


Course Book



**MANAGEMENT OF IT
SERVICES AND
ARCHITECTURE**

MWIT02-02_E

iu

INTERNATIONAL
UNIVERSITY OF
APPLIED SCIENCES

MANAGEMENT OF IT SERVICES AND ARCHITECTURE

MASTHEAD

Publisher:
IU Internationale Hochschule GmbH
IU International University of Applied Sciences
Juri-Gagarin-Ring 152
D-99084 Erfurt

Mailing address:
Albert-Proeller-Straße 15-19
D-86675 Buchdorf
media@iu.org
www.iu.de

MWIT02-02_E
Version No.: 001-2024-0411

Rafal Wlodarski (Units 1-5)
Created with Midjourney on behalf of IU, 2024, using the prompt: "male and female IT professionals working together in front of computer screens, discussing a software development project inside a modern data center room, multiple LED lights, atmosphere filled with technology, --style raw --ar 16:9 --v 6.0"

© 2024 IU Internationale Hochschule GmbH
This course book is protected by copyright. All rights reserved.
This course book may not be reproduced and/or electronically edited, duplicated, or distributed in any kind of form without written permission by the IU Internationale Hochschule GmbH (hereinafter referred to as IU).
The authors/publishers have identified the authors and sources of all graphics to the best of their abilities. However, if any erroneous information has been provided, please notify us accordingly.

TABLE OF CONTENTS

MANAGEMENT OF IT SERVICES AND ARCHITECTURE

Introduction

Signposts Throughout the Course Book	6
Suggested Readings	7
Required Reading	8
Learning Objectives	10

Unit 1

Introduction to IT Management	11
1.1 IT Management and IT Governance	13
1.2 IT Services	15
1.3 IT Service Management	17
1.4 IT Architecture Management (ITAM)	19
1.5 Reference Models for IT Organizations	21

Unit 2

IT Service Management: Incident and Problem Management	27
2.1 Overview	28
2.2 Service Quality, Service Level Agreements, and Customer Expectations	31
2.3 Problem Management	34
2.4 Software Tools for Supporting Incident and Problem Management	38

Unit 3

IT Service Management: Asset Management	43
3.1 Overview	44
3.2 Using a Configuration Management Database	46
3.3 Asset Lifecycle	48
3.4 Asset Analysis and Risk Management	50
3.5 Interrelation with Procurement and Financial Processes	52

Unit 4

IT Service Management: Supplier Management	59
4.1 Overview	60
4.2 General Sourcing Approaches in ITSM	63
4.3 Evaluating and Selecting Suppliers	67
4.4 Contracting and Service Level Agreements	70
4.5 Monitoring and Controlling Suppliers	72

Unit 5	
DevOps: Connecting Development and Operations	77
5.1 Development and Operation of Software in the Context of IT Management	78
5.2 Characteristics and Shortcomings of a Separation Between Software Development and Operations	81
5.3 The DevOps Idea: An Overview of the Concept and its Elements	84
5.4 DevOps vs. IT Service Management: How to Connect the Approaches	88
Unit 6	
IT Application Portfolio Management	93
6.1 Overview of IT Application Portfolio Management	94
6.2 Application Manual	97
6.3 Portfolio Analysis	100
6.4 Development Planning	103
Unit 7	
IT Architecture Management Basics and Terms	105
7.1 IT Enterprise Architecture	106
7.2 Goals of Enterprise Architecture Management	110
7.3 Processes in the Management of IT Enterprise Architectures	112
Unit 8	
Architecture Governance	117
8.1 Organizational Structure	118
8.2 Policy Development and Enforcement	120
8.3 Project Support	124
Appendix	
List of References	128
List of Tables and Figures	132

INTRODUCTION

WELCOME

SIGNPOSTS THROUGHOUT THE COURSE BOOK

This course book contains the core content for this course. Additional learning materials can be found on the learning platform, but this course book should form the basis for your learning.

The content of this course book is divided into units, which are divided further into sections. Each section contains only one new key concept to allow you to quickly and efficiently add new learning material to your existing knowledge.

At the end of each section of the digital course book, you will find self-check questions. These questions are designed to help you check whether you have understood the concepts in each section.

For all modules with a final exam, you must complete the knowledge tests on the learning platform. You will pass the knowledge test for each unit when you answer at least 80% of the questions correctly.

When you have passed the knowledge tests for all the units, the course is considered finished and you will be able to register for the final assessment. Please ensure that you complete the evaluation prior to registering for the assessment.

Good luck!

SUGGESTED READINGS

These are standard works and comprehensive literature relevant to the respective course. They are not relevant to assessments and may not necessarily be available via the IU library databases. Titles that are available are provided with a link.

GENERAL SUGGESTIONS

Sansbury, J., Brewster, E., Lawes, A., Griffiths, R. (2016). *IT service management* (3rd ed.). BCS. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=cat09158a&AN=iuo.oai.edge.iu.folio.ebsco.com.fs00001148.d35aba84.2516.55e0.9cec.9ccae.cb7fa42&site=eds-live&scope=site&custid=s6068579>

Thejandra, B. S. (2014). *Practical IT service management* (2nd ed.). IT Governance Publishing. Unit 2 <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=cat09158a&AN=iuo.oai.edge.iu.folio.ebsco.com.fs00001148.226a2272.62c9.4ff2.be92.35ffa68f442c&site=eds-live&scope=site&custid=s6068579>

AXELOS Limited (2019). *ITIL foundation: ITIL 4 edition*. The Stationery Office Ltd. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=cat09158a&AN=iuo.oai.edge.iu.folio.ebsco.com.fs00001148.59249fe8.8ba7.5814.9ecb.ddc33a265a0d&site=eds-live&scope=site&custid=s6068579>

REQUIRED READING

These are works that the students are required to read, and their content may be pertinent to assessments. All required readings are accessible via the IU library databases and provided with a link.

UNIT 1

Ahlemann, F., Messerschmidt, M., Stettiner, E., & Legner, C. (2012). *Strategic enterprise architecture management: Challenges, best practices, and future developments*. Springer, 219-225. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=537949&site=eds-live&scope=site&custid=s6068579>

Hanschke, I. (2010). *Strategic IT management: A toolkit for enterprise architecture management*. Springer, 55-60. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=341495&site=eds-live&scope=site&custid=s6068579>

UNIT 3

Agutter, C. (2020). *ITIL foundation essentials ITIL 4 edition: The ultimate revision guide* (2nd ed.). *IT Governance Publishing*, 56-76. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=2440229&site=eds-live&scope=site&custid=s6068579>

UNIT 6

Ahlemann, F., Stettiner, E., Messerschmidt, M., & Legner, C. (2012). *Strategic enterprise architecture management: Challenges, best practices, and future developments*. Springer. Chapters 1 & 2 <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=537949&site=eds-live&scope=site&custid=s6068579>

UNIT 7

Mozsár Kovácsné, A. L. (2017). Reducing it costs and ensuring safe operation with application of the portfolio management. *Serbian Journal of Management*, 12(1), 143–155. <https://search.ebscohost-com.pxz.iubh.de:8443/login.aspx?direct=true&AuthType=sso&db=edsdoj&AN=edsdoj.0786d9c981444205b22a336e9ce68d8b&site=eds-live&scope=site&custid=s6068579>

UNIT 8

Ahlemann, F., Stettiner, E., Messerschmidt, M., & Legner, C. (2012). Strategic enterprise architecture management: Challenges, best practices, and future developments. Springer. Chapter 4, Section 4 <https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=537949&site=eds-live&scope=site&custid=s6068579>

LEARNING OBJECTIVES

Management of IT Services and Architecture begins by exploring the foundations of IT management, including essential components such as IT governance, IT service management, IT architecture management, and reference models for the organization of IT. Each of these components plays a critical role in making sure that IT operations are efficient, effective, and in harmony with the business objectives of the organization.

Later, we'll turn to IT service management (ITSM) problem and incident management. We'll discuss service quality, how service level agreements (SLAs) shape customer expectations, and problem management techniques. Additionally, this course book will highlight software tools that are essential for managing IT incidents, ensuring optimal business support and facilitating user satisfaction.

Our discussion will include the configuration management database (CMDB) - a vital tool that maps out IT assets. We'll deal with the asset lifecycle, exploring ways to ensure effective asset utilization and management from the procurement stage to disposal. We'll touch upon the use of asset analysis and risk management to strategically align assets with business objectives while mitigating risks. Finally, we'll look at the intersection of ITSM asset management with procurement and financial processes. Overall, the aim of this discussion is to help you to understand how to embed IT assets smoothly into the overarching IT strategy.

This course book will also venture into the vital topic of supplier management within ITSM. In a landscape where seamless integration of various services is critical, supplier management emerges as a cornerstone, orchestrating a balanced and synergistic partnership between the enterprise and its various suppliers. Through meticulous monitoring and fostering positive relationships with suppliers, it aims to create a seamless IT service landscape that not only meets the present business objectives but also innovates and adapts to evolving market demands.

UNIT 1

INTRODUCTION TO IT MANAGEMENT

STUDY GOALS

On completion of this unit, you will be able to ...

