is a service level agreement? Give 2 examples of service levels.

T5.3: What is a service catalogue? Describe its relationship to cross-charging.

T5.4: Describe the function of a cost distribution sheet and how it can help to calculate unit costs.

T5.5: How do we find prices for IT services? What is the relationship of pricing to unit cost calculation?

5.5.3 HOMEWORK

H5.1: Find examples for service level agreements and service catalogues from the internet. Did you expect to find more or find less? If your expectations have not been met what could have been the reasons?

H5.2: How do international organizations master the problem of different cost levels in different countries?

H5.3: Conduct a research on handling of IT value chains with cycles. What did you find?

6 PROJECT MANAGEMENT AND BUSINESS CASES

Learning objectives

In this chapter you will learn,

- what a project is and why organizations need IT projects,
- how business cases can help us to make decisions to run a project or not or which projects we should select,
- how we can measure the performance of an IT project and how we can conduct a project evaluation,
- how we should manage changes of a project and when we should decide to stop a project.

Recommended pre-reading

- Kuster 2015

6.1 FOUNDATIONS OF PROJECT MANAGEMENT

Projects are one of the most important management objects of IT management, and they are of specific interest for IT performance management. But what is a project? Let us follow Kuster 2015, p6:

> **If a one-off initiative extends across departments, has a limited time frame, is focused on a specific objective, is interdisciplinary and is so important, critical and urgent, that it cannot be easily managed by the existing line organization, but instead needs special organizational measures to be taken, then we call it a project.**

To become very general we could state, that a project is every activity, which can be properly managed with approaches, methods, and tools of project management.

However, let us choose our often-used IPO model (input – process – output), here to be re-interpreted as **input – project – output**. What are the inputs of a project? What are the outputs of a project?

As we have experienced it for several times now, the input is quite easy to describe. A project will consume **resources**, and those resources will have a **financial value**. Please note, that we do not talk about costs, but about a financial value. The reader will soon understand, why we use a different term here. To describe the output of a project seems to be more difficult. Many people talk about the **outcome** of a project and do not use the term **output**. What is the reason?

To understand this, let us consider a typical IT project. What is the situation, when the project has been finished? In many, if not all cases at the end of a project there will be a new, extended, modernized or somehow changed technical system. Let us assume, that the project has delivered new application software. This is the result or output of the project, a piece of software, some kind of a machine. Now the customer of the project will take this machine and integrate it in his / her business processes. There it will (hopefully) contribute to the **customer´s business success**, save costs, reduce manpower, increase process speed or revenues, improve productivity or economic efficiency. This effect, which is only generated by using the project result, is the outcome or **success of the project**.

So, there must be a clearly **defined result** of the project, which is delivered to the project customer. Something must have been changed, compared to the point of time, when the project was started. The world must have changed and should have been improved through the project result.

The project result is characterized by some degree of **uniqueness**. Even if we only consider software development projects, every project will be unique, because the result of the project is always different. Nobody will develop a piece of software twice and identically. And if we consider the whole variety of IT projects we will find (see Kuster et al, p8):

- investment projects,
- infrastructure projects,
- product development projects,
- organizational projects,
- organizational development projects,
- information technology projects,
- construction projects.

Other typical project categories in the IT ecosystem are:

- development of new systems,
- migration of systems between platforms, e.g. hardware or operating systems,
- restructuring or maintenance of systems,

- introduction of new systems,
- integration of systems or organizations.

Let us fix:

Every project has to deliver a clearly defined result.

Every project is a somehow unique set of activities.

There is a third characteristic of IT projects. Though the target must be precisely defined, it may change every second. Hence project management has to deal with **moving targets**. Of course, this is also a challenge for project performance management.

Considering these investigations we get a first picture for discussing performance management of IT projects. We have three major stages, related to a project:

- the time and activities **before starting** a project,
- the time and activities, **when running** a project,
- The time and activities **after having finished** a project.

The activities before starting a project are summarized under the term **initiation**, the activities after having finished a project are summarized under the term **evaluation**.

6.2 PROJECT PLANNING

Let us start with an investigation of running projects, of course from the point of view of performance management. Thus let us assume, that a decision has been made to run a specific project. Now it starts with the **planning** of the project (see figure 25). This planning process has a specific logical structure. And as every activity in every project it has to deliver specific results, namely:

- the preparation of the **work breakdown structure** (WBS), which disassembles the total project task into a more or less big set of elementary activities, called work packages: Work packages are defined to be the smallest projectable elements of a project; they are the atoms of the project,
- the **sequence planning** of all work packages: Work packages have to follow a specific order, because in many cases the output of a specific work package is needed by one or more other work packages as input. And the latter can not start till then the

preceding work package is finished and has delivered the needed result. The result of this planning step is a list of all paths / sequences of work packages through the project (see figure 26),

- the **planning of efforts**: Each work package needs input. This input may consist of results of preceding work packages. Also several resources will be consumed to prepare the specific output of each work package. In IT projects we mainly deal with the consumption of human resources. But also a lot of technical resources, e.g. computing power, may be necessary,
- the **planning of deadlines**: Combining the resource need of each work package with available capacities leads to the duration of each work package. Combining these results with the sequence plan leads to the duration of each path of work packages through the project.

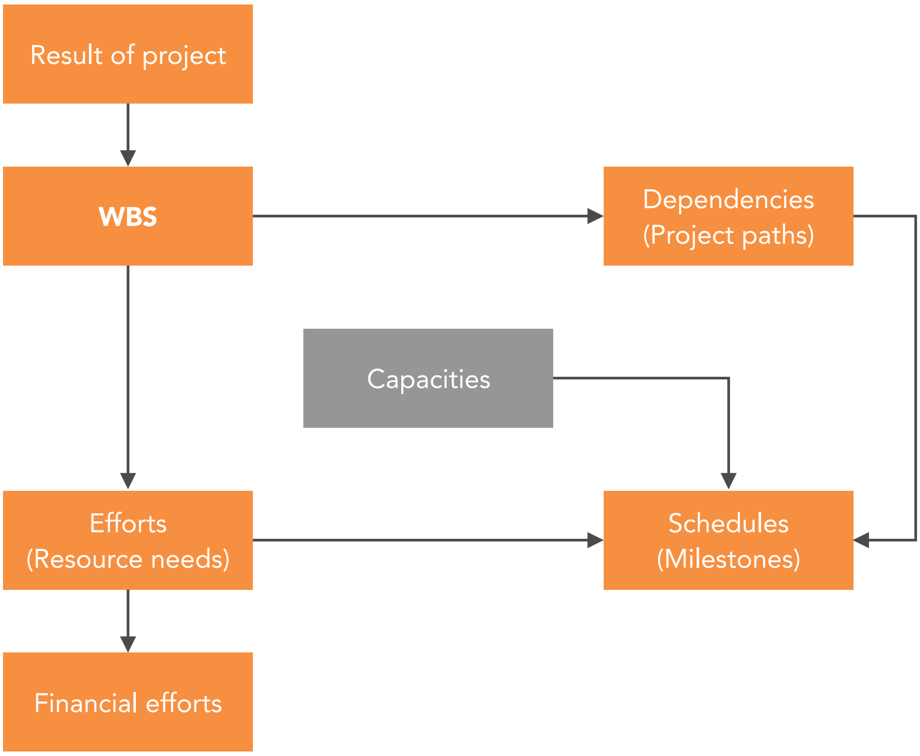


Fig. 25: Project plan

If we consider the duration of each work package path through the project, we will find that there is a maximal value for the total duration of paths. A path, where the duration takes this value, is called a **critical path** of the project. It is possible, that a project has more than one critical path. The duration of the critical path is the minimal duration of the project. Because the critical path must be completely worked through, the project cannot be finished in a shorter time span than the duration of the critical path.

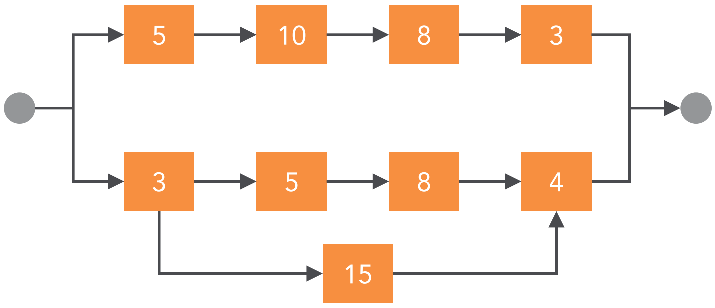


Fig. 26: Activity flow in projects

According to our investigations the project plan consists (at least) of the four mentioned partial plans. Sometimes the reader will find additional plans or planning activities in the books.

One example, which is often mentioned, is the quality plan or **quality assurance plan**. Quality is an important part of the project result. Hence all activities, related to quality, have to be content of the total project plan as well as of the planning activity itself. The quality plan will just be a sub-plan of the total project plan.

6.2.1 CONTINGENCY PLANNING

However, the four-step project planning can be complemented by some specific planning activities. The first activity is adding **buffers** to the project plan. As we have seen, the needed resource quantities combined with the available capacities leads to the planned duration of activities. But there are several **risks** or **threats**. The estimation of the planned resource quantity may be wrong, e.g. too low, or the expected capacity of the considered resource will not be available or only lower capacity will be available. These two phenomena will result in a longer duration of the activity. And if more resources are needed than planned the project will become more expensive. What should the project manager do to overcome these risks?

He should plan some **contingency**, on the one hand due to efforts, on the other hand due to the duration. So he / she will add two types of buffers to the project, **effort buffers** and **time buffers**. We will later understand, that we need such buffers, but they should not be too big. Good project management cares for a proper allocation of both buffer categories.

Time buffers are not only needed to absorb delays in the execution of a project. They are also needed to cover **waiting times**. Let us consider two examples, where we need waiting time buffers.

The first case is typical for **quality management**. If for example a programmer has finished a piece of software, this software has to be **tested**. While this testing is executed the programmer is not allowed to work on that piece of software. If during the testing errors are detected, some **debugging** is necessary, and after the end of those debugging activities some **re-testing** must be conducted. And the subsequent work packages are not allowed to work on that piece of software till it is released by quality assurance of the project.

The second case is the **cooperation with external partners**, who deliver some equipment or software to the project. Usually a delivery date is planned and contracted afterwards. But there may be an unplanned delay in the delivery. If this happens to a critical path the total project will have a delay accordingly. So, a careful project planning will foresee such a delay and plan some waiting time…

At this point we should find out, who is responsible for **buffer planning**. Many organizations and many people think, that buffer planning is not a task for project planning, but a task for the persons in charge for specific work packages (see figure 27). For a long time this has been an established philosophy of project management. The project planner made a plan, and the persons in charge had to deliver their results at the planned **finishing time** of their activity. The result was, that those persons in charge made a local plan but did not communicate it openly to the project planner. They said: If I have to deliver such and such result, then it will take me such working time and such related duration. Both parameters included contingencies, which were not communicated to the project planner. The culture was: I have to deliver my results at a specific point of time, and I (hopefully) will not deliver with delay, but I will also not deliver before the planned point of time. Of course this led to very **inflexible projects**.

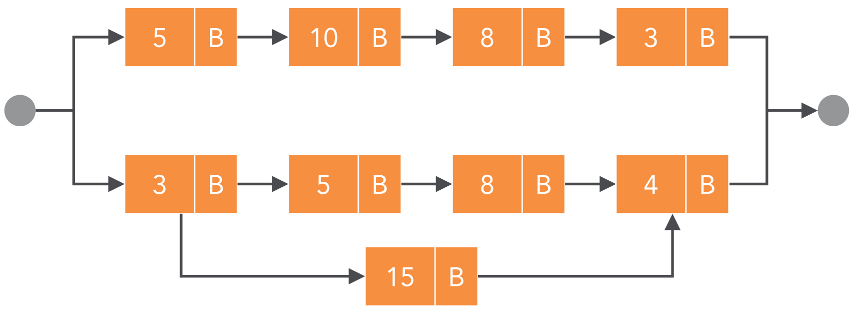


Fig. 27: Activity flow with local buffers

Modern project management has changed this way of thinking. The persons in charge are asked to tell the **most probable effort** and the **most probable duration**. Everybody knows, that things will differ from planning. The actual effort will be higher or lower than the planned effort, the actual duration will be longer or shorter than the planned duration.

Projects, which are planned due to this philosophy, will have a **total effort buffer**, which is planned by the project manager, and also a **total duration buffer**, which is attached at the end of the project, and the project duration can expand into this comprehensive buffer. Experience shows, that projects with such open central buffers are significantly more reliable due to effort and adherence to schedules (see figure 28).

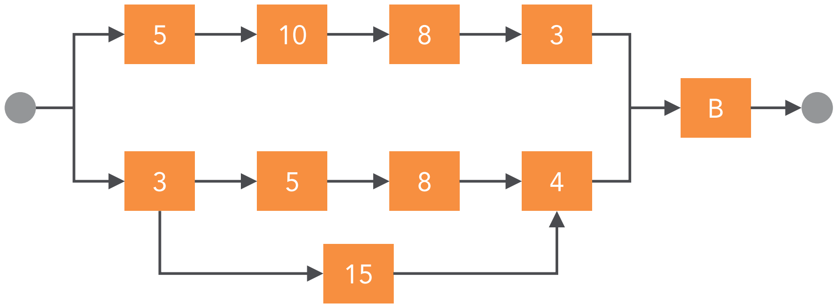


Fig. 28: Activity flow with global buffer

6.2.2 BACKWARD PLANNING

However, there are further challenges for project planning. One usual situation is, that the **final delivery date** for the total project result is given, e.g. due to legal requirements. If then the planning of deadlines comes to the result, that the project as it has been planned, cannot at all be finished within the given time span, then project planning has a serious challenge. Of course will they find ways to reduce the length of the critical path. But sooner or later they will come to the point that they have to **reduce quality**.

Finally, and this is the responsible approach, one should redesign the project resp. the project result. One strategy is to deliver the **final result in two stages**, a basic version first, which is ready at the given deadline and fulfills the strong requirements, and an extended version some time later, which provides all wanted functionalities. Experience shows, that a **staged delivery** of project results is possible in nearly all situations. Of course, it may not be the optimal result due to economic aspects. Delivering a result in two steps may be more expensive than delivering it completely and offhand. But to violate legal requirements may lead to punishments and penalties and thus be less economic than the two-stage delivery.

If there is such a deadline, then planning will be conducted backwards. Normally we start with the beginning, and we look for the end, when we start today or tomorrow. Planning backwards from a given deadline means, that we answer the question, when we should start, if we want to be through at that given deadline.