- understand the fundamental principles and components of IT management.
- explain the key concepts of IT governance, IT service management, IT architecture management, and IT project management.
- evaluate the role of different reference models such as ITIL, ISO/IEC 20000, FitSM, the Zachman framework, TOGAF, and COBIT in IT management.
- understand and analyze the ITIL v4 framework and its components, including the ITIL service value system (SVS) and the four dimensions model.

1. INTRODUCTION TO IT MANAGEMENT

Introduction

Information technology (IT) management is a discipline that involves overseeing, controlling, and orchestrating all the elements of IT. These elements include resources, infrastructure, policies, and processes that are crucial for the delivery of IT services and creating value for both organizations and its customers. IT management refers to how IT leaders strategically guide the use of technology in an organization, from conceptualization to execution. This involves planning, organizing, directing, and controlling the use of technologies and resources in a company.

The cornerstone of IT management is viewing IT as a strategic asset that can drive business value and enable an organization to achieve its objectives. This concept means that IT is not diminished to a support function. For example, when an organization decides to adopt a new technology platform to improve its services or operations, the decision should be based on a strategic evaluation of the potential benefits and risks, not just on the capabilities of the technology itself. This decision-making process is part of IT management, involving aspects like IT governance, service management, architecture management, and the use of various reference models.

In this approach, IT managers evaluate the new technology platform in light of the organization's strategy and objectives, the impact on IT services, the requirements for integrating the platform with the existing IT architecture, and the guidelines and best practices provided by relevant reference models. They then plan the implementation process, oversee the execution, monitor the performance, and make necessary adjustments to achieve maximization of the benefits and mitigation of the risks. This holistic and strategic approach is the essence of IT management.

What is IT?

In order to grasp all the concepts that will be introduced in the course book, let's begin by taking a closer look at what the term "information technology" encompasses. The Cambridge Dictionary defines information technology as "the study and use of electronic systems and computers for storing, analyzing, and utilizing information"(Cambridge University Press & Assessment, 2024). In practice, IT is much more than that and the term encompasses a much broader scope.

The main objectives of IT are to facilitate the efficient and effective processing, storage, retrieval, and communication of information. IT aims to provide reliable, secure, and accessible information systems and services that support the achievement of organizational goals and objectives.

Some of the specific objectives of IT include

- **facilitating communication and collaboration.** IT enables individuals and groups to communicate and collaborate across time and space, using various tools such as email, instant messaging, video conferencing, and collaborative software.
- **enhancing productivity and efficiency.** IT can automate repetitive tasks, streamline business processes, and provide tools that increase productivity and efficiency.
- **supporting decision-making.** IT provides access to information and data analytics tools that enable better, more informed decision-making.
- **enabling innovation.** IT provides a platform for experimentation, prototyping, and testing of new ideas and technologies.
- **ensuring data security and privacy.** IT implements security measures to protect data from unauthorized access, theft, and misuse.
- **providing customer service and support.** IT provides customer service and support through various channels, such as helpdesk, chat, and self-service portals.

Overall, the main objective of IT is to provide a competitive advantage to the organization by leveraging technology to improve its operations, reduce costs, increase revenue, and enhance customer satisfaction.

1.1 IT Management and IT Governance

The history of IT **management** and governance dates back to the early days of computing when it became apparent that the quickly expanding field of information technology required proper management and control. The rapid progress in technology led to increasingly complex IT systems. This resulted in the development of specialized disciplines focused on the strategic management and governance of these crucial systems.

IT management has its roots in the 1960s, when mainframe computers were developed. This was followed by the common adoption of personal computers in the 1970s and 1980s. Organizations started to invest heavily in technology in order to capitalize on its potential. Initial IT management approaches prioritized operational efficiency, cost optimization, and technical expertise, mainly concentrating on hardware and software maintenance.

The beginning of IT governance dates back to the 1990s when many organizations realized that IT needed to be compatible with their overall strategy and goals. The advent of the internet and the subsequent digital revolution underlined the necessity for a more structured approach to IT governance. IT governance grew progressively more important as organizations acknowledged the strategic value of technology and sought to ensure that IT investments were consistent with their goals.

In the current digital era, organizations rely heavily on IT for their everyday operations. The successful management of IT resources has therefore become more and more important for business success. IT management is a complex process that covers organizing, coordinating, planning, controlling, and overseeing IT resources, such as software and data, to meet organizational goals and objectives.

Management

This refers to the process of planning, organizing and controlling resources to achieve specific goals or objectives. It involves monitoring of resources to ensure that they are used effectively.

Efficient IT management requires a team of highly qualified professionals who are responsible for optimizing IT resources and aligning them with overall business strategies. The core responsibility of IT managers is to ensure that IT systems are secure, reliable, and scalable. To achieve this, they need a deep understanding of IT architecture, software development, database management, and project management principles. Apart from that, they must also be proficient in managing IT infrastructure, software, and data resources, as well as IT projects, budgets, and personnel.

Governance

Governance refers to the process and system of decision-making, control, and direction of an organization or a society. It involves establishing policies, procedures, and regulations that ensure the effective and efficient management of resources and activities in line with the organization's goals and objectives.

On the other hand, IT **governance** focuses on ensuring that the IT investments support the organization's goals and objectives, manage risks, and comply with regulatory requirements. This means that a set of policies, processes, and controls must be created for decision-making, accountability, and performance management of IT resources. Strong leadership is critical for effective IT governance, as well as clear policies and guidelines, and effective communication between IT and business stakeholders. Establishing procedures that secure the efficient usage of IT resources while aligning them with business strategies is the key to success.

Several key milestones have significantly influenced the development of IT management and governance, including:

- The creation of frameworks such as the **Information Technology Infrastructure Library (ITIL)**; (Thejandra B. S., 2014), which was first published in the late 1980s, but also the **Control Objectives for Information and Related Technologies (COBIT)** framework (Harmer, 2014), which was introduced in 1996. COBIT offers structured approaches to IT management and governance. These frameworks have evolved over time to accommodate the constantly changing needs of organizations and more advanced developments in technology.
- The rise of **IT service management (ITSM)** in the 2000s marked a shift towards a more customer-focused approach in IT management. ITSM focuses on the delivery of IT services that addresses the needs of both internal and external stakeholders. This approach led to the development of ITSM frameworks and methodologies, such as the ITIL service lifecycle.
- The constantly growing reliance on technology emphasized the importance of managing IT-related risks in organizations. This increasing demand led to the development of professional IT risk management frameworks, such as the ISO/IEC 27000 series (Harmer, 2014), which offers guidelines for information security management.

These milestones were not the end of the development of IT management and governance. The digital revolution of the 21st century and the innovations of recent years have further highlighted the significance of IT management and governance. This era brought forth new challenges, but also many opportunities, such as big data, artificial intelligence (AI), machine learning, and cybersecurity. These developments have expanded the scope of IT management and governance. To cope with these regularly evolving demands, organizations began paying more attention to digital transformation, which meant implementing digital technology in all aspects of business. This fundamentally changed how organizations operate and deliver value to customers. The need for strategic implementation of digital technologies required an increased role of IT, therefore the significance of IT management became more important, as it was responsible for optimizing IT resources,

ITIL

Information Technology Infrastructure Library (ITIL) is a widely recognized set of guidelines intended for managing IT services effectively. It provides detailed advice for ensuring high-quality IT service delivery within an organization.

ITSM

IT service management is a discipline that focuses on the design, delivery, management, and improvement of the ways information technology is used within an organization. The goal of ITSM is to ensure that IT services are aligned with the needs of the organization and its business strategy.

ensuring data privacy and security, and ultimately aligning technological capabilities with business objectives. This transformative period confirmed the importance of IT management and governance in steering organizations through the complex landscape of digital innovation.

IT management and IT governance are critical for organizational success in the current digital world. Effective IT management ensures that IT resources are optimized and utilized properly, whereas an effective IT governance framework helps to fulfill an organization's objectives, manage risks, and comply with regulatory requirements. These areas need to be constantly monitored and evaluated as IT performance and compliance are crucial elements if an organization wishes to remain competitive in the ever-changing digital landscape. Competent IT management teams and an effective IT governance framework are needed to achieve the desired business outcomes.

1.2 IT Services

According to ITIL standards, a service refers to a means of providing value to customers by facilitating desired outcomes without bearing the specific costs and risks involved (Axelos Limited, 2019). Customers typically seek services to achieve specific business objectives, such as conducting transactions and account management activities online or delivering state services to citizens in a cost-effective manner. The value of a service is contingent on its ability to effectively facilitate these outcomes.

Enterprises retain the responsibility of managing overall business costs, but they may choose to delegate ownership and management of defined aspects to internal or external entities with proven expertise in the area. This is a generic concept that applies to the acquisition of any service. Let's take the example of financial planning. Customers may not possess the necessary expertise, time or inclination to manage daily decision-making and investment management activities. In this case, they might engage professional managers to provide services that deliver value by increasing wealth at a price that they deem reasonable. They are willing to let service providers invest in necessary systems and processes to facilitate wealth creation activities as long as they receive satisfactory performance outcomes.

The concept behind IT services has evolved over time, as the complexity of organizations and their requirements have grown. In this context, an IT service refers to the application of technical competencies (for example, skills, processes, and technologies) for the benefit of a customer, typically to enable or facilitate business processes.

In the early days of IT, services were largely confined to the provision and maintenance of mainframe systems. These services were predominantly technical and were delivered by in-house teams. In the 1980s, as personal computers became more widespread in the business environment, the range of IT services expanded. Services such as user support, software development, and network management became more desirable. The expansion of the internet in the 1990s drastically altered the scope and scale of IT services. Services like email, web hosting, and later, e-commerce became significant components of the IT

service portfolio. At the beginning of the new millennium, as the business world was coming out of the dotcom bubble and the challenges of global recession, businesses began to seek control of IT costs and focus on their core competencies. This led to a surge in IT outsourcing, with third-party providers offering managed services that covered everything from infrastructure management to **Software as a Service (SaaS)**. In recent years, the concept of cloud computing technologies has transformed IT services once again. Services are now often delivered on-demand and organizations pay only for what they use. Services have become more scalable and flexible than ever before, with offerings like **Infrastructure as a Service (IaaS)**, **Platform as a Service (PaaS)**, and Software as a Service (SaaS) becoming a standard.

SaaS

This is a software delivery model in which software is hosted on a central server and accessed over the internet. Unlike traditional software that is installed directly on a user's device, SaaS applications are available on-demand and typically accessed through a web browser.

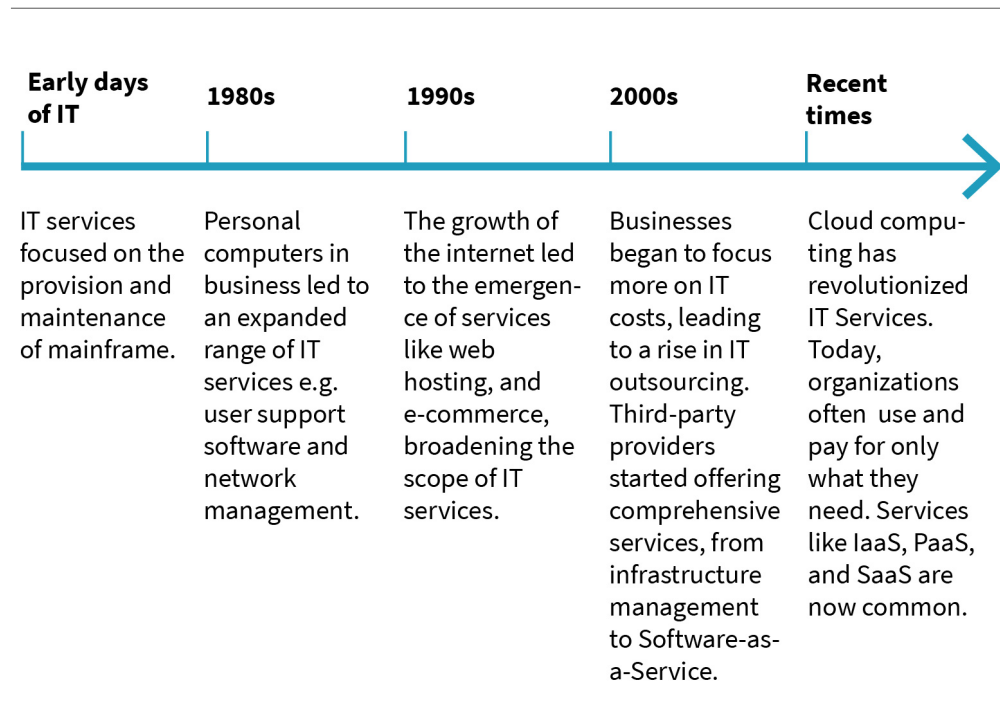
IaaS

This is a model of cloud computing that delivers virtualized computing resources over the internet.

PaaS

This is a type of cloud computing service that offers users a platform to create, run, and manage applications.

Figure 1: The Evolution of IT Services Over Time



Source: Rafał Włodarski (2023).

Today the term “IT services” covers a broad range of offerings designed to support and enable business processes - from providing hardware and software resources to managing complex cloud environments and ensuring cybersecurity. These functions are typically delivered through an IT service management framework that aims to align IT services with the organization's goals and objectives.

One of the primary goals of IT services is to ensure that IT systems are available and reliable, enabling the organization to operate effectively. IT services also need to be secure to ensure the confidentiality, integrity, and availability of data. What’s more, IT services must be scalable, flexible, and able to adapt to the changing needs of the organization and its users. IT service providers use various frameworks, such as the IT Infrastructure Library, to establish best practices for IT service management. Other frameworks, such as COBIT and ISO/IEC 20000, provide additional guidance for the effective management of IT services.

IT services perform a critical function in today's modern organizations, as they encompass all activities related to the design, development, implementation, and maintenance of IT systems (all of which support an organization's operations). Effective IT service management requires an understanding of the organization's business processes, the available IT resources, and the needs of the end users. IT service providers use various frameworks such as ITIL or COBIT and many more to establish best practices for IT service management, ensuring that IT systems are available, secure, scalable, and flexible.

1.3 IT Service Management

The term “IT service management” describes a systematic and competent approach employed by an IT department or a company to deliver reliable and effective information systems and support to meet the requirements of an organization. Although most organizations recognize the benefits of incorporating IT into their internal environment, they often overlook the significance of managing it correctly. Not managing IT equipment and services appropriately can have serious repercussions for an institution. It is crucial for any organization to utilize computers, software, and telecommunications technologies to operate efficiently and stay competitive. If a critical computer system malfunctions, the business may be forced to halt operations if it is not possible to switch to alternative manual processes for a period of time. Computer systems and networks are highly complex and it is not possible for non-technical personnel to maintain or support them independently. Specialized personnel who can grasp how these systems function and know how to manage them are needed. IT services should align with an organization's business strategy and objectives. From a practical perspective, IT service management involves ensuring that the personnel (who could be either outsourced or employed internally) professionally manage and preserve the technical equipment (including computers but also information systems, databases, and software) necessary to run a business.

Let's begin with a formal definition from the most relevant body of knowledge in the IT services domain, ITIL: "The implementation and management of quality IT services that meet the needs of the business. IT service management is carried out by IT service providers through a suitable combination of people, processes, and information technology" (Axelos, 2024, Para 6).

Now, to better understand its meaning, let's break it down into smaller parts. The first part emphasizes the main goal of IT service management, which is to implement and manage high quality IT services and align with business needs. This means that the IT services provided should be reliable, efficient, and effective in supporting the organization's operations and strategic objectives. The second part of the definition identifies the people responsible for delivering ITSM (the IT service providers) and the resources they utilize. These service providers can be either internal teams within an organization, or external vendors who are specialized in providing IT services. With the constantly growing importance of IT departments, the need for effective management of IT services has increased. Maintaining an IT department can be challenging as there are always issues that keep staff occupied and add to their workload. Formal approaches to IT service management arose as a response to common issues encountered by IT departments. These

approaches aim to minimize the negative effects that come from internal and external sources. This raises the question, what are common obstacles encountered by IT departments? The list below provides an outline of common issues.

- **Organizational structure and management**
 - Poorly defined roles
 - Lack of customer support mechanisms
 - Absence of help desk or service desk facilities
 - Understaffing (small IT teams or a single IT person handles all responsibilities)
- **Communication**
 - Business managers and technical staff do not understand each other's needs and workflows, leading to frequent disagreements
 - Technical staff may be too focused on technical matters, leading to a reluctance or inability to understand business needs
 - Frequent disagreements between IT and business departments in terms of service and cost expectations
- **Process definition and management**
 - Overly complex processes that are not defined precisely
 - No service level agreements, vendor agreements, and lack of technical training
 - Absence of proactive measures for preventing IT problems, reactive approach to support, resulting in longer breakdown times
- **Tools and technology**
 - IT department is out of sync with modern business demands due to the use of outdated tools and equipment

This leads us to the following question: How can professional management of IT services make a difference? How can a professional IT management approach help to overcome these obstacles and in turn, transform these challenges into opportunities for growth and development? Let's dive deeper and explore the answers to these pressing questions.

Benefits of Professional IT Service Management

The absence of a professional and proactive IT service department may have significant consequences for a company. Such an organization is vulnerable to devastating and revenue-threatening situations. For example, let's imagine a small online retailer, with a reactive and unprofessional IT service department. One busy shopping day, a server fails, causing the website to go offline. Customers can't make purchases, leading to a direct loss of sales. The unprepared IT department takes a full day to fix the issue, resulting in significant revenue loss for that day. However, if a business implements professional IT service management practices, the likelihood of encountering such damaging situations decreases significantly. Even if a crucial hardware or software component fails, with professional IT service management, it should be able to be restored in a matter of hours.

Poorly organized IT support has direct but also indirect consequences for primary business operations. For instance, clients are unlikely to want to open a bank account in an establishment with frequent computer malfunctions, virus attacks, and shutdowns. Similarly, computerized manufacturing operations are susceptible to significant losses, delays,

and overall business impact if frequent IT disruptions occur throughout the entire process. Given the complexity and relevance of present-day computer systems, it is crucial to establish some measurable and verifiable IT service standards. These standards can help business managers understand the IT department's work scope, outputs, limitations, and budgetary requirements, among other aspects. Both IT and business departments must comprehend that the IT infrastructure has a direct influence on the organization's quality, which affects its financial results.