6.2.3 EFFORT PLANNING

Planning a project has another serious challenge, namely the **estimation of efforts**. Especially to estimate human labor is always a big challenge even if you as a project planner have several years of experience. Why is it so difficult and challenging? One issue is the fact that the needed effort depends on the person who you can appoint to an activity. You have inexperienced and experienced members in your team. Whether somebody is very good or not so good, depends on the task and whether the person, who you have in mind, is talented for this task or not.

But even if you had the normalized and standardized project worker, the estimation of needed efforts would be challenging. The reason is the uniqueness of every project. In your total life you will never see two completely identical work packages, and you are talking about the future, too. You will never know, which situation you will have tomorrow, and which surprises are hidden in the task, you have to plan and to conduct. Books about project management will present you several methods and approaches, and you will find, that they are all helpful in some way. But finally you have to write down a number.

We will not list and discuss all the different **estimation methods**. Behind all those methods there is one principle, and you have to apply it. The general approach goes as follows: If you have to estimate the effort for a specific task or activity, find other tasks, which you have conducted in the past and where you know the final effort. Then compare them with your actually planned task. Is the actual task bigger or smaller than those former tasks? Is it simpler or more complex? Do you expect to complete it with the same effort, higher effort, lower effort? If higher or lower effort, how much, related to the former tasks? If the former task needed 20 person days and you think, that the actual task will need 20% less effort, then your estimation would be 16 person days.

This approach is called **conclusion by analogy**. We are convinced, that this approach is the most general one and can be applied to every project. But what are the prerequisites? The most important prerequisite is, that the project planner is a **very experienced person** and has already done a lot of project planning work. But what can you do, if you have to plan a project for the first time in your life?

Now we come to the point of **knowledge management** in an organization. It would be fine, if your organization had stored all data of formerly finished projects, including efforts or every work package. In this situation you had a project database and could find several work packages, which are similar to your actual work package having to be planned. Many organizations will not have such a database. Are you unable to soundly plan your project now? Usually not. Look around, and you will be able to identify several persons around, who have experience with project management and have successfully worked in IT projects.

Ask them to help you! The most effective way to use their experience is to make a first estimation by yourself, then to present it to them and ask for their response. They will give you an answer, and there will be a discussion. And you will learn quickly from them. Just do it. Project management is a **matter of experience**. This is, of course, also true for project planning. Exercise it! And learn from your own experience.

6.2.4 AGILE PROCEEDING

Now, many readers will have heard from **agile proceeding**, based on the so-called agile software engineering. What is the characteristic of this approach, and what is different to the approach described here?

The logic of project planning, which we described here, started from a given result and asked: How much effort will it cause, and how much time will it need? Agile thinking is different. It starts with a given time span and asks: How much can we achieve in this time span? The symbol of agile thinking is the **iteration** or **sprint**, a given time span. This time span is the rhythm of the project, and the project has to follow that rhythm.

To understand **agile thinking** you have to be clear, where it comes from. It comes from software maintenance, and the paradigm is, that at the end of the given time span the software is up and running again. May be, that the programmer can add more or less new functionality to the software, however, it is essential, that the **software is up and running**.

Many IT projects start from an existing software package and modify it, extend it, improve it. Those projects, of course, can apply the agile approach. But does it work if we develop completely new software? It is very probable, that you are not able to provide running software right from the beginning of the project. Later in the project this may be the case.

But if you reduce the agile approach to the simple formal element, namely the fixed time span, or, as it is called in project management environment, a **time box**, then it could make sense to define such a time box and do project planning from the beginning according to that time box structure. Which advantage would this offer? Which disadvantage could appear?

The advantages of the time-box approach are somehow similar to the advantages of standardized containers in the shipping business. All parts of your project making organization can be **synchronized**. New work packages of all projects start at the same time, and they are also finished at the same time. This makes it easier for management to exchange people, e.g. experts, between the different projects. **Multi-project management** becomes less complex.

However, there seems to be a disadvantage, similar to containers in the shipping business. Sometimes containers are not 100% loaded, or sometimes the goods will not fit into a container. Of course, working in the time-box mode will force the project manager to define work packages, which fit into the time-box structure. It may be, that they have to assign two or three small work packages to a time box. And if they have a task, which is greater than a time box, then there might be a problem. But because we are not in the world of physical goods, the problem is a quite small one. To master over-sized work packages we could combine two sub-sequent work packages… So the time-box approach can be easily applied, if the activities of the project are sized according to the size of the given time box.

The question, which must be answered, is, how to define such a time box. Due to practical experience we recommend to define the length in time as of two weeks and to assign two people to such a time box. Then it will have a capacity of 20 person days, which seems to be a reasonable size. If you now have an organization of 20 project workers, then one year would have (20/2) × (52/2) = 10 × 26 = 260 time boxes, which can be presented in a table with 10 rows and 26 columns. Planning could be done manually, and no complex software is needed (see figure 29).

	Time Boxes								
	1	2	3	4	5	6	7	8	9
Team 1									
Team 2									
Team 3									
Team 4									

Fig. 29: Time Boxes

6.2.5 MILESTONE PLANNING

Finally in our discussion about project planning we would like to introduce another term: milestone. What are **milestones**? A milestone is a point of time, which is very important for project management, a **distinguished deadline**. However, it is a deadline, nothing more. Though deadlines are often assigned to the delivery of specific results, the delivery itself is not the milestone, but the respective deadline.

Each project will at least have two milestones, the starting time and the finishing time. It will have a lot of deadlines, usually the beginning and the ending of every work package. The set of milestones is a sub-set of all deadlines of the project. Projects are often structured into **stages**, and the "natural" milestones are the starting and ending points of those stages. The reader should be aware, that not only the finishing deadlines are important. The starting deadlines are more important! Or did you ever arrive in time, when the train or plane departed with delay?

6.3 MANAGEMENT OF RUNNING PROJECTS

Now let us assume, that we have successfully finished the planning and the project operation has started. How can performance management help to make the specific project successful? See figure 30.

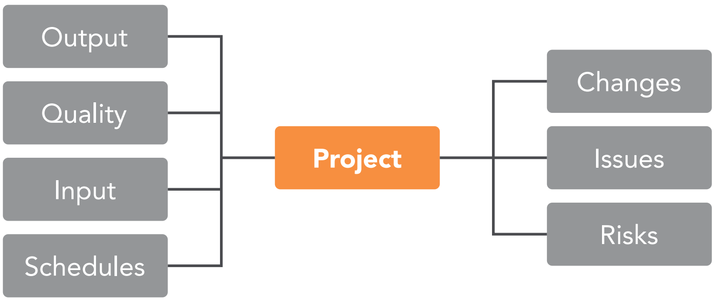


Fig. 30: Project characteristics

6.3.1 OUTPUT MANAGEMENT

Again we can use our well-tried output-input model. What is the output of a project? This seems to be a very hard question because of the uniqueness of each project. However, if we go back to the planning stage, we will easily find a solution. We have defined the **work package** as the smallest projectable entity of a project. So, if we know the total number of the work packages of a project, then the degree of target achievement can be measured by the ratio of the number of finished work packages and the total number of work packages of the project. This is, of course, a rough metric. But we can refine it. First we could consider, that different work packages have different sizes. How can we measure the size of a work package? The usual approach is to measure the size of a work package by the planned labor volume of this work package, and this labor volume is measured in man-days or person-days. Now the target achievement metric would be

$$a = \frac{\sum_{i=1}^{n} v_i \times f_i}{\sum_{i=1}^{n} v_i}$$

with v_i being the number of planned person-days of the work package with the number i, n being the total number of work packages of the project, and f_i being a binary variable associated with the work package i, and $f_i = 1$, if the work package i is finished, and $f_i = 0$ else. If you are very skeptical, then you should ask another question: What about those work packages, which are running for some time, but are not yet finished. There should already be some output, which actually should be valued. Here another traditional approach to measure target achievements can be used. This approach is related to the term **Estimate-to-complete** (ETC). What does this mean? This ETC approach goes as follows:

- Find out, how much labor volume has already been consumed by the considered work package (c = cumulated labor volume).
- Estimate, how much labor volume will still be needed to finish the considered work package (e = estimate to complete).
- The degree of target achievement is $a = c/(c + e)$ ("$c + e$" is often denominated as the **forecast** for this work package).

Now we can modify the formula for the target achievement level as follows:

$$a = \sum_{i=1}^{n} v_i^* \times a_i \text{ with } v_i^* = \frac{v_i}{\sum_{j=1}^{n} v_j}$$

Of course, you could apply the ETC method to the total project considering it as one big piece of work. You then have

$$a = \frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} c_i + e_i}$$

Subsequently we will see another approach to measure the target level achievement of a project, represented by the **Earned Value Management**.

6.3.2 QUALITY MANAGEMENT

But let us consider the second dimension of project output: quality. If we consider the quality dimension carefully we will come to the conclusion, that quality is already included in the target level agreement as we have worked out it before. If there is a bad quality, then the considered work package cannot be finished resp. The ETC parameter will increase, because we need more labor volume to finish the work package.

However, it is quite usual to measure quality or the absence of quality separately. How can this be done? Again we propose to use a somehow naïve approach based on project planning. With project planning a variety of **quality assurance activities** has to be planned. Now the level of quality can be defined as the ratio of the number of quality assurance activities, which have already been conducted and did not find errors, and the number of quality assurance activities, which have already been conducted. This requires an explicit planning of all quality assurance activities, even of those, which are an integral part of work packages, and a consequent monitoring of all quality assurance activities with protocols documenting the detected errors, is needed.

If the project management does not want to plan quality in this way, they could decide to establish an **error log**. Then the number of entries in this error log would be a metric to measure the absence of quality.

6.3.3 INPUT MANAGEMENT

Now let us consider the input of the project. The traditional approach is to measure the **degree of budget consumption**, namely the ratio of the already consumed budget and the total planned budget of the project. But which budget should we consider, the fiscal budget or the labor budget? Finally the performance manager has to consider both, because the project will have a planned labor volume, and it will also have a planned fiscal budget.

For managing the daily operation of the project the **measurement of labor consumption** is more helpful, and it strongly correlates to the measurement of target achievement. So the labor consumption level will be defined as the ratio of the already consumed labor volume and the planned total labor volume of the project. This may exceed the value 1…

We will measure the fiscal budget consumption similarly as the ratio of the already consumed fiscal budget and the planned total budget of the project.

The reader should be aware of the different behavior of both budget consumption metrics. The labor consumption will continuously increase while the project is running. The fiscal budget consumption will go up by leaps and bounds. The reason for this erratic behavior is, that the fiscal budget includes payments for the purchase of hardware and software or invoices sent in by external service providers. When invoices are paid big volumes of money will leave the fiscal budget of the project.

6.3.4 SCHEDULES MANAGEMENT

Now, as we have considered and measured the output and the input of a project there is a third dimension to be measured, the behavior of the project in time. We have to consider the **adherence to schedules**. This measurement is to be based on due dates. Which due dates do we have in a project? Again we go back to planning and will find, that we have planned a lot of due dates, namely at least two of them for each work package, the starting time and the ending time. So we easily find a first candidate for measuring the degree of adherence to schedules by the ratio of all due dates, which have already been met and have been met as planned, and the number of all due dates, which have already been met.

In the mean time the reader should have become sensitive to performance management thinking, and he / she should ask two questions:

- If we do not meet a deadline, does it matter, how big the difference between the planned and the actual time is?
- And is it always a good result, if we meet a deadline earlier as planned?

Those questions are difficultly to answer, and whether we have to measure the degree of the deviation from the planned deadlines, may depend on a specific project. However, due to the critical path challenge and the contracted delivery dates it will have priority to measure delays.

Thus a **delay metric** could be the ratio of all due dates, which have been met, but with a delay of more than x days, and all due dates, which have already been met. Another approach focuses on the critical path and works with the ratio of the forecasted duration of the project and the planned duration of the project. And finally performance management should have an eye on those due dates, which are met significantly before planned. The metric could be the ratio of the number of due dates, which have been met more than y days before schedule, and the number of due dates, which have been met without delay. Why should performance management consider those "early birds". They could be signals for bad resp. too pessimistic planning. They also could be a signal for hidden quality problems or a misunderstanding of customer requirements.

6.4 ENVIRONMENTAL CHARACTERISTICS OF PROJECTS

Now we should ask, whether the discussed information is sufficient to manage a project. When thinking about this, we will come to the conclusion, that it would be helpful for the project manager to have some more information. Due to experience there are three aspects of project management, which are important:

- the volume of changes,
- the volume of open issues,
- the volume of threats or risks.

The perspectives, which we have considered up to now, represent targets, which the project manager has to achieve with the project. He / she has to care for target achievement, good quality, keeping to budgets and schedules. The three perspectives of changes, open issues, and threats, cannot be directly controlled by the project manager. Those subjects are generated from the environment of the project. They are like weather, and the project manager has to register it and has to master it.

6.4.1 CHANGE MANAGEMENT

Changes are related to the phenomenon of **moving targets**. If a project runs for some time, a lot of changes will go on in the business, and this causes new or modified requirements to the project. The project itself is some kind of a learning organization. Customer and user representatives as well as technicians will learn, that things cannot be realized identically as they have been planned. They will discover issues, which were not foreseen during the initiation and the planning of the project. External **goods** and **services** will change, technologies will go on. The **legal environment** may also change, sometimes even dramatically. And last but not least it may turn out, that involved people had not worked as carefully as they should have worked. All those effects will lead to **change requests**. Documents will be prepared and sent to the project manager, and those documents will ask for changes and extensions in the project results. If the project manager would accept the change requests, the work to be done would increase (usually). It is quite improbable, that the scope or functionality of the project result is reduced. This must have effects on **budgets** and **schedules**. So there must be a decision to accept or refuse a change request, and this decision must be made by the **steering committee**. Only this body is allowed to change the scope, the budget, and the schedule of the project.

If we shortly review these thoughts, we easily see, that each change request needs a **business case**, though it may be a small business case. The change must be **economically attractive** for the organization, hence the change request must include some kind of **ROI consideration**, either isolated or as a modified business case for the total project. It also has to describe the effects on budgets (increase!) and schedules (shift of due dates and milestones). Hence each business case is obliged a modified and updated project plan. In addition, incoming change requests must be reviewed by the project manager, and the steering committee will ask him / her for a reasonable **recommendation**. The original project plan must contain budget and time for handling change requests.

It is clear, that the project will change significantly, if there is a high number of changes, which are accepted by the steering committee. And the number or volume of changes will help the project manager to explain to all parties involved, why the budget goes up, and the project takes more and more time. So he / she should **measure the change volume**. But what would be the best metric here?