Professional IT service management, or ITIL, can protect an organization since it provides industry best practices. Alternatively, a company can follow proprietary service processes and methodologies that may not provide adequate protection and lack necessary features. By implementing IT service management, entrepreneurs can rest easy knowing that the IT infrastructure crucial for running their businesses is secure.

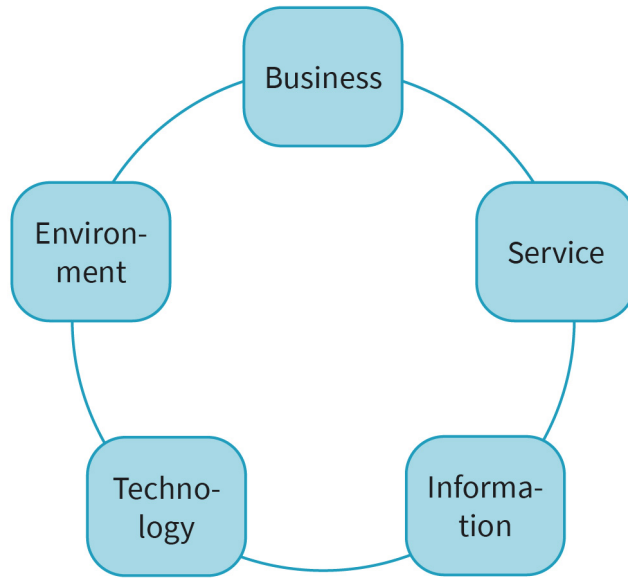
Nonetheless, introducing a quality IT service management system does not require an organization to discard all their current processes or methods and start from scratch. Instead, a gradual implementation of an IT service management system can prove to be more beneficial than a method developed in-house.

1.4 IT Architecture Management (ITAM)

IT architecture management is a discipline that focuses on the structural design of IT within an organization. It is usually perceived as a part of IT governance and helps to form a link between an organization's strategy and its IT infrastructure. According to ITIL management practices, the role and purpose of IT architecture management is to help understand every element of an organization and how they interact, facilitating the accomplishment of the organization's goals. It delivers the guidelines and instruments to manage change in an orderly and flexible manner. As organizations grow more complex, the challenges they face also become more significant. It can become particularly challenging to maintain efficiency, agility, and resilience. IT architecture management can help organizations not to become burdened by convoluted processes, redundant customizations, legacy systems, and badly coordinated third-party contracts.

Effective architecture management covers five main domains: business, service, information, technology, and environment, as shown in the graphic below. These can be integrated into a single architecture for less complex organizations.

Figure 2: Main Domains of IT Architecture Management



Source: Rafał Włodarski (2023).

The list below provides a brief outline of each of these areas:

1. **Business architecture:** This is a roadmap that aligns the organization's capabilities with its strategic goals, with a focus on delivering value to the organization and to its customers. It involves identifying and addressing gaps between the current state and the desired state.
2. **Service architecture:** This offers a comprehensive view of the services the organization offers. It includes the structure, dynamics, and interactions of each service. A service model can serve as a blueprint for other services.
3. **Information systems architecture:** This shows the management of the organization's logical and physical data assets and demonstrates how they are shared. Since information is critical for managers and affects their decision-making, systems must be designed to ensure its completeness, accuracy, and accessibility.
4. **Technology architecture:** This focuses on the software and hardware infrastructure and outlines needs to support the organization's products and services.
5. **Environmental architecture:** This encompasses external factors influencing the organization. It covers all aspects of environmental control and management, which includes impacts from various fields, such as technological, political, economic, and legal.

These main domains of IT architecture management illustrate how architecture management intertwines with all activities in the service value chain. It also plays a critical role in the following value chain activities: plan, improve, design and transition (Axelos, 2019). These and other value chain activities are explored briefly in the list below.

- **Plan:** The role of architecture management in this area is to develop and then maintain a reference architecture that illustrates the present and desired architectures for business, information, data, application, technology, and environment perspectives. This becomes a foundation for all planning actions in the value chain
- **Improve:** The business, service, information, technology, and environment architectures, if properly assessed, can reveal numerous opportunities for improvement.
- **Engage:** The role of architecture management practice here is to understand the organization's readiness to tap into new or under-reached markets, broaden the array of products and services, and adapt more swiftly and appropriately to changing circumstances. It assesses the alignment of the organization's capabilities with all the detailed activities that are necessary for co-creating value for the customers.
- **Design and transition:** As soon as the development of a new or modified product or service is approved, the architecture, design, and build teams must regularly check whether the product/service fulfills the investment targets. Architecture management includes the service architecture, which describes the structure (compatibility of service components) and the dynamics (flow of resources, interaction, and activities) of the service. A quality service model can become a template for multiple services and is crucial to the design and transition activity.
- **Obtain/build:** The reference architectures (business, service, information, technical, and environmental) are crucial in recognition what products, services, or service components must be built or obtained.
- **Deliver and support:** Reference architectures are continually used in the operation, repair, and upkeep of products and services.

As you can see, IT architecture management affects the most important aspects of business. It is necessary to apply good practices as they can help streamline processes, improve efficiency, and reduce costs. By implementing a well-structured IT architecture, an organization can ensure that its IT systems are aligned with its business goals, can adapt to changing business environments, and are capable of supporting future growth.

Applying renowned frameworks can help put components and governance comprehensively together to guide architects in working under and with **enterprise architecture (EA)**. The most popular EA frameworks are: Information Technology Infrastructure Library (ITIL), The Open Group Architecture Framework (TOGAF) and Zachman framework (Mulder, 2023).

Enterprise architecture
This is a strategic planning discipline that an organization uses to create a holistic view of its strategy, processes, information, and IT assets. The purpose of EA is to map and align the business model and processes with IT strategy and infrastructure.

1.5 Reference Models for IT Organizations

In the context of IT, reference models are conceptual frameworks or diagrams that represent an idealized view either of an entire architecture or system, or a specific part of it. They are used not only to organize various aspects of an IT system or infrastructure but also to provide a common language for the stakeholders involved. There are several types of reference models used in IT organizations, and they often form a part of broader architectural frameworks. Some examples are included in the list below.

1. **Information Technology Infrastructure Library (ITIL):** This is a set of best practice publications for IT service management. It provides a framework of processes and procedures, roles, and checklists that are not organization-specific, and can be used by an organization to establish a minimum level of competency. It also establishes a baseline from which to plan, implement and measure.
2. **ISO/IEC 20000:** This is an international standard for IT service management that provides a set of standardized requirements for an IT service management system. With this standard, organizations are able to demonstrate excellence and prove the use of best practices in IT service management. ISO/IEC 20000 includes a set of procedures and processes for efficient management of IT services that aim to ensure the needs of customers are met. It is based on a service lifecycle approach, similar to the one found in ITIL, but also includes requirements for key processes, reporting, and continual improvement.
3. **FitSM:** This is a standard designed specifically for small and medium-sized IT organizations. It provides a set of simple and achievable IT service management requirements, and aims to improve IT services provision to align with business needs. Based on the same fundamental principles as larger frameworks like ITIL or ISO 20000, FitSM focuses on the core aspects of IT service management. It includes aspects like service planning, service delivery, relationship management, problem and incident management, and continual service improvement. It can be perceived as an excellent stepping-stone for organizations that aim to eventually implement more extensive frameworks or standards.
4. **Zachman framework:** This is a framework for enterprise architecture that provides a structured way to view and define an enterprise. This framework is a matrix that consists of six distinct perspectives (planner, owner, designer, builder, subcontractor, and functioning enterprise) against six different aspects (what, how, where, who, when, why). Each cell in the matrix represents a specific perspective of the enterprise.
5. **The Open Group Architecture Framework (TOGAF):** This framework was conceived for enterprise architecture and provides a comprehensive approach to the design, planning, implementation, and governance of IT architectures. It includes an **architectural development method (ADM)**, a set of guidelines for developing architectures, and an enterprise continuum, which is a "virtual repository" of all the architectural assets - models, patterns, architecture descriptions, etc.
6. **Control Objectives for Information and Related Technologies (COBIT):** Created by the Information Systems Audit and Control Association (ISACA), the framework provides a reference model that identifies the major IT processes within an organization and specifies the key activities and key performance indicators for each process.

Architectural development method
ADM is a cycle of phases that system architects go through to develop and manage an enterprise's architecture. It includes eight interrelated phases that are executed in a recurring cycle.

These reference models can serve as a valuable guide for IT organizations as they work to create effective and efficient IT infrastructures that align with company's business goals and objectives.

We will focus on ITIL v4, as this is the most important model for the subject of the course book. The essential elements of the ITIL 4 framework are the ITIL service value system (SVS) and the four dimensions model (AXELOS Limited, 2019).