We think, that one basic metric is just the **number of change requests**, which have come in since the beginning of the project. Of course, one could ask for a weighted number of change requests with the weights being the person-days of work, which will be caused from that request. However, when a request comes in, it will not always be clear, which effects it will have on the project´s budget.

A second metric could make sense, namely the portion of change requests, which have been accepted and led to a "Move" of the project.

If the number of change requests coming in per reporting period goes up or down, this may indicate coming changes in the project environment. And if the portion of accepted change requests goes up or down, this also indicates, that the project environment is going to change or a lot of trouble is coming up. This metric works as an **early-warning indicator** for the project.

6.4.2 OPEN-ISSUES MANAGEMENT

Now let us consider the second environmental characteristic, the open issues. During a running project a lot of **questions** arise, and those questions have **to be answered**. Otherwise the project runs the risk to fail. Those questions represent open issues. They will come up during the running project work, they will be stated in project meetings, and they should be documented in an **issue log**. If you establish such a log, a simple metric to measure the open issue status is just the number of the (actually) **unresolved issues**. If this number goes down, the project is able to master its troubles. If the number goes up, then the project runs the risk to fail again. If there are too many open issues, the project will not longer be able to proceed. So this indicator measures the **problem solving capability** of the project.

Another approach to measure the problem solving capability of the project is the number of issues, which are longer in the open issue list than a given time span. This metric assumes, that it doesn´t matter, how much open issues have been detected, if they are resolved within an given time span.

6.4.3 RISK MANAGEMENT

And finally we consider the threats or the risks of a project. Of course should a project have a risk-management. Some authors even say, that project management is a synonym for risk management, or vice versa. However, threats can lead to a failure of the project, and project management has to identify threats and manage the risk, caused by those threats.

How can and should performance management measure risks?

If we follow traditional approaches, then risks have a **financial value**, calculated from the multiplication of the **occurrence probability** with the **expected amount of loss**. Mathematically more correct is the risk the expectation value of a random variable "amount of loss". This sounds convincing, but is not. How can you find out the probability distribution of the amount of loss, if this parameter can take any value between 0 and ∞ (infinity).

For a project manager it is much easier just to write a list of the identified and still lasting risks. Then the risk of your project is just the **number of the actually documented threats of the project**. Yes, this is a very rough metric, but it is much better than to have nothing. Of course can we refine this metric by introducing a weight for every risk. With such a weight we would be able to identify changes of single risks. But we have some additional effort. We have to conduct a weighting of all identified risks, and we have to redo it periodically.

We should not underestimate the **monitoring effort** of those simple metrics. If the reader would follow our given recommendations, he / she would need appropriate processes to **identify** and **document** all incoming change requests, all emerging open issues, and all identified risks. And it is not enough to do this once. It must be redone **periodically** to be able to detect changes. If you do not already have those processes, you will have to define them, document them, train the people in the project and around the project, and finally you have to ensure, that everybody involved sticks to the established processes. This can be a cumbersome and challenging task.

Finally the question rises, whether the considered seven aspects of a project are the right ones. Are there better aspects? Do we need more? There is no clear answer. It depends on the specific project, more on the specific environment and its **project management culture**. We think, that this set of characteristics is very reasonable and can help effectively in nearly every project.

6.5 EARNED VALUE MANAGEMENT

But now let us close our considerations with a short look on the so-called **Earned Value Management** (see Fleming 2000). In the meantime this approach has become well known all over the world. The fascination from this approach comes from the fact, that it seems possible to manage a project with just two indicators. So let us investigate the method. The key term is the earned value of the project. What does this mean exactly?

To understand the concept of the earned value, we have to go back to project planning. The essence of project planning is the distribution of activities and thus a **consumption of budgets** on the time axis. Now we can start at the beginning of the project and run along the time axis until the (planned) end of the project. Now let us consider the **cumulated budget consumption**, which is 0 at the beginning of the project and the total planned budget at the end of the project. If we take a coordinate plane with x- and y-axis, the x will be the time, and y will be the cumulated consumed budget according to the time: $y = y(x)$. This curve is monotonously not decreasing: $y(x_2) \geq y(x_1)$, if $x_2 \geq x_1$. Now let us consider a specific point of time t_0. At this time the planned cumulated budget consumption will be $y(t_0)$. However, the actually consumed cumulated budget may be different. Let us denominate it with $z(t_0)$. But now be careful! $z(t_0) > y(t_0)$ does not automatically mean, that the project performance is bad, and vice versa $z(t_0) < y(t_0)$ does not automatically mean, that the project performance is excellent. Here a third entity comes in, the **earned value** of the project. This earned value is based on an evaluation of the already accomplished project work, and this **level of accomplishment** is measured by that consumed budget, which due to the project planning would have been assigned to this specific level of accomplishment. This earned value, which we denominate with $v(t_0)$, has to be considered together with $y(t_0)$ and $z(t_0)$. With the help of those three parameters two indicators are created:

$$CPI(t_0) = \frac{v(t_0)}{z(t_0)} \text{ and } SPI(t_0) = \frac{v(t_0)}{y(t_0)}.$$

CPI is an abbreviation and stands for **cost performance index**. SPI similarly stands for **schedule performance index**. How do we have to interpret those indicators?

If $CPI(t_0) = 1$, which means $v(t_0) = z(t_0)$, then the value of the accomplished project results is the same as the really consumed budget. This is, what project management wants to achieve. If $CPI(t_0) > 1$, then the project has created more value than it consumed budget. It has accomplished its intermediate results with less budget than planned. Otherwise $CPI(t_0) < 1$ shows a bad cost performance of the project. To reach those specific results, they have spent more money than planned.

Let us now consider the SPI, which is more difficult to interpret than CPI. If $SPI(t_0) = 1$, which is equivalent to $v(t_0) = y(t_0)$, then the earned value has come just in time. In t_0 it had been planned with the value of $y(t_0)$, and the project has realized this. If now $SPI(t_0) < 1$, this means, that the actual earned value is smaller than the planned earned value for this point of time. The smaller value had been planned for an earlier point of time. So with $SPI(t_0) < 1$ the project is behind its schedule. Vice versa, $SPI(t_0) > 1$ signals a schedule performance, which is better than planned.

The Earned Value Management is nothing but the transfer of the **standard costing** approach to project management (see chapter 3). $y(t_0)$ is the planned budget consumption, $z(t_0)$ the actual budget consumption, and $v(t_0)$ represents the targeted budget consumption for the accomplished intermediate project results.

What are the advantages of the earned value management compared to the **magic triangle** of project management or the **devil´s square** of project management?

The major advantage is, that the method is **well known worldwide**. The disadvantage is, that the concept of earned value somehow is quite abstract, and people sometimes have to try hard to understand the philosophy behind the method.

The difference to our approach is not so big as it seems to be. Here we need the planed budget consumption, the actual budget consumption, and the budget consumption, which we have assigned to the already achieved results. The aspect of quality is integrated into the intermediate results or the earned value. If quality is bad, then the earned value is low(er). Results are only then valued positively, if quality is on the planned level.

Our first approach came from the actually accomplished results of the project, the earned value management starts at the given point of time. In our approach the time is valued, based on the actual target level achievement. Here target level achievement is valued, based on the actual point of time resp. the planned earned value, which has been planned for this point of time.

6.6 ATTRACTIVENESS MEASUREMENT

Now, that we have learned something about the performance management of projects already running, we still have to consider the **initiation** of projects and how projects can come through this preliminary stage successfully. In this initiation stage the organization finds out, whether it will run the project or not, whether it is attractive for the organization or not. From the point of view of performance management we want to know, how we can measure the **attractiveness** of a project. To do this we can use the well-tried concept of **return on investment** (ROI). What does this mean?

The basic idea of this concept is well known. Assume, that you have got some money, and you want that money make the world go round. You can lend this money to a bank (or any other organization). And you will expect two beneficial effects. First you expect (or hope) to get your money back after an agreed time span. And you secondly expect to get some additional money back. This amount of additional money is the interest on the given amount of money. Hence, there is the following mechanism: Give an amount of money S to another party, wait for some time, namely n time intervals, and then get back another amount of money E with $E > S$. The rate of return is $R = \frac{E}{S}$, and you expect $R > 1$.

To measure the attractiveness of a project we apply exactly this method. To run a project we need some money to buy hardware and software, pay the salaries of people, working in that project, and pay the invoices of external service suppliers and consultants. This money must be borrowed from our organization resp. the owner of our organization. They will give us the needed money only, if the project guarantees or promises, that after some given or agreed time the **project pays back** so much money, that the organization resp. its owner is interested in making this investment. The starting capital S must be found out by the project initiator, precisely by the project customer, because he / she is looking for the project result to improve his / her business processes. But how can we come to the final capital E?

Conceptually this is quite simple. Let us assume, that the project result will be an application system. This is something like a machine. When running it, some **operating efforts** come up, and the organization must pay for it. On the other hand, with the help of this new application system some **benefits** will be generated, e.g. some parts of a business process can be automated, the number of employees in this process can be reduced, and the benefits are the savings of salaries and corresponding non-wage labor costs. So we have for the returning capital $E = B - X$ with the financial value B of the gross benefits and the operating expenses X. Thus we have , where all parameters are given in **currency units**. $R = \frac{B-X}{S}$ is the metric of the project attractiveness, which we have looked for. It looks a very easy approach, and from a mathematical point of view it really is. However, getting the figures is hard work. What are the challenges? Let us consider the model step-by-step.

6.6.1 INVESTIGATION OF STARTING CAPITAL

First we have to find out, how much money the project will need. The first answer will be: Look to the project planning. You will get this figure from the **budget planning**. But here we have the first problem. The project planning will be conducted after the decision to run the project has been made. But we are in the initiation stage of the project now, and nothing has been planned nor will it be planned now. So there must be a first and rough estimation of the financial project budget before having planned the project in detail. But if with this estimated value S the project will be allowed to be run, then top management will not accept, that the detailed project planning comes to another project volume S^* with $S^* > S$. If we are pessimistic, then R will be lower, and the project may become not attractive enough for top management. So the estimation of S must be **optimistic** enough to make the project attractive and **reasonable** enough to minimize the risk of a budget overrun.

6.6.2 INVESTIGATION OF OPERATING EXPENSES

We also have to find out the amount of operating expenses to be expected. If the found value of X is low, then R will go up, and if it is high, then R will go down. The challenge is to make a sound and reasonable estimation. The main risk here is, that some effects are forgotten totally. Here we are talking about IT projects, and the IT part of the operating expenses has to be delivered from the IT experts. The main parts are the technical infrastructure (hardware), which is needed to run the new system, license fees, if purchased software is used, and labor volume to maintain and administer the system in the daily operation.

But there may also be operating efforts in the customer´s organization. Think of new employees to administer master data or be responsible for scheduling or coordination activities, the check and evaluate the work, which has been done with the new application system, etc. Those efforts are often forgotten. The data must be delivered by the customer organization. Because IT performance management will be involved in calculating the project attractiveness, they have to care for the involvement of the customer´s performance management, too.

6.6.3 INVESTIGATION OF THE (GROSS) BENEFITS

Due to the above given formula we have to find a financial value to measure the benefits, which are caused by the project results. The beneficiary of the project, the project customer, has to deliver these data. The first problem is to come to a realistic value. The risk is to overestimate the impact of the project, which generates a too high value of B and subsequently a too high value of R. The second problem is, that we often see a lot of benefits, but feel unable to assign a financial value to them. Let us consider some specific situations:

- The most comfortable situation is that the project results cause **cost reductions** in the customer´s organization. Then B is the financial volume being saved.

- If the execution of business activities can be accelerated then this is a benefit, of course. But what is its financial value? The internal effect of speeding up business processes is, that capacities to be provided for the business processes can be reduced. And this **reduction of technical capacities and financial means**, which are bound in the business operation, can be evaluated. Usually this is accompanied by **increased customer satisfaction** and / or **better competitiveness**.
- As we have just seen, there are often benefits, which can be described clearly, but we feel uncomfortable to assign a financial value to it. What is the financial value of customer satisfaction or improved competitiveness? We could use very sophisticated methods to calculate the financial value of it, because higher customer satisfaction should lead to more sales, and increased sales will improve out profit. However, those approaches are not so easy from a mathematical point of view, and they include very **serious assumptions**, e.g. probabilities of sales increase, etc. We will describe a general and very pragmatic concept for dealing with this kind of benefits later.
- We sometimes want to conduct projects to **avoid negative impacts** on our business. A typical category of this type of benefits is legal requirements, which very often lead to new systems or changes of existing systems. The benefit of those projects is the **avoidance of punishment and reputation damages**. If we do not act according to laws we will be punished by court decisions. However, if our business partners know that we have problems to run our business according to law they will refuse to be our business partners in the future. Our suppliers will end the relationship with our organization, and our customers will buy from our competitors. Again we have the challenge to find a financial value for this category of business impact.

What can we do to find or estimate a financial value for benefits, which are originally not of financial value or will cause financial impact for our organization in a very complicated and non-deterministic way. We can construct specific models to calculate that. But this would go far beyond the scope of this book.

However, we can present a very general approach here. Let us get back to the formula $R = \frac{B-X}{S}$. We assume, that our organization has a decision rule, that each project has to deliver a minimal rate of return R_0. They ask for $R > R_0$. We now see, that this means $\frac{B-X}{S} > R_0$. With the assumption of $S > 0$ this inequality can be transformed to $B > X + S \times R_0$. What is the advantage of this formula? R_0 is given in our organization, and S as well as X can be evaluated financially. Now the responsible person for initiating the project has to answer the question: Why should top management think, that the financial equivalent of the project´s benefit is greater than $X + S \times R_0$? Or to put it in other words: If there were another project with S and X, but a financial benefit of B_1, such that $R_1 = \frac{B_1-X}{S} > R_0$, would he / she surrender the financial benefit B_1 and value the non-financial benefit of the considered project higher than B_1?

We have seen that the concept of ROI is very general and can be used to measure the attractiveness of projects. The application of this concept has two steps. The first step is, that the project has to proof, that its rate of return will be better than a given lower limit R_0.

6.6.4 PORTFOLIO ISSUES

But then a question will rise. If the organization has so much projects with a rate of return exceeding the minimal value of R_0, that the organization does not have the resources to run all attractive projects, then they have to select the **most attractive projects** to be conducted. To do this, they will set an overall project budget of *P*, and then all attractive projects (exceeding R_0) will be sorted in decreasing order of *R*. If S_i denominates the budgets of those projects, the organization will select the first N projects such that $\sum_{i=1}^{N} S_i \leq P$ (The reader, who is familiar with optimization issues, will see a (theoretically) more satisfying approach, but here we only present a very simple heuristic approach to make ideas clear to the reader). This selection of the most attractive projects, which can be financed by the overall project budget P, is called the **project portfolio process**.