The ITIL SVS outlines how the organization's different components and operations can contribute collaboratively to generating value through IT services. These elements can be combined in a flexible manner, but the approach needs to be integrated and coordinated to achieve consistency. The ITIL SVS facilitates this integration and coordination, offering the organization a durable, unified, and value-oriented direction. The principal elements of the ITIL SVS include:

- The ITIL service value chain
- The ITIL practices
- The ITIL guiding principles
- Governance
- Continual improvement

The ITIL service value chain introduces a framework that accelerates the creation, delivery, and continual improvement of services. This versatile model identifies six core activities that can be combined in various ways, forming numerous value streams. The flexibility of the service value chain allows organizations to respond to their constantly evolving needs of stakeholders effectively and efficiently. The ITIL practices further amplify the adaptability of the service value chain. Each ITIL practice contributes to several service value chain activities and provides ITSM practitioners with a comprehensive and adaptable toolset. The ITIL guiding principles primarily help to direct an organization's decisions and actions and can promote a shared understanding and consistent approach to service management across the organization. These principles accelerate the establishment of the groundwork for an organization's culture and behavior, influencing everything from strategic decision-making to daily operations.

ITIL v4 defines four essential elements that should be contemplated in every aspect of the service value system for a well-rounded approach to service management (Axelos Limited, 2019). These four dimensions are

1. organizations and people,
2. information and technology,
3. partners and suppliers, and
4. value streams and processes.

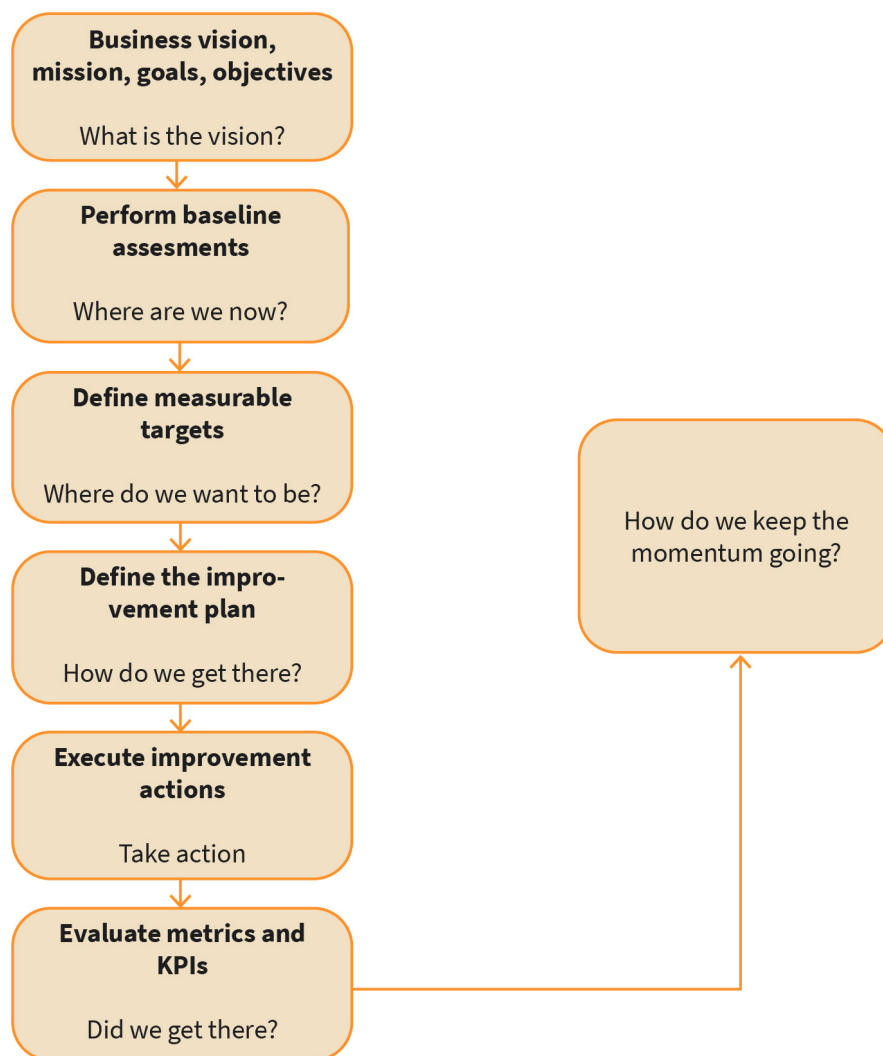
Organizations can maintain a balanced and efficient SVS by paying adequate attention to each of these four dimensions. The optimal functioning of the SVS relies on the harmonious integration of all four dimensions, ensuring a cohesive and comprehensive approach to service management.

The ITIL SVS includes 14 general management practices, 17 service management practices, and three technical management practices, all of which are subject to the four dimensions of service management (AXELOS Limited, 2019). These practices are a set of organizational resources designed for performing work or accomplishing an objective. They encompass a broad spectrum that goes beyond processes to include various resources like people, workflows, information, and technology. The main objective is to provide a more holistic and practical approach to service management.

1. **General management practices:** These 14 practices include concepts of business analysis, project management, and continual improvement that are integral to the entire organization, not just IT services. Examples include risk management, information security management, and continual improvement.
2. **Service management practices:** This category is comprised of 17 practices that focus directly on the delivery and improvement of services. Examples include: incident management, service request management, and service level management.
3. **Technical management practices:** These three practices address specific technological considerations and capabilities for service management. Some examples include deployment management and infrastructure and platform management.

The image below illustrates the objectives of various ITIL practices.

Figure 3: ITIL Practices



Source: Rafał Włodarski (2023), based on AXELOS Limited (2020, p. 23).

Each ITIL practice is described in terms of its purpose, scope, value, guiding principles, and the steps of the process. Each practice also consists of a checklist of roles, responsibilities, and key performance indicators (KPIs). These thorough guidelines ensure that the organizations know how to implement each practice effectively within their specific context. It is crucial to remember that the purpose of ITIL practices is not to implement all of them but to select and adapt those that are most relevant and beneficial for your organization.

The landscape of IT organizations is complex, leading to a need for common language and methodologies. Reference models such as ITIL, ISO/IEC 20000, FitSM, the Zachman framework, TOGAF, and COBIT provide useful frameworks to streamline processes, facilitate communication, and align IT services with business objectives. In this course book, particular attention is paid to ITIL v4 to underline its value as a comprehensive and flexible model. ITIL v4 is designed to provide a unified, value-oriented direction for IT service management. The ITIL service value system (SVS) and four dimensions model, along with the varied ITIL practices, provide the means for organizations to develop a well-rounded, efficient, and adaptable approach to service management.

However, the key takeaway is the notion of adaptability. The purpose of these practices and principles isn't wholesale adoption but rather selective implementation based on the organization's specific needs and context. By doing so, organizations can leverage the power of ITIL and other reference models to enhance their IT infrastructure and better serve their business goals.



SUMMARY

IT management is a broad term that encompasses ITSM, an approach that coordinates IT services with business needs. This alignment is often facilitated by implementing the best practices provided by the Information Technology Infrastructure Library (ITIL).

Another important concept is IT governance, which focuses on adjusting IT investments to business objectives, with the aid of governance frameworks such as COBIT. It ensures that the IT strategy is compatible with the business strategy while appropriately managing risks.

Technical design and structuring within an organization fall under IT architecture management (ITAM). It aligns an organization's strategy with its IT infrastructure while tackling challenges related to efficiency. Frameworks like ITIL, TOGAF, and the Zachman framework assist in managing this complex area.

Reference models like ITIL, ISO/IEC 20000, FitSM, the Zachman framework, TOGAF, and COBIT provide a common language, streamline processes, and synchronize IT services with business objectives. ITIL v4, in particular, offers a comprehensive and flexible approach to IT service

management. The overall goal of IT management is to use these practices to adapt to an organization's specific needs and context, ultimately creating business value.

UNIT 2

IT SERVICE MANAGEMENT: INCIDENT AND PROBLEM MANAGEMENT

STUDY GOALS

On completion of this unit, you will be able to ...

- understand the basic principles of service quality in IT service management (ITSM) and how they relate to problem and incident management.
- explain the importance of service level agreements and how they set customer expectations for IT services.
- describe the main steps and approaches in problem management.
- recognize different software tools that support problem and incident management and get to know their main features.
- grasp the connection between what customers expect, how services are delivered, and the main role of ITSM in making sure they match up.

2. IT SERVICE MANAGEMENT: INCIDENT AND PROBLEM MANAGEMENT

Introduction

IT service management (ITSM) includes several processes, of which incident and problem management are two key components. The process responsible for managing the lifecycle of all incidents is called “incident management”. Its primary objective is to return the IT service to users as quickly as possible. In this context, incidents can be defined as disruptions in the normal operation of an IT service. They can be something as minor as a single user being unable to access a document or as major as a complete network failure. Problem management is a process that aims at managing the lifecycle of problems. Its primary objective is to prevent incidents from happening, but also to minimize the impact of incidents that cannot be prevented. Problems can be defined as the causes of one or more incidents and they are typically identified through the incident management process.

In the vast ecosystem of information technology, these two key components (ITSM problem and incident management) can be seen as the first responders and investigators, respectively. When a digital problem arises, whether it’s a temporary software hiccup or a full-blown network breakdown, incident management acts like a quick response team, ensuring immediate resolution to keep business operations running smoothly. Problem management plays the role of the detective of the IT world. After an incident is resolved, this facet of IT management helps an organization to delve into the "whys" and "hows" behind the occurrence. It's not just about quick fixes but understanding the root causes and implementing strategies to prevent similar issues in the future.

Together these two elements ensure the stability and resilience of an organization's IT infrastructure. While adequate incident management offers the peace of mind that any disruptions will be swiftly handled, problem management brings the assurance of continuous improvement and long-term reliability. In the ever-evolving landscape of IT, these dual functions serve as the pillars that uphold the structure, ensuring both immediate responsiveness and strategic growth.

2.1 Overview

To get a better understanding of both terms, let’s imagine an organization that has an email server that occasionally crashes, making it impossible for employees to access their emails during these times. Each time this happens, the IT department gets many individual reports about the email system being down. Each of these reports is an incident. The IT team, using their incident management process, treats each report as an isolated event. They work quickly to get the server up and running again - resetting the server, for example - to minimize downtime. Their primary focus is to restore the service as quickly as possible so the employees can continue their work. However, after several such incidents, it

becomes clear that there is a recurring problem with the email server - it crashes under high traffic. This is where problem management comes into play. Problem management is concerned with identifying and resolving the root cause of these incidents. The IT team starts a thorough investigation, they might find out that the server is outdated and cannot handle the current load anymore.

The solution might be to upgrade the server or increase its capacity to handle higher traffic. By resolving this problem, the IT team can prevent similar incidents from occurring in the future, leading to a reduction in downtime, less disruption for employees, and overall, an improvement in the service quality. Thus, while the incident management process dealt with the immediate effect of the server issue (i.e., employees not being able to access their emails), the focus in terms of problem management was on addressing the underlying cause to prevent or reduce such incidents in the future. While incident management is concerned with restoring service and addressing user issues promptly, problem management is applied to dig deeper, finding the root cause of the incidents to prevent future occurrences. Both of these management processes work simultaneously to ensure that IT services are provided effectively and efficiently to all users in an organization.

Incident management aims at restoring normal service operation as quickly as possible when disruptions occur, to minimize impact on business operations. An incident, in this context, is an event that causes or may cause an interruption or a reduction in the quality of a service. The process begins with incident identification and logging, followed by incident categorization and prioritization based on its impact and urgency. Then, the appropriate diagnostic procedures are carried out for incident resolution, and the service is restored.

Problem management is a more proactive and investigative process. Its objective is to identify the underlying causes of incidents (referred to as “problems”) and manage them. The main goal of problem management is to keep incidents from happening, and to reduce the negative impact of incidents that cannot be avoided. This process involves problem identification, logging, categorization, investigation, and resolution. It's important to note that while incident management focuses on addressing the symptoms (i.e., the incidents themselves), problem management addresses the root cause to prevent recurrence.

Both these processes are crucial in an ITSM framework. While incident management ensures business continuity, problem management is key for improving the overall quality of services and reducing the number and severity of incidents over time. Incident management is part of the service operation phase of the ITSM core lifecycle. It is symptom driven, and the only concern is speed of response and the continuation of the business.

The main responsibilities involved in incident management include:

- Attending to the incident as soon as possible
- Classifying incidents and matching against a known error
- Resolving the incident or providing a workaround until the incident can be resolved
- Alerting support groups if the incident cannot be resolved

- Keeping the end user informed about the status
- Closing the incident

Incidents have different levels of priority. Not all incidents can be given equal attention or be considered equally as urgent. Certain incidents can wait, while others require immediate attention. Accordingly, incidents are classified based on predefined priority levels. The service desk is responsible for determining incident priorities as it receives them. Different organizations have different priority levels, for example:

- High, low, medium
- Immediate, urgent, moderate, ordinary
- Severity 1, 2, 3, 4 (1 being the most urgent)
- Critical, medium, routine

Realistic categories and turnaround times should be established in cooperation with the business, customers, and end-users. Incident management shouldn't pledge to address every incident within a short timeframe to appear effective. Such a promise is not feasible, nor is it recommended. It is highly advisable to classify incidents based on their **business impact** so that incidents with potential business implications receive higher priority. Prioritization is crucial to proper allocation of resources and personnel. Otherwise, staff might be preoccupied with a regular incident, while a high-priority incident remains unresolved.

Business impact
This is anything that affects, or has the potential to disrupt, the business.

In addition to business priorities for incidents, they also have to be categorized according to equipment, end users and so on. Typical examples of categorization include (Thejandra B. S., 2014):

- Desktop computer incidents
- Server computer incidents
- Telecommunication incidents
- Software incidents
- Security violation incidents
- Virus incidents
- Password issues (can be routed to request fulfilment)

Problem management is part of the service operation phase of the ITSM core lifecycle. Recurrence of the same incidents leads to lost time and frustrated users. Effective problem management can prevent this from happening. The main responsibilities involved in problem management are in the following list:

- Problem control
- Error control
- Proactive prevention of problems
- Identifying trends
- Management reports
- Major problem reviews

Problem managers can take on both proactive and reactive roles. In proactive roles they identify and solve problems before incidents start occurring. An example of a proactive task is to inform all tech staff to apply a video driver update on all computers to prevent potential monitor failures. In a reactive role, they identify and solve problems after incidents start appearing, informing all the tech staff to apply a video driver update, on all computers, after noticing several monitors are failing due to a video driver fault.

Incident and problem management are vital to an organization's IT infrastructure. Incident management ensures business continuity, while problem management helps to improve service quality, reducing incidents over time. Both processes underline the importance of efficient IT service management in achieving business objectives.

2.2 Service Quality, Service Level Agreements, and Customer Expectations

Understanding and managing service quality, service level agreements, and customer expectations is crucial in IT service management, particularly in the domains of incident and problem Management. In the context of ITSM, a service is a means of delivering value to customers by helping them achieve their desired outcomes without the need to manage specific costs and risks. Essentially, it's what an IT organization provides to its users to support their business activities. This can include, for example:

- Cloud services such as data storage or computing resources
- Software as a Service (SaaS) such as customer relationship management (CRM) tools or human resources management systems
- Infrastructure services offering server space, bandwidth, or virtual machines

It's crucial to understand that while these are services provided to the user, ITSM processes like incident management or problem management are not considered IT services. Instead, these are processes or approaches used by IT teams to ensure the smooth delivery and support of the above services. For instance, while a hosted email service is an IT service, the process to handle email outages (incident management) isn't a service itself but a support mechanism for that service.

Service Quality

Service quality refers to the effectiveness and efficiency of IT services in meeting the requirements and expectations of the business and its end users. It ensures that the services provided are reliable, responsive, and aligned with business objectives. Both incident and problem management are closely tied with the quality of service; high-quality services experience fewer incidents and problems, and when they happen, they are managed more effectively and efficiently.

Service level agreements (SLAs)

SLAs are formally negotiated agreements between an IT service provider and its customers. They list all the specific services that will be provided, the standards to which those services should be delivered (including uptime and responsiveness), and the solutions or penalties if these service levels aren't met.

Within the broader scope of service delivery, SLAs often set out specifics like response times, escalation processes, and resolution durations. While these can relate to activities in incident management and stipulate periods for activities like root cause determination in problem management, it's crucial to understand that SLAs are not exclusively designed for these processes; they pertain to the overall service being provided. Typically, an SLA starts with an introduction, outlining the agreement, parties involved, and the effective date. Following this, there is a section detailing the scope of services to be provided. Metrics to measure the service performance such as uptime, speed, and responsiveness are then outlined. The responsibilities of both the provider and the customer are defined. There is a section for service management detailing how service delivery, monitoring, reviewing, and reporting will be conducted. Procedures for escalations and complaints are set forth, along with security and compliance measures. The mechanism to review and audit service performance is described, followed by the terms and conditions outlining pricing, service level, and termination of the agreement.

The SLA lifecycle starts with the creation of the SLA, based on negotiations between the service provider and the customer. Once agreed upon, the SLA is implemented. The service parameters defined in the SLA are constantly monitored. There is a regular review of performance against the SLA, identifying areas for improvement. Necessary amendments or updates to the SLA are made based on reviews and changing requirements or circumstances. At the end of the SLA term, the agreement is either renewed with updated terms or terminated.

A simple example could be an SLA for a systems applications and products in data processing (SAP) software solution, where the system is expected to be available 99.5% of the time during business hours, with incident response time within two hours for high priority issues, and resolution time within 24 hours for critical issues. Maintenance would be scheduled between 12:00 AM to 4:00 AM on Sundays. Another example could be for a printer service, where the availability is 98% during business hours, with a technician onsite within four hours of a reported issue, and a quarterly preventative maintenance check is performed.

Operational level agreements (OLAs)

OLAs are internal (within the IT department of a company) agreements that outline how different departments or teams within an organization will work together to ensure services are delivered smoothly. Unlike SLAs, which are made with external suppliers or customers, OLAs are created internally to make sure everyone involved knows what's expected and who is responsible for what.

For example, if there is an OLA between the IT Support and Network Management teams it means that this agreement ensures smooth network operations by outlining how the two teams will handle network issues. IT Support will report network problems to Network Management within ten minutes of detection. Network Management will acknowledge the issue within 15 minutes and aim to resolve it within two hours or provide a progress update. This agreement will be reviewed quarterly to ensure it remains effective and relevant.

Underpinning contracts (UCs)

UCs are formal agreements between an IT service provider and a third party, usually a supplier or service provider. These contracts outline the responsibilities, deliverables, and standards the third party should adhere to while providing services or goods. UCs are crucial to ensure that the IT service provider can meet its SLAs with its customers.