Finally performance management has always to consider the **total project portfolio**. If something has to be changed in one project, this will have an impact on the total portfolio. If the budget need for one specific project goes up significantly, the impact can be, that another project has to be eliminated from the portfolio. However, portfolio performance management is very fascinating and challenging, but to investigate it reasonably, we would need another book… So we will not consider this subject here in more detail.

6.7 EVALUATION OF PROJECTS

At the end of our considerations with respect to project performance management we have to investigate the evaluation stage of a project, which starts, when the project is finished and the operation of the project results has been started. Is there a task for performance management? The answer is: yes.

Before we started the project, we calculated the (expected) rate of return. Now, after finishing the project we have to find out the **actual rate of return**.

Immediately after finishing the project we will only have the actual value of the consumed project budget. The planned values have been S_1, X_1 and B_1, leading to R_1, which was the basis to decide for this project positively. Now, after having finished the project, we have an actual value S_2, actual estimations B_2 and X_2, and this will give an actual estimation of R_2 of the rate of return. Hopefully it will turn out that $R_2 \geq R_1$. Even $R_1 > R_2 \geq R_0$ will be an acceptable result. But if it turns out, that $R_2 < R_0$, then there will be an issue…

Some time after having finished the project there should be another evaluation, some kind of review of the project attractiveness. We will have $S_3 = S_2$, but B and X will probably have changed again due to new forecasts. E.g. a software supplier could have changed his license fees…

We have not yet discussed the **time frame** T_1, on which we based the attractiveness measure R. Usually organizations expect, that the expected rate of return is realized within a given time span. For IT projects we see a usual **operating time horizon of 3 years**. If we add the time for the execution of the project, this time span has to be extended up to 4 years (roughly spoken). If the organization expects an annual interest rate of 15%, then they will expect a rate of return of at least 160% for a period of 4 years.

This given time span rises two questions. There may be big projects, which need more time to realize an attractive rate of return, or where we know, that the system, which is going to be realized by the project, will be much longer in use than 3 years, maybe 5 or 8 years for example. How can we **compare projects with different operating horizons**? We can do it by a small modification of the metric, we have used up to now. We replace it by $ROI = \frac{B-X}{S \times T_1}$. *ROI* stands for return on investment. If a project with a minimal rate of return of 160% in 4 years is considered, then the *ROI* of this project is 40%. That means, we expect, that an average of 40% of the invested capital flows back every year. With this approach we can compare projects with different values of T.

There remains one issue. How should we consider projects, where we know, that the results will be used forever? E.g. the use of an ERP system will normally not be stopped after 5 or 8 years. It will run “forever”. But there will be subsequent and sometimes big projects to migrate the system to new technology platforms, to replace the system, because the software supplier has completely re-developed it, e.g. on the basis of a modern programming language, new database technology, etc. Do such projects, which must be conducted, have an attractive return on investment? This question is not easy to answer, but its investigation would also go beyond the scope of this introductory book.

6.8 DISCOUNTED CASH-FLOWS

Let us close with a final remark. The reader, who is familiar with **investment appraisals**, will miss, that we have presented the attractiveness evaluation based on **discounted cash flows**. The logical structure is the same, only the values, which are fed into the formula for *R*, have to be modified accordingly. And the formula for *ROI* has to be modified, too. But we skipped the discounted cash flow consideration here for two reasons.

First reason is the number of pages of this book, which was allowed by the publisher. And the second reason is, that due to the random nature of the values, which we have to deal with here, **discounting effects** for time horizons of 4 years are so small that they are below the variation of the input values. Remember: The result of the attractiveness calculation of a project is just a proposal for a management decision. If $R - R_0$ is small, then we should be aware of the fact, that a slight modification of the input values could lead to $R - R_0 < 0$. It is up to the management to allow such a project to enter the portfolio process or not. The formula or algorithm is just a tool to help management, not to replace management.

6.9 EXERCISES

For information to answer the questions and to prepare for the final examination see chapter 10.6!

6.9.1 QUESTIONS FOR YOUR SELF-STUDY

Q6.1: Describe the difference between the "traditional" and the agile approach to software engineering.

Q6.2: If you find out, that the project duration is longer than the available time, what can you do to fulfill the need of restricted available time, e.g. given by legal requirements?

Q6.3: How would you handle the buffer issue in your projects?

Q6.4: Could project performance management by earned value management be better than our 7-characteristcs-approach, which we have presented in this book? Make up your mind accordingly.

Q6.5: What should we do to improve the evaluation of projects?

6.9.2 PREPARATION FOR FINAL EXAMINATION

T6.1: What are the logical steps of project planning? Describe the results of each planning step!

T6.2: What is meant with the term “backward planning”? Describe it by using the logical steps of project planning.

T6.3: How do we measure the attractiveness of projects? Which input data do we need?

T6.4: What is the critical path of a project? Is the critical path unique for a specific project? Can it change over time?

T6.5: Describe the 7 characteristics to control a running project.

6.9.3 HOMEWORK

H6.1: Find out different approaches to measure the project attractiveness.

H6.2: How could you deal with the fact, that input data for project attractiveness calculation are imprecise and uncertain?

H6.3: How does project management in the various project management models deal with the challenges of change, open-issues, and risk management?

7 SECURITY, RISK AND COMPLIANCE MANAGEMENT

Learning objectives

In this chapter you will learn,

- what security, risk and compliance are in an IT environment, and what is its meaning for an IT manager,
- how we can measure the levels of security, risk and compliance in an IT organization,
- what business continuity is and how it should be managed,
- how we can measure the level of business continuity, which an organization has achieved,
- who is responsible for business continuity.

Recommended pre-reading

- Hayden 2010

7.1 DEFINITION OF TERMS

7.1.1 SECURITY

The discussion about security has increased in the area of IT management. But what is security? In general security is understood as the **degree of protection** against danger, damage, loss, and crime. Obviously security is not a digital property of something and thus can take the values 0 or 1. This means, that there is either no security or total security. An object, which is considered under the aspect of security, obviously can have different and changing levels of security.

In the area of IT the term "security" is pre-fixed with the term "IT" as well as with the term "information". So we are talking about **IT security** as well as about **information security**. Is there a difference? There must be a difference. Otherwise we would not use different terms. Thus let us start with two definitions.

IT security is the protection of IT systems from theft or damage, including the information in those systems, as well as protection from disruption or misdirection of the services they provide.

Information security is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction of information. It is a general term, that can be used regardless of the form of the data may take.

Thus information security includes IT security, because a major part of the information in modern organizations is stored in and processed with IT systems. However, the IT manager can only be responsible for the part of IT security. There should be an **information manager** having the overall responsibility for the organization´s information assets.

7.1.2 RISK

The second management object, which we have to consider here, is risk. We define it as follows:

Risk is the possibility of loss or injury or, due to ISO 31.000, the effect of uncertainty on objectives.

Due to this definition risk management is described as the identification, assessment, and prioritization of risks, followed by a coordinated and economical application of resources to minimize, monitor, and control the probability and / or impact of unfortunate events or to maximize the realization of opportunities.

There is a strong relationship between risk and (information or IT) security, because there are a lot of **threats**, which lead to security risks. However, we also have risks in IT, which do not have a (direct) relationship to (information or IT) security.

7.1.3 COMPLIANCE

Now let us consider the third topic and initially make a definition.

Compliance means conforming to a rule, such as a specification, policy, standard, or law.

More specifically we discuss about **regulatory compliance** sometimes, which describes the goal, that organizations aspire to achieve in their efforts to ensure, that they are aware of and take steps to comply with relevant laws, policies and regulations.

Again there is a direct relationship to risks, because each organization runs the **risk to be incompliant**, especially, if we consider regulatory compliance, then to be not compliant may lead to punishment.

7.2 MEASUREMENT OF SECURITY

Now let us consider, how we can measure the presence or absence of IT resp. information security. To find appropriate approaches to IT security management, let us consider the objectives (see NIST SP 800-160 volume 1, p. 230):

- **Confidentiality:** Rules that govern access to, operations on, and disclosure of system elements (including, but not limited to, data and information). While confidentiality policy typically is considered in terms of information and data, it also applies to restrictions on the knowledge of and the use of system functions and processes;
- **Integrity:** Rules that govern the modification and destruction system elements (including, but not limited to, data and information) and that govern the manner in which system elements can be manipulated;
- **Availability:** Rules that govern the presence, accessibility, readiness, and continuity of service of system elements (to include, but not limited to, data and information).

Other terms, which are often denominated, are

- **authenticity**, which means the realness or credibility of an object / subject, which is verifiable,
- **obligation**, which means, that a transaction is binding, if the executing subject is not able to disclaim the transaction afterwards,
- **authorization**, which is the power and right to conduct an activity.

The above mentioned publication also states (see p. 15): „Systems security is an emergent property of the system. This means that system security results from many things coming together to produce a state or condition that is free from asset loss and the resulting loss consequences. In addition, system security is rarely defined in its own context. Rather, system security is typically defined in the context of stakeholder concerns driven by business or mission needs or operational and performance objectives. System security may also be defined in the context of other emergent system properties including, for example, agility, maintainability, reliability, resilience, safety, scalability, and survivability."

And also (see p221): „Providing adequate security in a system is inherently a system design problem. It is achieved only through sound, purposeful engineering informed by the specialty discipline of systems security engineering [Ware70]. Having established an understanding of the basic need for protection across all contributing perspectives, the protection needs are then satisfied by the employment of specific *security functions*55 that are deemed adequate to protect the stakeholder and system assets. The security functions represent the security-relevant portions of the system and the security-relevant aspects of the systems engineering effort. The selection of security functions is informed by adversity in the form of the disruptions, hazards, and threats anticipated across all stages of the system life cycle, stakeholder risk, and asset loss tolerance."

Are we able to measure those issues positively and on a day-to-day basis? It seems, that security is a topic, where it is not possible to measure the presence of it. We only have a chance to **measure the absence of security** – more or less.

So we can define **security events** and then automatically monitor, if and when such security events occur. Such events could be:

- a failed access to a system,
- an external attack to our systems,
- a checkpoint in our automated processing notes an irregular action or some incorrect data.

Here two indicators seem to be reasonable:

- the number of security events, which have been detected (automatically) in the reporting period,
- the number of security events, which have been detected (in the reporting period or earlier), which have not yet been analyzed properly, that means, that the problem behind the event has not yet been resolved.

The first indicator shows, how much security events come in every month, and whether the security situation is stable or changing. The second indicator shows the ability of the organization to master the security events in a timely manner. If the number of entries in this list increases, then security management may have a competence or a capacity problem.

Another indicator for a bad security status could be

- the number of incidents, clearly or potentially caused by security violations, and being generated in the reporting period,
- the number of incidents, clearly or potentially caused by security violations, and which are still not resolved.

The selection is the same as with the events.

Another source for the proof of absence of security are **periodic or specific reviews**. Those reviews deliver **findings**. And now similar to events and incidents we can define another two indicators:

- the number of security findings through audits or reviews during the last reporting period,
- the number of security findings, which have not yet been resolved.

Finally it could be interesting to ask for the **costs of IT security**. Like for quality the security costs of an IT organization would be 100% of the IT costs, because all activities of an IT organization contribute to good or better security as they also contribute to good or better quality. So we should ask for **IT security management costs** and define

- the portion of costs for IT security management with respect to total IT costs.

Due to the security control structure given in NIST SP 800-53 rev4 (p9) this includes activities in the following areas:

- access control,
- awareness and training,
- audit and accountability,
- security assessment and authorization,
- configuration management,
- contingency planning,
- identification and authentication,
- incident response,
- maintenance,
- media protection,
- physical and environmental protection,
- planning,
- personnel security,
- risk assessment,
- system and services acquisition,
- system and communications protection,

- system and information integrity,
- program management.

7.2.1 RETURN ON SECURITY INVESTMENT

Finally we should take a short look on a well-known indicator, named **ROSI** (return on security investment). It is $ROSI = R - ALE$ with the actual risk value R and the annual loss expectancy $ALE = R - E + T$, where $R - E$ is the risk value after running the investment and T is the financial effort resulting from the security investment.

Thus $ROSI$ is not a security specific ROI as we have defined it for IT projects (see chapter 6), but the net present value $B = N - K - P$ with $ROSI \approx B$, $E \approx N$, and $T \approx K + P$. We recommend to value every investment in an IT organization with the same attractiveness metric and not to introduce separate metrics for the security area.

7.3 RISK MANAGEMENT

Now let us consider the measurement of risk. Behind every risk there is (at least) one **threat**. And then we would like to make an assessment of each risk due to the potential **risk management strategies**:

- Eliminate,
- Reduce,
- Transfer,
- Accept.

So we do not only want to know the number of threats, but also the „value“ of the corresponding risks. We need to conduct an assessment about what is possibly happening and what will be the potential **impact**. By analyzing the relationship between threats and risks we will come to the conclusion, that there is a complex **m:n relationship**, that the composed compound damage of two risks may be higher or even lower than the sum of the damages, which we would have, if each risk would occur separately. What can we do to measure this appropriately? Our disappointing answer is, that you can do nothing if you do not want to develop complicated mathematical models, which nobody of your addressees will (want to) understand.

What can we do, if we want to keep it stupid and simple (due to the KISS principle)? We should simply build the following metrics:

- Number of threats, which have been identified,
- Number of risks, which have been identified,
- Portion of risks, which have been qualified as „high“ or „business critical“.

Those very simple metrics will help management to improve **effectiveness** of risk management significantly.

If the number of entries in the **threat log** changes, you will see, whether the danger to your business is stable, or increasing, or decreasing. This is a clear signal that the environment in the role of being **an enemy to your organization** is changing or not. On the other hand, if the number of entries in the **risk log** changes, then you have a signal on the impact of threats to your organization. And the third measure shows, where you have to put your focus on.

However, there is a **problem of monitoring** again. How do those entries enter or leave the log? And how is the identification of „high“ risks done? This will not happen automatically. People in your organization will have to maintain those lists. May be that you establish a special **risk management group** or that all members of your IT management are included there. This expert group should come together periodically and review the threat and risk status. Additionally you could establish a process, in which every member of your organization can create a **threat or risk ticket** and send it to that risk management group, so that they have to check and evaluate it. Experience shows that **multi-expert** assessments deliver sound and reasonable results.

It may also happen, that in some **periodic or one-time reviews** findings due to threats and risks are made. Those findings should be included in the mentioned logs.

Due to the above mentioned risk management strategies there should be activities to eliminate or reduce risks. This will lead to another indicator:

- Number of risk projects not being finished,
- Portion of risk projects, which have not yet been started.