Let's imagine a UC between XYZ Corp and ABC Internet Service Provider. This contract outlines the terms for internet service provision by ABC to XYZ Corp to ensure uninterrupted internet connectivity. ABC agrees to provide a 99.9% uptime guarantee and to resolve any connectivity issues within a maximum of four hours from the time reported. In case of service disruption, ABC will provide a detailed report and compensate with service credits as per the agreed terms. This contract will be reviewed annually or upon major service changes to ensure it continues meeting the operational requirements of XYZ Corp.

Customer expectations

Customer expectations refer to what the business and its end users expect from the IT services. These expectations could relate to the reliability, availability, responsiveness, and overall performance of IT services. In terms of incident and problem management, customers generally expect quick response and resolution times so the impact on their operations is minimized. They expect clear communication about the status of their incidents and description of the steps that are being taken to resolve them. Furthermore, customers expect the IT service provider to learn from incidents and take proactive measures (through effective problem management) to prevent recurrences.

Service Level Management (SLM)

Ensuring service quality, negotiating SLAs, and fulfilling customer expectations are all a part of SLM. This term refers to the method of establishing, agreeing upon, recording, and controlling an effective IT service aligned with business needs. It involves managing SLAs, OLAs, UCs, and related activities such as periodic reviews, updates, as well as publishing documents and sharing them with all the stakeholders. The main goal of SLM is to ensure that the IT services delivered are of a high quality and acceptable to the business and to the end users, while at the same time maintaining cost-effectiveness. SLM is incorporated during the service design stage of the ITSM core lifecycle.

For a real-world example, let's consider a multinational corporation with a complex IT infrastructure. To ensure the quality of their IT services, they've set up strict incident and problem management processes. However, despite these measures, they have started to

receive complaints from users about slow response times and frequent crashes of the application. The IT service provider conducts an investigation and discovers that the existing SLAs do not factor in the severity and frequency of these incidents adequately. For example, the agreed upon help desk availability is not sufficient – longer hours and more employees are needed. This misalignment has led to a decrease in perceived service quality and dissatisfaction among end users. In response, the IT service provider renegotiates the SLAs, setting more stringent guidelines for incident response and resolution times, and tighter controls on application uptime. They also implement a more proactive problem management process to identify and resolve the root causes of frequent issues. The revised SLAs and enhanced problem management process lead to a significant reduction in the number and severity of incidents, and the response times for critical incidents are halved. The perception of service quality among end users significantly improves, demonstrating the importance of properly managed SLAs and proactive problem management in meeting and exceeding customer expectations.

As we have seen on the example above, SLAs are crucial for providing quality IT services. In the field of IT, it's typical for multiple customers to share a single service, and conversely, for a single customer to use a variety of services. This creates flexibility in the creation of SLAs. They can be customer-based, where an SLA encompasses a range of services provided to a specific customer; or they can be service-based, in which case a common SLA covers all customers of particular service. Customer-oriented SLAs are agreements that are made with specific customers. This makes the relationship between IT and the customer simpler, but this kind of SLA can be complex and hard to manage and additionally it may not meet every specific customer need. Service-oriented SLAs are agreements with all users of a specific service, practical when all customers receive the same service level. However, it may become complicated when different service levels are necessary or wanted by different customers. Multi-tier SLAs are used when a corporate-level SLA covers common service aspects, with customer-level and service-specific SLAs addressing issues related to individual customers or services.

Maintaining service quality, properly managing SLAs, and meeting customer expectations are essential in ITSM, particularly in incident and problem management. When properly implemented and managed, these elements can significantly improve service quality, enhance customer satisfaction, and ensure the alignment of IT services with business objectives.

2.3 Problem Management

A problem is an incident, or multiple incidents, for which the root cause is not known. Sometimes problems can be discovered because of multiple incidents exhibited in similar systems, for example, a computer occasionally not booting-up is an example of an incident but the same computer (or all similar models) not booting-up every Monday morning is a problem that needs further investigation. Until someone finds a solution, those end-users will face the same issues every week. While the main goal of incident management is to restore services, problem management targets the root cause of the issue. Problem

management oversees all of the IT issues, conducting root cause analysis and coming up with solutions. The responsibility of problem management persists until resolutions are executed through change management and release and deployment management.

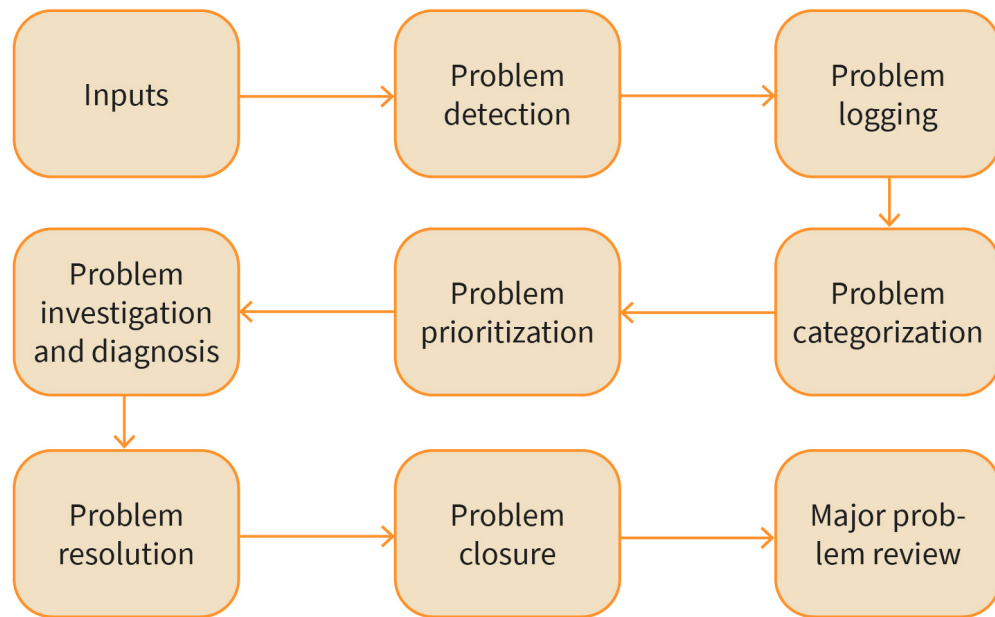
Problem management brings value to an organization by preventing, alleviating, and managing the negative effects of problems. It enables services to be more reliable and resilient, leading to increased availability. The main responsibilities of problem management are:

- preventing problems and resulting incidents from happening
- avoiding repetition of incidents
- reducing the impact of incidents that cannot be prevented

Problem management encompasses both reactive and proactive components, much like many other processes. The reactive role of problem management handles the entire problem lifecycle from identifying issues to eliminating them. It is achieved by determining the root causes and implementing necessary changes to avoid recurrence. The proactive aspect of problem management, in contrary, aims to prevent future incidents where possible, or at least diminish the impact of unavoidable incidents.

A structured process flow for problem management (as shown in the graphic below) ensures that problems are handled systematically and effectively. Throughout the stages of this process, it is vital to keep stakeholders informed of the status of the problem and any impact on services. It's also important to note that problem management works closely with other ITSM processes, particularly incident management and change management.

Figure 4: Problem Management Process Flow



Source: Rafał Włodarski (2023).

The steps of problem management process flow are described in closer detail in the list below (Sansbury, J., Brewster, E., Lawes, A., Griffiths, R., 2016):

- **Inputs:** Data for problem management can come from different areas including incident management, event management, the service desk, supplier information, proactive problem management, and other processes.
- **Problem detection:** Identification of problems can take place through various means such as the service desk, secondary support, service management tools, proactive problem management, and other processes.
- **Problem logging:** It is vital to document all pertinent details of a problem, including all incidents precipitated by the problem and the time they occurred.
- **Problem categorization:** The same system of classifying problems should be used for both incident and problem management.
- **Problem prioritization:** Problems should be prioritized in a manner similar to incidents, taking into consideration factors like frequency of recurrence.
- **Problem investigation and diagnosis:** The objective here is to uncover the root cause of the problem. Resources should be allocated based on the problem's priority, which should be reassessed over time.
- **Problem resolution:** A permanent solution should be applied as soon as it becomes practical, taking into account aspects such as cost, risk, and the possibility of service disruptions. A request for change (RFC) should be initiated for any necessary changes.
- **Problem closure:** Problem records should only be concluded after a change has been effectively implemented and the problem resolution has been verified, potentially through testing. Sufficient time should be allocated to confirming the success of the fix.

- Major problem review: Following a major problem, a review should be conducted, with the focus being on understanding and learning from the experience, rather than finding fault. This review should evaluate what was effective, areas needing improvement, preventative measures, and tactics to lessen the impact of future problems.