You can be sure, that those indicators will have the attention of top IT management and C-level management of your organization as well.

7.4 COMPLIANCE MANAGEMENT

Finally let us consider the compliance issue. Again we first have to decide, whether we are able or want to measure the **presence or absence of compliance**. Again it seems much easier from a monitoring point of view to identify lacks of compliance. Similar to security you will have several automatic controls built into your IT systems. And this leads to the following metrics:

- Number of compliance events, which have occurred in the reporting period,
- Number of compliance events, which are registered, but have not yet been resolved.

Similarly you may have established a system or process to create **compliance tickets**. Then similarly to security we have the indicators:

- Number of compliance tickets, which have been created during the reporting period,
- Number of compliance tickets, which have been stored, but not yet been resolved.

This compliance ticket process should have an interface to your business partner, because sometimes they will find a compliance deviation and due to your reputation you should show, that you treat this seriously.

Finally, and this again is similar to security, we will have **periodic or one-time reviews**, and they will deliver some **findings**. Corresponding indicators are:

- Number of compliance findings in audits or reviews during the last reporting period,
- Number of compliance findings, which have not yet been resolved.

Finally it could be interesting to ask for the **costs of IT compliance**. Like quality the compliance costs of an IT organization would be 100% of the IT costs, because all activities of an IT organization contribute to good or better compliance. So we should ask for **compliance management costs** and define the indicator:

- Portion of costs for IT compliance management with respect to total IT costs.

However, we have to differentiate between security, risk, or compliance management costs, which the NIST proposal obviously does not precisely.

7.5 BUSINESS CONTINUITY

Finally let us consider business continuity management respectively IT continuity management. Here we define:

Business continuity management is the process of creating and running IT systems of prevention and recovery to deal with potential threats to an IT organization.

The situation to measure the level of being ready for business continuity is similar to IT security. We can measure the **detected defects**, nothing else. Here it makes sense to define the subsequently given indicators for an ongoing measurement:

- Number of business continuity improvements, which have been defined, but have not yet been implemented,
- Level of fulfilment of defined test and training activities in the actual reporting period,
- Number of findings, e.g. application not well prepared for business continuity, in business continuity reviews, which have not yet been resolved.

Though we have only defined a few metrics, and these metrics are very simple mathematically, we are convinced that a **day-to-day application** of those metrics will effectively measure, whether the level of readiness for IT continuity has been improved or not.

Now, at the end, let us consider, who is responsible. Of course, IT management is responsible for continuity readiness of IT systems and IT infrastructure. But business continuity is not only a task for IT. IT can just be a part of it. Hence the **responsibility for business continuity** is primarily assigned to business management, and only the IT part of it is delegated or transferred to IT management.

7.6 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.7!

7.6.1 QUESTIONS FOR YOUR SELF-STUDY

Q7.1: Specify specific security objectives and compare it to the very generic security objectives, which have been presented in this chapter of the book.

Q7.2: Give examples of threats or risks, which are related to IT or information security, and give also examples, which are not related to IT or information security.

Q7.3: Give examples of documents to which an IT organization has to be compliant.

Q7.4: Find additional metrics for IT security, IT risk, or IT compliance management.

Q7.5: Risk is usually defined as the product of the probability of occurrence and the value of the expected damage, which is caused by the risk. Discuss this approach critically. How can you provide the necessary information to run the necessary calculation?

7.6.2 PREPARATION FOR FINAL EXAMINATION

T7.1: How are IT and information security defined? What are the differences?

T7.2: Give the list of the basic security objectives.

T7.3: What are the four risk management strategies?

T7.4: How is the term “compliance” defined?

T7.5: What is the responsibility of business continuity management?

7.6.3 HOMEWORK

H7.1: Which indicators do you find in the internet to measure security, risk, or compliance?

H7.2: Give a recommendation to your organization, how they should prepare themselves to IT continuity.

H7.3: How can you identify costs for security, risk, or compliance in your organization? Could it be easier or more successful to ask for the costs of security management, risk management, or compliance management?

8 PROCESS MANAGEMENT AND ORGANIZATIONAL PERFORMANCE

Learning objectives

In this chapter you will learn,

- what a process is, and what are the roles of a process owner and a process manager,
- what are the targets of a process and how we can measure the process performance,
- how we can measure the performance of a group of processes or even a total organizational unit,
- how process management is related to cost management,
- how process frameworks can help to identify and define processes in the own IT organization.

Recommended pre-reading

- Damij 2014

8.1 FOUNDATIONS OF PROCESS MANAGEMENT

Processes play an important role in IT management. Well-known frameworks like COBIT or ITIL are more or less driven by thinking IT as a set of processes. Hence, also IT performance management has to consider processes and has to find ways to measure processes and their performance.

But what is a process? What, more specific, is a business process? Finally, what are workflows and how can they be differentiated from processes? And last but not least, what is a process manager, and what are his / her duties?

We start with a definition of **business** as it is used by Damij 2014, p7: "Business covers all activities within an organization, such as planning, organizing, and producing marketable products or services at a required level of quality, and selling at competitive prices."

Based on this definition the term "**business process**" is defined as follows (see Damij 2014, pp. 15-17): "By a business process we mean the different processes conducted within various types of organizations whose purpose is creating outputs that are produced to serve customer´s needs." Usually the total set of business processes is sub-divided into **core processes**, which concentrate on satisfying external customers, and **support processes**, which concentrate on satisfying internal customers (see Damij 2018, p18).

IT performance management, of course, considers IT processes, which are those processes running to provide IT services to another organization. The core processes of an IT organization are those processes, which are needed to deliver IT services to the IT customers, and the support processes are those IT processes, which deliver some output to other parts of the IT organization. **Business processes** are those processes, which run under the responsibility of the IT customers, and the delivered IT services represent some of the inputs, needed by those business processes.

Against that background we can give the basic definition of a process, which is a very general definition, but, of course, holds for IT processes, too (see Damij 2014, p17, and figure 29):

A process is a set of related and interconnected activities, that takes input from one or more suppliers, transforms this input into output for one or more customers.

We should note that this definition is consistent to the IPO model, which we have already used for several times in this book. However, we should ask, whether this definition is sufficient to fully understand processes. We have to describe, what is really going on within and between the activities of a process. Thus we need **process rules**, which describe, what is going on within the activities, and we need **decision rules**, which describe the sequences and the relations between the different process activities. Finally we need some **responsible persons**, and here we should differentiate between the **process owner** and the **process manager**.

The **process owner** is a person, who is responsible for a process in general. He or she is looking onto the process from an outside perspective. Due to our understanding in this book he/she is the manager, and the process is the management object.

However, in an IT context we also use the term "**process manager**". This is a person working in the process, and he or she is responsible for a **specific execution** of the process. There may be an identity between the process owner and the process manager. But there may also be a one-to-many relationship between process owner and process manager, e.g., if we consider an IT service desk, where processes are executed in 7×24 mode. Here the shift manager has the role of process manager in this specific meaning.

The different roles of process ownership and process (execution) management point to a phenomenon of processes. We have to differentiate between the process as a structure and a set of rules (or: the process model) and the many process executions.

And this consideration leads to the missing element of our process definition. We need a signal or **trigger**, by which a process execution is started, and another signal or trigger, by which a process execution is finished. Now we can completely describe a process (see figure 31).

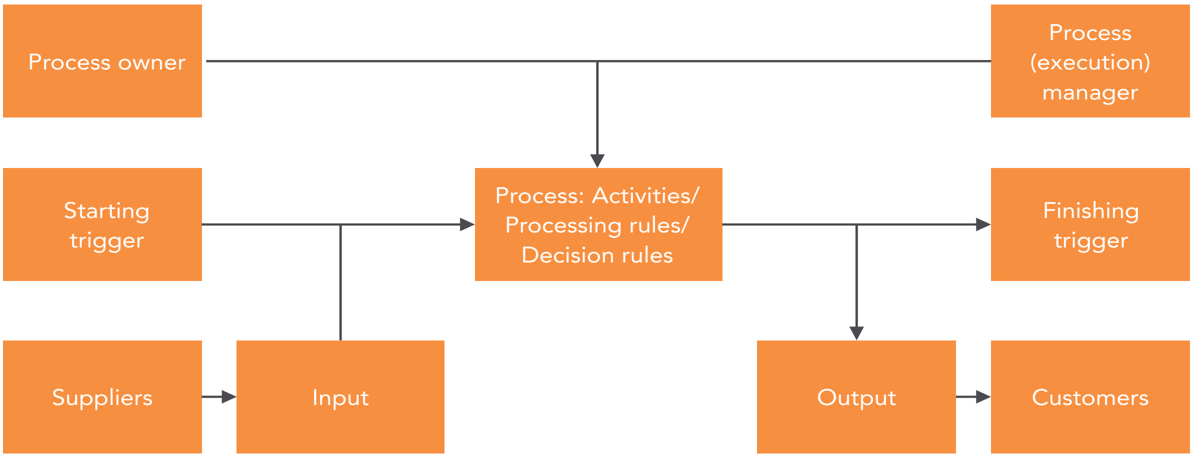


Fig. 31: Process model

Strongly related with the starting and finishing trigger of a process is the **flow unit**. This is an item, which is inserted into the process with the starting trigger, flows through the process, and is finally deleted by the finishing trigger. At any time during its journey through the process it can be found either in an activity or on the way between two subsequent process activities. A customer request initiating a process execution or some ticket, which is generated in the initiation step of a process execution can be the carrier of the flow unit.

The preceding considerations are based on the assumption, that there is a flow unit for the process. If no flow unit is in operation, then the process is waiting. If there is at least one flow unit in the process, then the process is up and running.

But what about processes, where we do not have a flow unit or ticket? Consider e.g. configuration management. If a CI (configuration item) is added, deleted or changed, then we have a ticket / flow unit, of course. But which situation do we have, if no such activities are running? Configuration management will consume resources, even if there is no activity on CI level. What is the output of that hidden process? Is it the CMDB (configuration management database) and its availability? Then it is the job to keep it **available for one time unit**. Obviously the corresponding process has a somehow specific or even strange nature.

How can we integrate this in our flow unit model? There is exactly one flow unit allowed to be in that process at one point of time. And when the flow unit leaves the process at the end of a time interval, then it enters the process again immediately, seamless with no interruption.

The same holds for the **provision of resources**. Also a resource pool must be provided and kept available.

Finally we have **management processes**. What is the output of a management process? In the sense of ticketing it could be decisions. But what is going on, if no decision has to be made? Also management is some kind of a resource, which must be available.

Here we can come to the point to introduce the term "workflow". Often **workflows** are defined like highly automatable processes. This definition shows, that a workflow is a process, but not every process will be a workflow. Here the flow unit can help to make a clear definition:

A workflow is a specific category of processes, which are controlled by flow units.

In a workflow the flow unit takes over the control of the process execution It is the driver of the process, and with a workflow management system (WfMS) the process execution can be automated completely such that the workflow management system resp. the flow unit, which has been assigned to a process execution, signals when it needs some human interaction. The human resource is just a resource, nothing else.

Finally let us compare the two process categories (see figure 32).

Workflow processes (WF processes)	Availability and management processes (nWF processes)
• separate executions, possibly running in parallel • clearly defined start and end of process execution • can be automated to a high degree (and controlled by a WfMS) • clear structure with activities and decision rules (for selection of different process branches)	• exactly one execution at a specific point of time • continuously running • needs human interaction • has to deal with unexpected situations • often clear internal structure missing • output definition harder than for workflows

Fig. 32: Process categories

8.2 MEASUREMENT OF WF PROCESSES

Now we can consider the first and one of the most important indicators to measure process performance, the cycle time (alternatively: flow time). To define the **cycle time** we need the term "flow unit". Such a **flow unit** is an item that flows through the process. At any time during its journey through a process it can be found either under a certain work performed in an activity or waiting in a buffer. Depending on the specific nature of the process it may be a unit of input such as a customer order (see Damij 2014, p62).

The cycle time of a process is the total time spent by a flow unit within the process boundaries.

We can this cycle time differentiate:

- processing time (related to value-adding activities),
- inspection time,
- transporting time,
- storage time,
- waiting time (planned and unplanned delay time).

If we consider the **theoretical cycle time** as the minimum amount of time required for processing a typical flow unit without any waiting, then we have the equations:

- cycle time = theoretical cycle time + waiting time,
- cycle time efficiency = theoretical cycle time / average cycle time.

Another important metric for process management is **flow rate**.

The flow rate of a process is the number of flow units, that flow through a specific point in the process per unit of time.

There may be different in-flows and out-flows in a process, being related to different flow rates. The result of that metric depends on the point of measurement, which we have chosen. Related to the term "flow rate" is the term "process capacity":

The process capacity is the maximum sustainable flow rate of a process.

In practice we will find another four metrics in use:

- the inventory, which is the number of flow units, that may be found within the framework of the process at any point of time,
- the average flow rate, which is the average number of flow units that flow through the process per unit of time,
- the average cycle time, which is the average across all flow units, that exit the process during a specific period of time,
- the average inventory, which is the average number of flow units within the process boundaries at any point of time.

A process is **stable**, if the average in-flow rate is the same as the average out-flow rate. In a stable process there is **Little´s law** valid. It says, that in a stable process we have the equation $I = R \times T$, which means, that the average inventory I is the product of the average flow rate R and the average cycle time T.

However, there is another challenge for performance management of (IT) processes. The cycle time is extremely varying. For example consider an incident ticket. It may be resolved within a few minutes, but it may also take days or weeks. Does the average cycle time make sense as a metric for the cycle time?

Another challenge is, that for a real system we can talk about the theoretical capacity. But do we have a real chance to find it out for very complex systems, which cannot be tested due to economic reasons? The answer obviously is No.

To measure the flow time of a ticket process we should not take the average value but use quantiles. What is the flow time value, such that 80% of the process executions have been finished with that time span and 20% of the process executions exceeded that time-span.

8.2.1 MONITORING CHALLENGES

Now let us come back to daily operations. What can be measured resp. must be measured, if we want to manage the performance of processes? We see the following approaches:

- the number of jobs, which have been processed / finished in a specific period of time,
  - This metric can be used in case of processes with flow unit / tickets, the so-called ticket-based processes (TBP).
  - What about jobs, which failed, that means, that those jobs had been started, but not finished successfully?

- the number of requests for jobs, which were given to the process within a specific time span (e.g. reporting period),

- the number of output units, which have been produced at the requested quality level by jobs, which have been finished in a specific period of time (e.g. reporting period),
  - If a process generates different outputs, then the different output units should be transformed into a normalized output unit (by using weights or equivalence numbers).

- the number of input units, which have been consumed by the considered process within a specific time span (e.g. reporting period),
  - If different resources are used, then we must build a weighted sum of those input units – we usually will talk about costs, then.
  - If a resource pool is provided exclusively for that process, then it might make sense, to differentiate between used capacity units and unused capacity units.
  - If we have a shared pool of resources, which is used by different processes, then we have the problem, that the pool capacity is dynamically assigned to or taken by different processes.
  - The proceeding and issues of input measurement can be considered for every type of processes, also for providing capacities and running the necessary infrastructure.

- the flow time of a process,
  - It is different for each job / process execution.
  - If processes have different branches, then the target flow time can differ significantly.
  - Here the following indicator would make sense: portion of jobs, which were done within the given amount of time (This time could be different for different branches of the process. In an extreme situation the target flow time could be individual for each single job).
  - Measurement of flow time does not make sense for processes, which are not workflows.

8.3 MEASUREMENT OF NWF PROCESSES

How can we measure availability? See chapter 2 and 5 on service availability.

Outputs of nWF processes can be:

- time units of availability,
- number of assets under administration (e.g. licenses, configuration items, knowledge items, etc.),
- capacities (Is the number of CIs a specific category of capacity? Is the output capacity-hours?).

Performance management for processes (in an IT organization) should concentrate on capacity utilization.

Finally we have to consider bundles of processes. The strategy of measurement will be similar to the measurement strategy for single processes. We will measure:

- resource consumption (see resource pools, which have to be shared with other process bundles),
- output quantities (here consider the mixture of WF and nWF processes),
- flow time (only for the included WF processes),
- process quality (which for a WF process is the quality of the output units).

For nWF processes quality can be the quality of the administered asset or capacities. Does it make sense to speak about quality of availability? Is it reliability here?

Within a process we can measure processing time and waiting time. However, we have to define measurement points.

8.4 MEASUREMENT OF RESOURCE POOLS

Let us now consider resource pools, because the availability of resources will be an essential factor that impacts the capacity of a process. Related to the flow rate we have the theoretical capacity of a resource unit, which is the maximum sustainable flow rate, if it is fully utilized (Damij 2014, p79). Thus the theoretical process capacity is determined by the smallest theoretical resource capacity. And this leads to another metric. This is capacity utilization, which is the ratio of the actual flow rate and the theoretical process capacity.

Our problem in IT performance management is, that our resource capacities are used / consumed by different processes. Hence bottlenecks will change due to the actual total capacity utilization by different processes. Today processes may have another theoretical capacity than yesterday.

8.5 MEASUREMENT OF PROCESS BUNDLES

Can we transfer the concept of idle time to nWF processes? Can we refer to service management here? Does it make sense to speak about idle times in case of services? Yes, if there is no service request and all service jobs have been finished, then the service is idle. This is similar for configuration management. If nobody is using the CMDB, then it is idle. Thus idle time is a specific case of capacity utilization. During idle times capacity utilization is 0%.

Considering now a **complete organization** with a process oriented view means, that we consequently look for **output and customers**, **input and suppliers**. If we are not able to denominate the output of a process, then it could make sense to eliminate it. If we cannot identify the customer of a process, then we should also eliminate it, because a process without any customer does not make any (economic) sense. The customer is seen here in a very general meaning as a person or organization taking a specific output. The customer could also be a state authority, which we have to deliver something due to legal requirements.

On the other side our processes need **resources and capacities**. This must be considered with respect to the delivered output. Due to the **economic principle** the resource and capacity provision must be strictly limited to those quantities and capacities, which are necessary to deliver the requested output.

Process thinking makes us thinking our organization in beginning with the output side and restrict our activities strongly to those activities, which are necessary to produce and deliver that output.

How can we measure the **size** and **complexity** of processes? Let us start with a WF process. Here the number of included activities could be a metric for the size of the process (in a static sense), assuming, that all activities have the same size – across all considered processes. If the activities differ in size, then we have to define weights for the activities according to their local size. The size of the process then is just the number of the weights.

Considering complexity we can refer to complexity metrics of software, because each software can be considered as a specific WF process.

But what is the size of a nWF process? This is an open question. Or not? If it is a process, then we should be able to give a list of activities of the considered nWF process. To measure complexity will be a problem, because we do not have a network structure in this case.

The next challenge is to measure **productivity** or **economic efficiency** of a process. Productivity is – please remember – the ratio of an output quantity and an input quantity. This is easy to calculate, if the consumed input can be assigned to a specific output. If not, then we have the situation of indirect costs, and that makes it difficult. However, also for processes unit costs are the traditional measure of economic efficiency. This can be done as long as the output unit is unchanged. Otherwise we have to compare the new output unit and the old output unit via an equivalence number.

Do we need a variant of standard costing, if we discuss fixed costs issues?

8.6 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.8!

8.6.1 QUESTIONS FOR YOUR SELF-STUDY

Q8.1: Please list core processes and support processes of an IT organization.

Q8.2: What are the tasks and responsibilities or a process manager and a process owner?

Q8.3: How can we apply Little´s law to processes in an IT organization?

Q8.4: Make a recommendation how to measure the performance of a complete IT organization.

Q8.5: Make a recommendation to determine unit costs of processes.

8.6.2 PREPARATION FOR FINAL EXAMINATION

T8.1: Describe the main process categories and their main differences.

T8.2: Explain the terms "cycle time" and "flow rate".

T8.3: What are the elements of the process model, which we have presented in this book?

T8.4: List at least three monitoring challenges in the area of process performance management.

T8.5: What do we understand by the term "process bundle"?

8.6.3 HOMEWORK

H8.1: Find process models for IT organizations. Could we apply process models from other areas to an IT organization?

H8.2: Conduct a research, which organization could offer consulting to your organization to improve your IT processes.

H8.3: Conduct a research on performance management for business processes. What do you find? Can you apply this to IT processes?

9 PERFORMANCE MANAGEMENT TOOLBOX

Learning objectives

In this chapter you will learn,

- what are the most important tools for every performance management expert,
- how those tools can be used effectively in your daily operation,
- that each tools has its specific advantages as well as its disadvantages and limitations.

Recommended pre-reading

- Oh 2016

9.1 SCORING

Scoring (also: value benefit analysis) is a very useful tool for preparing decisions. Its actual advantage is that non-quantitative, so-called "soft", criterions can be taken into account. That allows us to use the scoring approach in a vast set of practical applications. Scoring is a multidimensional method, and thus integrates different perspectives or characteristics of the objects to be valued. Thus the target system is very flexible.

9.1.1 MATHEMATICAL ASPECTS

From a mathematical point of view scoring is nothing but building a **weighted sum**:

$$V = \sum_{i=1}^{n} \alpha_i \times c_i$$

Here we have chosen n criterions c_i, $1 \leq i \leq n$, $c_i \geq 0$, which are weighted by $\alpha_i \geq 0$. Usually we restrict the criterions to $0 \leq c_i \leq 10$ or $0 \leq c_i \leq 5$. And often the weights are standardized to $\sum_{i=1}^{n} \alpha_i = 1$. This has the advantage, that then $0 \leq V \leq 10$ resp. $0 \leq V \leq 5$. We also should note, that the value V increases or decreases by α_i, if we increase or decrease c_i by 1, and this does not depend on the original value of c_i. We also see, that V is unchanged, if we increase c_i by 1 and decrease c_j by $\frac{\alpha_i}{\alpha_j}$. In other words: V is unchanged, if all values of c_i, $1 \leq i \leq n$, lie on a hyper-plane in the n-dimensional space.

Decision-making means, that we have different objects and we have either to find out the best object, e.g. in case of procurement decisions the best product or best supplier, applicants for a job, etc., or to have a group of management objects and want to establish an order or ranking among those objects.

Typical applications in the IT area are:

- customer / user satisfaction index,
- selection of software products or suppliers or consultants,
- evaluation of technologies,
- scoring of projects in a project portfolio,
- selection of new employees.

Scoring is conducted in several stages as it is shown in figure 33.

Fig. 33: Stages of a scoring process

9.1.2 ENUMERATION OF ALL RELEVANT ALTERNATIVES

We have to find all solutions of our problem, which we want to include in our decision making.

9.1.3 DETERMINING CRITERIONS FOR THE COMPARISON

We have to develop the criterions that we want to use for the assessment. The value of Scoring will become more distinctive by using many criterions but on the other hand the difficulties to assess each criterion in comparison with the others will heavily increase. Hence, all persons to be included in the decision-making should agree on the selected criterions.

9.1.4 ESTABLISH WEIGHTS FOR THE CRITERIONS

We have to evaluate the relation between the criterions. A very good way to do that is to arrange a matrix in which the rows and the columns represent the criterions. The main diagonal remains empty. Now we take each criterion and asses it in pairs with all other criterions weather it is more important than the other one. If we say that it is not more important we assign a zero to the according matrix field. If we say that it is equal to the compared one we assign a 1 to the according matrix field. If we say that it is more important than the other one we assign a 2. We do this assessment pairwise with each criterion. After that we sum up all the numbers we assigned for each criterion and divide this sum by the total sum over all single sums in the matrix lines. This decimal number represents our weight we assign to the criterions. One can say that these weights represent the portion that this criterion contributes to the final decision.

9.1.5 ESTABLISH FACTORS TO RATE THE DIFFERENT CRITERIONS

In this step we have to establish a rating scale in order to evaluate the fulfillment of the criterions for each alternative. An example for a rating scale could be:

- 0 - 2: insufficient fulfillment
- 3 - 5: medium fulfillment
- 6 - 8: good fulfillment
- 9 - 10: excellent fulfillment

Such a rating scale allows us to rate the alternatives regarding each criterion differentiated.

9.1.6 DETERMINE THE VALUE OF BENEFIT

Now we can combine the rating scale and the weights by judging how good an alternative accomplishes the criterions. This means how many rating points do they obtain? After we assign the rating points to the alternatives we have to calculate the value of benefit. Therefore we multiply the rating points with the corresponding weights and sum these products up for each alternative. That alternative with the highest sum (value of benefit) should be selected.

Example: A new enterprise resource planning (ERP) software system shall be purchased and installed in your enterprise. Since there are several providers for such systems the providers have to be assessed in order to make the correct choice. The assessment method that is used is the value of benefit analysis. We have to follow the steps as shown above:

9.1.7 DEVELOPING CRITERIONS FOR THE COMPARISON

We have to develop criterions and have to label them according the importance. We will use the criterions given in figure 34.

Criterion	Importance
1. Over-all impression if the enterprise	*Most important*
2. Experience and expertise of the consultants	*Most important*
3. Estimated duration of the whole project	*Important*
4. Costs per day and per consultant	*Most important*
5. Similar reference projects have been successfully completed	*Less important*

Fig. 34: Qualitative weighting of criterions

9.1.8 ESTABLISHING WEIGHTS FOR THE CRITERIONS

Criterion	1	2	3	4	5	Weight	Factor
1	-	1	0	1	1	3	0,136
2	1	-	1	2	1	5	0,227
3	0	1	-	1	2	4	0,182
4	1	2	1	-	1	5	0,227
5	1	1	2	1	-	5	0,227
					Sum	**22**	**1,000**

Fig. 35: Determination of weights

If one criterion is more important than another we give a "2". If it is equivalent we give a "1" and if it is less important we give a "0" (see figure 35).

9.1.9 ESTABLISH FACTORS TO RATE THE DIFFERENT CRITERIONS

We will use the rating given in figure 35. That leads us to the matrix given in figure 36.

		Provider A		Provider B		Provider C	
Criterion	Factor	Rating A	Value of benefit	Rating B	Value of benefit	Rating C	Value of benefit
1	0,136	4	0,544	6	0,816	8	1,088
2	0,227	7	1,589	6	1,362	6	1,362
3	0,182	6	1,092	8	1,456	7	1,274
4	0,227	5	1,135	7	1,589	2	0,454
5	0,227	3	0,681	4	0,908	5	1,135
	Sum:		**5,041**		**6,131**		**5,313**

Fig. 36: Execution of a scoring

Since provider B offers the highest value of benefit this provider has to be chosen to purchasing the ERP system.

9.1.10 MULTI-LEVEL SCORING

The scoring can be generalized to a multi-level scoring. This means, that we have groups of characteristics, which are valued group-wise, and then those partial scores are combined to the total score by building a weighted sum of those partial scores. We show the formula for a 2-level scoring:

$$V = \sum_{i=1}^{n} \alpha_i \times v_i = \sum_{i=1}^{n} \alpha_i \times \sum_{j=1}^{m} \beta_j \times c_j$$

9.2 ABC ANALYSIS

The **ABC analysis** is a categorization method. We consider a set of n homogeneous objects, and we assign a quantitative value to each of those objects. Then we sort those objects according to the considered (numeric) characteristic in decreasing order. Now we consider the two-dimensional vectors $(1, v_1), (2, v_2), (3, v_3), \ldots, (n, v_n)$ with $v_1 \geq v_2 \geq v_3 \geq \ldots \geq v_n$. Now we build

$w_1 = v_1$, $w_2 = v_1 + v_2$, $w_3 = v_1 + v_2 + v_3$, …, $w_n = \sum_{i=1}^{n} v_i$ with $w_1 \leq w_2 \leq w_3 \leq \ldots \leq w_n$. Now we put the points $(1, w_1)$, $(2, w_2)$, $(3, w_3)$, …, (n, w_n) into a cartesian system of coordinates and link subsequent points by straight lines. Then we get a typical curve as it is shown in figure 37.

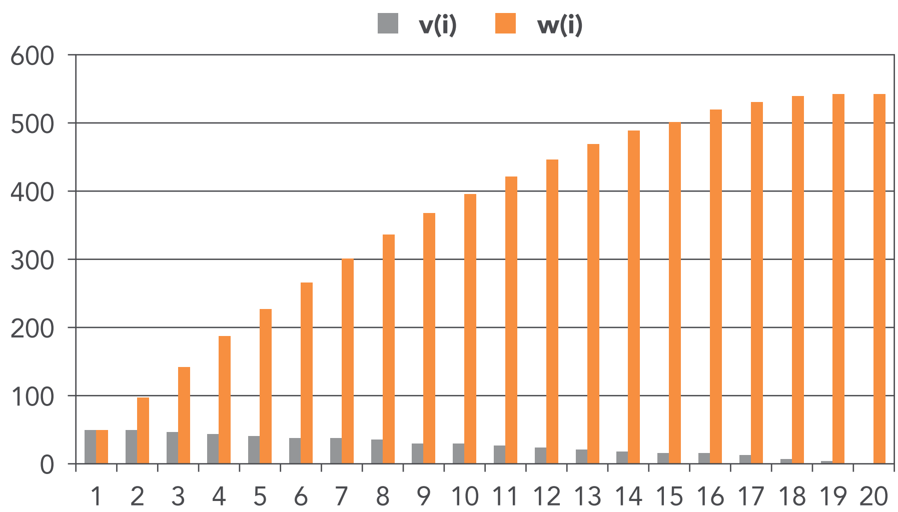


Fig. 37: Graphic representation of an ABC analysis

If we organize a set of objects in this way, then we can ask for the number j, such that

$\sum_{i=j}^{n} v_i \leq \gamma \times w_n$ (C-elements) or $\sum_{i=1}^{j} v_i \leq \alpha \times w_n$ (**A-elements**). The denomination of the low-value end with C and the high value end with A has generated the denomination of this method as ABC analysis. The origin of this method lies in the procurement area. The suppliers with a big procurement volume are the A-suppliers, and the get a lot of attention. Procurement management does not love suppliers with a low procurement volume are… Accordingly, if we set $\alpha = \gamma = 20\%$, then the value of j will be quite low because only few elements will concentrate 20% of the total sum of all assigned values, and the value of $n - j$ will be a quite high value, because it needs a lot of low-value elements to have 20% of the total sum of all assigned values. In figure 35 there will only be two A-elements but nine. The reader will find that many sources say that the ABC concept is based on Pareto's law.

Applications of ABC analysis in the IT environment could be:

- distribution of service volume to customers,
- distribution of costs to cost types or cost centres,
- distribution of projects according to man-day volume.

9.3 SWOT ANALYSIS

SWOT analysis (or **SWOT matrix**) is often called a strategic planning technique used to help a person or organization identify the strengths, weaknesses, opportunities, and threats related to business competition or project planning. From a more general point of view and making it an interesting tool for IT performance management, it is a method to help rendering judgments about management objects. Terms and phrases are collected and each of them is assigned to one of the four mentioned categories, and this leads to a structured presentation of all collected arguments.

Strengths and weakness are frequently internally-related, while opportunities and threats commonly focus on environmental placement. The first two categories are considering the actual situation, the second two categories look into the future.

- *Strengths*: characteristics of the management object that give it an advantage over others,
- *Weaknesses*: characteristics of the management object that place it at a disadvantage relative to others,
- *Opportunities*: elements in the environment that the management object could exploit to its advantage,
- *Threats*: elements in the environment that could cause trouble for the management object.

9.3.1 WHEN TO USE SWOT ANALYSIS

The uses of a SWOT analysis by a community or organization are as follows: to organize information, provide insight into barriers, that may be present while engaging in changes, and identify strengths available that can be activated to counteract these barriers.

In an IT environment a SWOT analysis can be used to (see figure 38):

- evaluate new technologies or methodologies,
- render judgments on suppliers or products,
- discuss organizational issues, e.g. centralized vs. decentralized IT organization,
- discuss strategic directions, e.g. individual vs. standard software,
- find out the best balance between in-house activities and outsourcing,
- identify best candidates for expert or management positions.

	Actual situation	Future situation
Positive aspects	Strengths:	Opportunities:

Negative aspects	Weaknesses:	Threats:

Fig. 38: SWOT analysis

9.3.1 BENEFITS AND ADVANTAGES

The SWOT analysis is beneficial because it helps organizations decide whether or not an objective is obtainable and therefore enables organizations to set achievable goals, objectives, and steps to further change or development effort. It enables organizers to take visions and produce practical and efficient outcomes that effect long-lasting change, and it helps organizations gather meaningful information to maximize their potential. Completing a SWOT analysis is a useful process regarding the consideration of key organizational priorities. Last, but not least, the SWOT matrix is a helpful tool to discuss issues with higher management levels.

9.4 SENSITIVITY ANALYSIS

For functions of more than one independent variable, $y = f(x_1,\ldots,x_n)$ the partial differential of y with respect to any one of the variables x_i is the principal part of the change in y resulting from a change dx_i in that one variable. The partial differential is therefore

$$\frac{\partial y}{\partial x_i} dx_i$$

involving the partial derivative of y with respect to x_i. The sum of the partial differentials with respect to all of the independent variables is the total differential

$$dy = \frac{\partial y}{\partial x_1} dx_1 + \frac{\partial y}{\partial x_2} dx_2 + \cdots + \frac{\partial y}{\partial x_n} dx_n$$

which is the principal part of the change in y resulting from changes in the independent variables x_i.

How can we apply the total differential to performance management? Due to its mathematical nature we must have a function with several independent variables. If we consider a scoring function $v = \sum_{i=1}^{n} \alpha_i \times c_i$, then we have $dv = \alpha_1 \times dc_1 + \alpha_2 \times dc_2 + \cdots + \alpha_n \times dc_n$ with $\frac{\partial V}{\partial c_i} = \alpha_i$. So we can calculate, how much v will change approximately, if c_i increases by *0,1* and c_j decreases by *0,2*. The result will be $dv = \frac{1}{10} \times \alpha_i - \frac{2}{10} \times \alpha_j$.

Another interesting application is the ROI formula, which we introduced in chapter 6.6. It is $ROI = \frac{B-X}{S \times T}$. This leads to

$$dROI = \frac{\partial ROI}{\partial B} \times dB + \frac{\partial ROI}{\partial X} \times dX + \frac{\partial ROI}{\partial S} \times dS + \frac{\partial ROI}{\partial T} \times dT$$

$$dROI = \frac{1}{S \times T} \times dB - \frac{1}{S \times T} \times dX - \frac{B-X}{T \times S^2} \times dS - \frac{B-X}{S \times T^2} \times dT$$

Let us now consider a project with *B = 170.000*, *X = 20.000*, *S = 100.000* and *T = 4*. What happens, if the expected value of benefits has to be reduced by 10.000 (*dB = -10.000*) and the project volume increases by 10.000 (*dS = 10.000*)? The original ROI was 0,375, and now it will decrease by 0,0625.

Usually sensitivity analysis is done in a different way. The performance manager considers a function $y = f(x_1,\ldots,x_n)$, selects one of the variables x_i, fixes all other variables to the actual values, and then looks for the maximum or minimum value of x_i, such that y does not exceed or does not fall below a given value y_0. Let us again show an example with the ROI formula. We start with the value used above leading to *ROI = 0,375*. Now we can fix *B*, *X*, and *T*, and ask, how much the project budget is allowed to increase such that ROI does not fall below 0,3. We have $ROI = \frac{170000-20000}{4 \times S} = \frac{37500}{S} \geq 0{,}3$. This leads to *S ≤ 125.000*, if all over variables do not change. Thus the total differential has the advantage, that we can investigate the simultaneous changes of different variables.

9.5 PORTFOLIO ANALYSIS

The portfolio analysis as we define it here is a tool to present and communicate structures and the noteworthiness of a set of management objects, which are represented by a two-dimensional vector of (numerical and non-negative) values. We consider those objects as points in a Cartesian coordinate system. The distribution of those points in the plane can show specific issues or problems (see figure 39).

The portfolio analysis can be applied to many management issues in the IT environment, e.g.:

- economic efficiency vs. strategic value in a project portfolio,
- benefits vs. risks in a project portfolio,
- payback period vs. benefit volume in a project portfolio,
- age of system vs. its operating costs,
- age of employee vs. his / her experience on the job,
- output volume vs. unit costs,
- IT costs per workplace vs. revenues per workplace.

9.5.1 ASSESSMENT OF THE METHOD

Portfolio analysis effectively supports communication with (top) management. Top management consultants use it is very often, they love it. They often present the portfolio as a 2×2 or 3×3 matrix and try to find out particularities of each cell in this matrix. And indeed we can easily see, whether the majority of data points is concentrated in specific areas of the plane or whether there are areas with no or very few data points in it. Then management can discuss such a phenomenon, find out the reasons, and finally come to a conclusion, whether this is a positive or a negative situation.

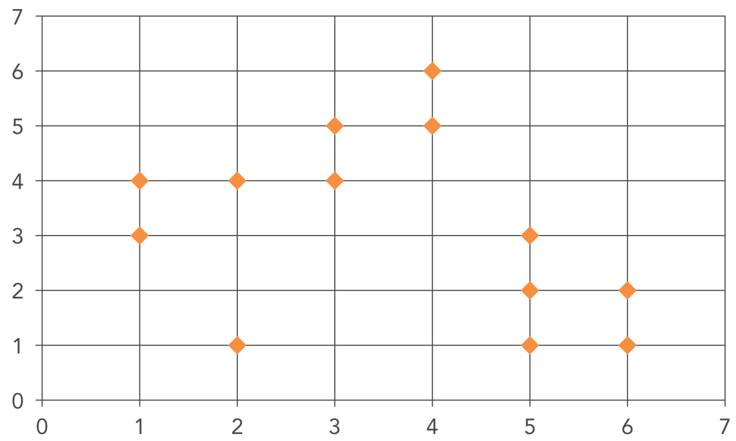


Fig. 39: Portfolio Analysis

9.6 RADAR CHART AND POLARITY CHART

The portfolio analysis can be generalized to n dimensions such that each management object under consideration is represented by a n-dimensional vector. If we want to present this on a piece of paper, then we can do this by representing each management object by a closed line with n vertices. This leads to radar charts / Kiviat charts, e.g. risk profile (see figure 40). We recommend, to use not more than 6 dimensions.

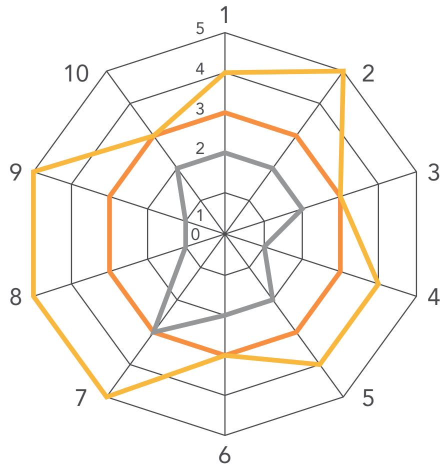


Fig. 40: Radar chart

An alternative graphical representation of n-dimensional objects is given by polarity charts. Here the zero-vector is not a point as it is for radar charts but a straight line (see figure 41).

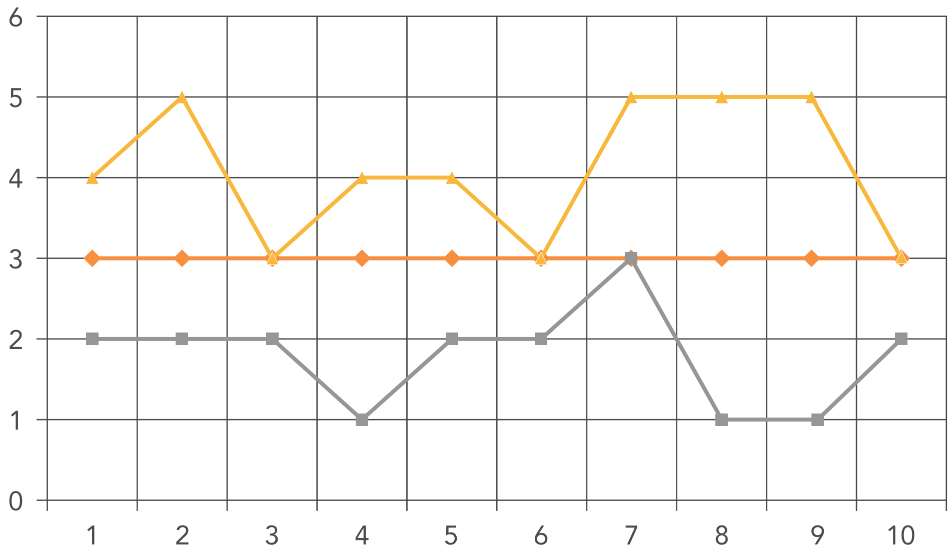


Fig. 41: Polarity chart

9.7 PAIRWISE COMPARISON

We have already discussed the usage of equivalence numbers. No we present a method to generate such numbers from pairwise comparisons systematically.

The starting point was is a set of n objects with a measurable parameter $w_i > 0$, $1 \leq i \leq n$. Now we build the matrix A with the elements $a_{ij} = \frac{a_i}{a_j}$. This matrix has some nice properties:

- The elements a_{ij} are positive.
- The matrix is denominated as a reciprocal matrix due to $a_{ji} = \frac{1}{a_{ij}}$.
- We have $a_{ik} = a_{ij} \times a_{jk}$.
- The maximal eigen-value of this matrix is n.
- The eigen-vector of this matrix, related to the eigen-value with the highest value n, is $(w_1, w_2, \ldots, w_n)$. All other eigen-values are 0.

Matrices with these properties are denominated as **consistent matrices**.

Now the question is, whether we can calculate the coordinates of the eigen-vector, if we only know the relations between the coordinates. Or mathematically spoken: Let a reciprocal matrix with positive elements be given. Can we determine the eigen-vector? If we are able to determine this eigenvector and generate the corresponding consistent matrix, how much is it different from the original reciprocal matrix?

Fortunately mathematics can help us here. The American mathematician Thomas Saaty has shown, that the eigenvector of a reciprocal matrix can be approximated in a way that the sum of the coordinates is normalized to 1. He has given the following algorithm:

- Determine the ratios of the considered parameters by the values are 1, 3, 5, 7, 9, 1/3, 1/5, 1/7, 1/9. To take those values is a convention; there is no mathematical reason behind it.
- Build the corresponding reciprocal matrix V.
- Take the elements of each column and normalize them so that its sum is equal to 1.
- Take the (normalized) elements of each row and build its sum.
- Take the elements of that new column and normalize it so that its sum is equal to 1. This resulting column is the (approximated) eigen-vector of the reciprocal matrix.
- If you want a better approximation, build V^2, set $V = V^2$, and start the normalization steps again. Otherwise you can stop the algorithm.

Figure 42 shows an example of the AHP algorithm, which has been stopped after the first iteration.

	1	2	3	4	Weights
1	1:1	3:1	1:1	5:1	**0,37**

2	1:3	1:1	7:1	9:1	**0,40**
3	1:1	1:7	1:1	3:1	**0,18**
4	1:5	1:9	1:3	1:1	**0,05**
Sum	2,53	4,25	9,33	18,00	1,00

Fig. 42: Results of an AHP process

This AHP is widely used in market research or forestry. It is not yet well known in IT, but already used e.g. in the area of requirements management.

9.8 TOPSIS

TOPSIS stands for **Technique for Order of Preference by Similarity to Ideal Solution**. It is a multi-criteria decision analysis method, which was originally developed in 1981. The method is based on the idea, that, if you start from point A, aim to reach point C, and have reached a point B, then a metric for your target achievement level is the ratio of the distance AB and the sum of the two distances AB and BC.

The idea behind TOPSIS is also similar to the idea of scoring. We have different management objects and we have either to find out the best management object or to establish a ranking for the whole set of management objects under consideration. Each management object is represented by the vector of its partial scores, and we assume, that increasing scores are equivalent to a higher value of the respective management object.

We have $m_i = (c_{1i}, c_{2i}, \ldots, c_{ni}), 1 \leq i \leq n$ and add two extreme management objects to this set, namely the best management object $\underline{m}_u = (max\ c_{1i}, max\ c_{2i}, \ldots, max\ c_{ni})$ and the worst management object $m_l = (min\ c_{1i}, min\ c_{2i}, \ldots, min\ c_{ni})$ (see figure 43). Now we select a metric d and for each management object i we define

$$e_i = \frac{d(\underline{m}_l; \underline{m}_i)}{d(\underline{m}_l; \underline{m}_i) + d(\underline{m}_i; \underline{m}_u)}$$

$$\frac{d(m_i; m_u)}{d(m_i; m_l)} = \frac{1 - e}{e}$$

For $\underline{m}_l$ we have $e_l = 0$, and for $\underline{m}_u$ we have $e_u = 1$. For all management objects we have $0 \leq e_i \leq 1$. Those management objects lead to the same value e, which fulfill the following equation

$$\frac{d(m_i; m_u)}{d(m_i; m_l)} = \frac{1-e}{e}$$

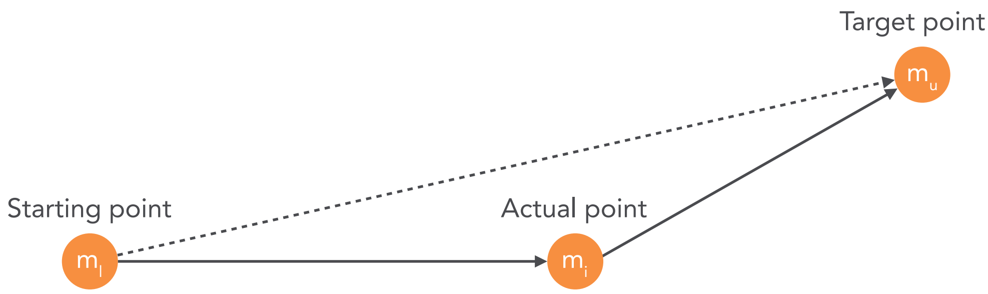


Fig. 43: The geometry behind TOPSIS

We have already used this method to measure target achievement levels (see chapter 2).

9.9 EXERCISES

For information to answer the questions and prepare for the final examination see chapter 10.9!

9.9.1 QUESTIONS FOR YOUR SELF-STUDY

Q9.1: List application areas for scoring. Is it possible to apply this method to your private life?

Q9.2: Which application potential do you see in your organization to apply a multi-level scoring. Does it make sense to work with more than two levels of group-wise aggregation?

Q9.3: If you do not want to use the total differential, how would you run sensitivity analysis in your organization?

Q9.4: Portfolio analysis, radar charts and polarity charts are vehicles for communication. List the pro´s and con´s of these three methods.

Q9.5: Find possible applications for TOPSIS in the area of IT performance management.

9.9.2 PREPARATION FOR FINAL EXAMINATION

T9.1: Describe the method of scoring. Give at least five criteria to select a software package.

T9.2: What is a SWOT analysis? Conduct a SWOT analysis for introducing a new technology.

T9.3: Top management wants to get information, which department in your organization consumes much IT. Can ABC analysis help to deliver this information? How would you proceed?

T9.4: List at least three application opportunities for the portfolio analysis.

T9.5: What do portfolio analysis and radar charts have in common. Explain both approaches and their differences.

9.9.3 HOMEWORK

H9.1: Run a research for further methods or tools, which could be helpful for IT performance management.

H9.2: Could it make sense to combine SWOT analysis and scoring? How would you proceed? Give the reasons for your specific approach.

H9.3: Find out, which tools or methods can be taken to support sensitivity analysis.

10 ADVICES FOR EXERCISES

In this chapter you will find some information, which can help you to answer the questions for your self-study and to prepare for the final examination. No information is given with respect to the home-work because it would need so much additional pages that it would have exceeded the allowed volume of this book.

10.1 MANAGEMENT, PERFORMANCE, AND GOVERNANCE

10.1.1 QUESTIONS FOR YOUR SELF-STUDY

Q1.1: See chapter 1.2 for the management control cycle. For information related to PDCA take any book about quality management.

Q1.2: See chapter 1.3 and consider any firm, which is using information technology to support business processes.

Q1.3: See chapter 1.4.

Q1.4: See chapter 1.7. Examples: IT department has to run the corporate SAP system. Only IT department is allowed to make contracts with external service providers, hardware and software suppliers.

Q1.5: See chapter 1.7. IT services are output of the IT supply organization, and they are input for the IT demand organizations.

10.1.2 PREPARATION FOR FINAL EXAMINATION

T1.1: See figure 1 in chapter 1.1 and figure 4 in chapter 1.6.

T1.2: See figure 2 in chapter 1.2.

T1.3: See figure 3 in chapter 1.3. Service examples: workplace service, consulting. Process example: incident management, capacity management. System examples: storage system, application software. Project examples: development of new online shop, implementation of new SAP release.

T1.4: See figure 5 in chapter 1.6.

T1.5: See figure 6 in chapter 1.7.

10.2 MEASUREMENTS, METRICS, AND MONITORING

10.2.1 QUESTIONS FOR YOUR SELF-STUDY

Q2.1: See chapter 2.3.

Q2.2: See chapters 2.2 and 2.4.

Q2.3: See chapter 2.5.

Q2.4: See chapter 2.7.

Q2.5: Examples: hit a given point precisely, exceed a given boundary, go below a given value, do not hit a given interval, realize x% more/less than the year before.

10.2.2 PREPARATION FOR FINAL EXAMINATION

T2.1: See figure 7 in chapter 2.1.

T2.2: See chapter 2.4.

T2.3: To meet target values of performance indicators is in the responsibility of the manager. Environmental indicators are driven by external forces and influence management behavior.

T2.4: See definitions in chapter 2.6.

T2.5: See chapter 2.8 and have in mind, that indicators have to be fed by measurement data.

10.3 COST MANAGEMENT

10.3.1 QUESTIONS FOR YOUR SELF-STUDY

Q3.1: See chapter 3.3.

Q3.2: See chapter 3.4 and external sources lie books or internet.

Q3.3: See chapter 3.4. Mainly well defined workflows with a high number of executions.

Q3.4: See chapter 3.2 and organization charts, which you find in your organization, in books or in the internet.

Q3.5: See chapter 3.2. Examples: personnel costs, costs for external services

10.3.2 PREPARATION FOR FINAL EXAMINATION

T3.1: Cost type / cost type / cost center / cost unit / cost unit / cost center.

T3.2: Standard costing. Planned costs: 800.000 EUR. Actual costs: 1.200.000 EUR. Target cost: 1.200 × 800 EUR = 960.000 EUR.

T3.3: You need two storage units with 360.000 EUR per year. This means 10 EUR per GB and month.

T3.4: View: cost unit. See chapter 3.4, section "Activity-based costing".

T3.5: See chapter 3.3. A primary-cost center delivers its services to external customers. A secondary-cost center delivers its output only to other cost centers in the same organization / organizational unit.

10.4 RESOURCE AND CAPACITY MANAGEMENT

10.4.1 QUESTIONS FOR YOUR SELF-STUDY

Q4.1: See chapter 4.2.

Q4.2: See chapter 4.2.

Q4.3: See chapter 4.6.

Q4.4: See chapter 4.4.

Q4.5: See chapter 4.8.

10.4.2 PREPARATION FOR FINAL EXAMINATION

T4.1: See chapter 4.2.

T4.2: If we have over-capacity, capacity is idle. It is not used, but causes costs. If capacity is too small, processes, activities, or services will have to wait until capacity is freed.

T4.3: See chapter 4.6. Buy: ownership of equipment, free in usage, low unit costs in case of good capacity utilization, risk of over-capacity. Rent: flexibility, no idle capacity, dependency from external organization.

T4.4: See figure 17 in chapter 4.7.

T4.5: See chapter 4.8.

10.5 SERVICE MANAGEMENT AND CHARGING

10.5.1 QUESTIONS FOR YOUR SELF-STUDY

Q5.1: See chapter 5.2.

Q5.2: See chapter 5.1.

Q5.3: Adherence to schedules: daily production planning, all activities with just-in-time characteristic. Run-time: Response time for banking transactions, re-organization of databases.

Q5.4: See chapter 5.4 and distribution of indirect costs.

Q5.5: See chapter 5.4. It can make sense to differentiate between internal and external customers.

10.5.2 PREPARATION FOR FINAL EXAMINATION

T5.1: See chapter 5.1. Examples: workplace service, application service.

T5.2: See chapter 5.3. Examples: availability, user satisfaction.

T5.3: See chapter 5.1. Adding a price to each service makes the service catalogue a price list.

T5.4: See chapter 5.4 and figure 21.

T5.5: See chapter 5.4. Calculate unit costs and add a margin. Make a management decision, which price you want to charge for each delivered service unit.

10.6 PROJECT MANAGEMENT AND BUSINESS CASES

10.6.1 QUESTIONS FOR YOUR SELF-STUDY

Q6.1: See chapter 6.2, section "agile proceeding".

Q6.2: See chapter 6.3. Reduce quality, or reduce functionality, or deliver basic version in time and missing features later.

Q6.3: See chapter 6.3, section "contingency planning".

Q6.4: See chapters 6.3 and 6.5.

Q6.5: See chapter 6.7.

10.6.2 PREPARATION FOR FINAL EXAMINATION

T6.1: See figure 25 in chapter 6.2.

T6.2: See chapter 6.2, section "backward planning".

T6.3: See chapter 6.6.

T6.4: See chapter 6.2.

T6.5: See figure 30 in chapter 6.3.

10.7 SECURITY, RISK, AND COMPLIANCE MANAGEMENT

10.7.1 QUESTIONS FOR YOUR SELF-STUDY

Q7.1: See chapter 7.2.

Q7.2: Related: failure of hardware components, data espionage by foreign competitors. Not related: Decrease of share values of IT supplier, breakdown of markets.

Q7.3: See chapter 7.4. Examples: laws, standards and norms, internal regulations.

Q7.4: See chapters 7.2, 7.3. and 7.4.

Q7.5: To be given: probability distribution of occurrence all expected sizes of damage.

10.7.2 PREPARATION FOR FINAL EXAMINATION

T7.1: See chapter 7.1.

T7.2: See chapter 7.2.

T7.3: See chapter 7.3.

T7.4: See chapter 7.1, section "compliance".

T7.5: See chapter 7.5.

10.8 PROCESS MANAGEMENT AND ORGANIZATIONAL PERFORMANCE

10.8.1 QUESTIONS FOR YOUR SELF-STUDY

Q8.1: See ITIL or COBIT.

Q8.2: See chapter 8.1.

Q8.3: See chapter 8.2.

Q8.4: See chapter 8.5.

Q8.5: Apply ABC. See chapter 3.4, section “activity-based costing”.

10.8.2 PREPARATION FOR FINAL EXAMINATION

T8.1: See figure 32 in chapter 8.1.

T8.2: See chapter 8.2.

T8.3: See figure 31 in chapter 8.1.

T8.4: See chapter 8.2, section “monitoring challenges”.

T8.5: See chapter 8.5.

10.9 PERFORMANCE MANAGEMENT TOOLBOX

10.9.1 QUESTIONS FOR YOUR SELF-STUDY

Q9.1: See chapter 9.1.

Q9.2: See chapter 9.1.

Q9.3: Take the given formula and increase / decrease each variable until the result reaches upper or lower limits.

Q9.4: See chapters 9.5 and 9.6.

Q9.5: See chapters 9.8 and 2.4.

10.9.2 PREPARATION FOR FINAL EXAMINATION

T9.1: See chapter 9.1. Criterions: coverage of business needs, price, reputation of supplier, consistency with own architecture, interfaces.

T9.2: See chapter 9.3.

T9.3: Find a way to measure IT consumption in total. Identify IT consumption of each department. Sort due to consumption volume in decreasing order.

T9.4: See chapter 9.4.

T9.5: Portfolio analysis: two dimensions, each object represented by a point. Radar chart: more than two dimensions, each object represented by a closed polyline.

REFERENCES

Abdallah, W 2004, Critical Concerns in Transfer Pricing and Practice, Praeger Publishers, Westport

Bhimani, A & Horngren, Ch & Datar, S & Rajan, M 2015, Management and Cost Accounting, Pearson Higher Education, Harlow, 6th edn., ISBN 978-1-29206346-1

Carr, N 2003, IT doesn´t matter, Harvard Business Review, May 2003 issue

Damij, N & Damij, T 2014, Process Management, Springer-Verlag, Berlin Heidelberg, ISBN 978-3-642-36639-0 (eBook)

Demartini, Ch 2014, Performance Management Systems, Physica-Verlag, Berlin Heidelberg, ISBN 978-3-642-36684-0 (eBook)

Fleming, Q & Koppelmann, J 2000, Earned Value Management, Project Management Institute, Newtown Square, 2nd edn., ISBN 978-1-880410-27-1

Hayden, L 2010, IT Security Metrics: A Practical Framework for Measuring Security and Protecting Data, The McGraw-Hill Companies, New York, ISBN 978-0-07-171340-5

Kuster, J & Huber, E & Lippmann, R & Schmidt, A & Schneider, E & Witschi, U & Wüst, R 2015, Project Management Handbook, Springer-Verlag, Berlin Heidelberg (1st English language translation following 3 German editions), ISBN 978-3-662-45373-5 (eBook

Oh, T 2016, The Management Toolbox, Lulu.com, ISBN 978-1-326-52496-8

Pyhrr, P 1977, Zero-Base Budgeting: A Practical Management Tool for Evaluating Expenses, John Wiley & Sons, New York, ISBN 978-0-47103721-7

van Looy, B & Gemmel, P & van Dierdonk, R 2013, Service Management: An Integrated Approach, Pearson Education Limited, Harlow (GB), ISBN 978-0-27373203-7

Weill, P & Ross, W 2004, IT Governance, Harvard Business School Press, Boston, ISBN 978-1-59139-253-8

Wiggers, P & Kok, H & de Boer de Wit, M 2004, IT Performance Management, Elsevier Butterworth-Heinemann, Amsterdam, ISBN 978-0-7506-5926-0