Root cause analysis (RCA) can be useful for avoiding the repetition of problems. It is a methodical process aimed at identifying the underlying causes of problems or incidents within an IT environment. The primary goal is to discover the core issues facilitating these disruptions, rather than just addressing the symptoms or immediate concerns. By identifying and resolving the root cause, organizations can prevent similar incidents from recurring, thus improving the overall quality and reliability of IT services. An example of an RCA is outlined in the eight steps listed below:

1. Incident identification: An organization's IT team notices a pattern of network disruptions that frequently occur during peak usage hours, causing service outages and affecting customer satisfaction negatively.
2. Incident logging and analysis: All incidents are logged meticulously in the incident management system. Data related to network traffic, system configurations, and user reports during the disruptions are collected and analyzed. The incidents are categorized based on their severity, impact, and frequency in order to understand the scope of the issue.
3. Initial diagnosis: An initial diagnosis is performed by the IT team, which reveals that the network disruptions predominantly occur during peak hours when network traffic is at its highest.
4. Root cause analysis: The team employs RCA techniques, such as the 5 Whys, to delve deeper into the issue. They discover that the network's current bandwidth is insufficient to handle the surge in traffic during peak hours, leading to disruptions.
5. Remediation and corrective actions: The team devises a short-term workaround using load balancing techniques to manage network traffic more effectively during disruptions. They also plan a long-term solution which includes upgrading the network infrastructure, increasing bandwidth capacity, and implementing redundancy measures to handle traffic surges.
6. Preventive measures: Proactive measures are implemented to prevent future disruptions. This includes enhanced monitoring of network traffic, capacity planning, and the implementation of quality of service (QoS) mechanisms to prioritize critical traffic and ensure network reliability.
7. Documentation and reporting: The entire process, from incident identification to preventive measures implementation, is documented meticulously. This documentation is shared with relevant stakeholders and also updates the organization's knowledge base. A report highlighting the RCA findings, recommended improvements, and the impact of implemented measures is prepared and shared.
8. Continuous improvement: The network performance is continuously monitored to ensure the effectiveness of implemented measures. The IT team analyzes incident trends, gathers user feedback, and conducts periodic reviews of the RCA process to identify further improvements. Lessons learned from the incident are used to refine the organization's problem management practices, ensuring ongoing enhancement of network reliability and service delivery.

Known error

A known error is an incident for which there is a solution.

To accelerate the process of solving encountered problems it is important to have a database in which solutions, root causes and workarounds are available and documented. This is the **known error** database, and it contains solutions of all internal, and possibly some external, known errors. The known error database is used to improve efficiency in handling incidents and problems. When a new incident or problem arises, service desk personnel can check the database to see if the issue has occurred before and, if so, what steps were taken that helped to mitigate or resolve it. This can often lead to quicker resolution times and more consistent service.

A typical entry in a known error database should include information like the symptoms of the error, the circumstances under which it occurred, and the status of efforts to eliminate the error. Maintaining a known error database as part of problem management is a recommended practice in ITIL methodologies. It is recommended because it helps resolve incidents faster. When a known error occurs, solutions or workarounds from the database can be used for quicker resolution, improving service availability and user satisfaction. The database promotes knowledge sharing among support staff, enabling them to handle issues more efficiently. It also helps with analyzing recurring issues to find permanent solutions, aiding continuous improvement. By facilitating faster resolution and efficient workflows, operational costs are reduced.

Problem management has a crucial role in an organization because it enhances the reliability and resilience of services. This is achieved by preventing, reducing, and managing the impact of problems. Both reactive and proactive measures are included in this process. Problem management can be used to track the lifecycle of problems, identify root causes, and make necessary changes to prevent recurrence, while also proactively working to prevent future incidents or mitigate their impacts.

2.4 Software Tools for Supporting Incident and Problem Management

Software tools for incident and problem management, commonly referred to as ITSM tools, are vital for IT organizations to efficiently handle and resolve issues. These tools aid automation and streamlining of the incident and problem management processes, ensuring that issues are addressed promptly, root causes are identified, and long-term solutions are put into place to prevent future incidents. There are many options, and each manager can choose those that suit their needs and preferences. As the market for IT services is constantly growing, so are software tools that support incident and problem management. When choosing the best tool, the following factors should be considered:

- **Cost:** In many cases there will be the initial cost of hardware and software, as well as maintenance after the system goes live. It should be examined so that it doesn't exceed budget limitations.
- **Complexity:** This aspect should be considered carefully. On the one hand, a standalone deployment can increase complexity due to the addition of more servers to manage. However, on the other hand, looking for a combined deployment can complicate the

system's configurations. Hence, it is crucial to strike a balance by assessing the available in-house expertise. It is important to ask the question: Is there a team in place capable of managing the infrastructure and accommodating the addition of more servers? Is there an experienced administrator available who can effectively manage a complex software tool?

- **Security:** This is especially important if service management projects are publicly accessible and software projects are internal only. Extra steps will be needed to ensure data security if both public and private projects are in the same instance.
- **Scalability:** This is a factor that can easily be overlooked. When deploying such a tool, it's crucial to ensure that existing hardware can accommodate the additional load stemming from service management projects. This consideration becomes particularly important if projects are to be accessed publicly.
- **Maintenance:** Once an organization starts using a service management software tool, they will likely make customizations and add third-party apps, which can add complexity at upgrade times.

When all of the points above have been defined, the most suitable service management software tool can be chosen. An overview of well-known tools is shown in the table below.

Table 1: Service Management Software Tools

Software tool	Description
Jira Service Management	This Atlassian product is often used in software development settings and integrates well with other Atlassian products. It offers robust tracking, customizable workflows, and integration with knowledge base tools.
ServiceNow	This is a platform that supports a wide range of ITSM processes, including incident and problem management. It offers features such as automation of routine tasks, integration with other IT systems, and comprehensive reporting and analytics.
BMC Remedy	This is an enterprise-oriented tool that offers strong multi-channel support, knowledge management capabilities, and predictive service management features.
Freshservice	This is a user-friendly, cloud-based ITSM solution that provides broad incident and problem management capabilities, including ticketing, automation, and service catalog management.
Zendesk	While more commonly known for its customer support solutions, Zendesk also offers a suite of ITSM tools. These include incident and problem management, ticketing, reporting, and integration with other systems.
SolarWinds Service Desk	This tool provides strong capabilities for incident, problem, and change management, along with risk detection, advanced reporting, and automation capabilities.

Source: Rafal Wlodarski (2023).

There is no one “best” software tool, it all depends on the company needs. These tools typically provide all, or some of the previously mentioned capabilities that aid service management. Examples of the capabilities typically provided by service management software tools are listed below:

- **Ticketing system:** Helps to track incidents/problems from their creation to resolution
- **Workflow management:** Allows for the creation of automated workflows to streamline and standardize the handling of incidents/problems
- **Knowledge base:** Stores documentation and solutions to common issues to assist in problem resolution and to encourage self-service
- **SLA management:** Helps monitor the SLA compliance, ensuring that the IT team meets its performance standards
- **Reporting and analytics:** Provides insights into common issues, resolution times, and team performance
- **Collaboration tools:** Enables team members to work together more effectively on resolving incidents and problems
- **Integration capabilities:** Allows integration with other systems used in the organization, such as monitoring systems, to allow for faster detection and resolution of incidents/problems

ITSM tools aid in automating various ITSM processes, ensuring effective service delivery. One of the key advantages is centralized information sharing. When all information regarding an incident is stored in one place, it promotes real-time sharing and updating of incident data, which is critical for timely resolution. This centralization eradicates the problem of scattered data across different platforms, which could lead to confusion and delays. Another vital aspect of ITSM tools is incident monitoring. Monitoring features can help to keep track of incidents from the moment they are reported until they are resolved. This continuous monitoring ensures that all incidents are accounted for and handled according to the SLAs.

It’s also important to pay attention to time measurement, as it is essential to assess the efficiency of the IT service management process. By automatically tracking the time, ITSM tools help to analyze performance and identify areas for improvement. Incident documentation is facilitated by these tools, creating a structured documentation process that is easily accessible and searchable. Documenting incidents, including their cause, impact, resolution, and other relevant details, is critical for future reference and analysis. Workflow automation is another feature of ITSM tools that reduces manual intervention, thereby speeding up the incident resolution process and ensuring consistency. Automating workflows enables auto-routing of incidents, notifications, and other automated actions based on predefined rules.

Several additions can provide benefit to users – from entry level employees to management. Providing a user interface for stakeholders to check the status of incidents can improve transparency and customer satisfaction. ITSM tools usually come with a user-friendly frontend, allowing users to track the progress of their incident tickets and receive updates. Another useful addition is a management dashboard that provides a high-level

view of ITSM performance, including incident statistics, SLA adherence, and other crucial metrics. These dashboards allow management to gain insight into the operations and make informed decisions.

Software tools for incident and problem management are integral elements in the effective functioning of IT organizations. These tools facilitate efficient issue resolution by streamlining and automating the incident and problem management processes. They play a vital role in timely issue resolution, identification of root causes, and implementation of long-term preventative measures. The most prominent tools currently on offer provide a variety of features, each suitable for different organizational needs. However, no single tool can be deemed the best for every organization. The choice should be influenced by the specific needs, budget, size, and complexity of each organization. Thus, selecting the right ITSM tool is a strategic decision that requires careful consideration and planning. It is a crucial step towards improving the efficiency and effectiveness of an organization's incident and problem management processes.



SUMMARY

ITSM is structured around two core processes: incident management and problem management. Incident management addresses IT service disruptions and prioritizes them based on factors like business impact and urgency for swift restoration. Problem management delves deeper, investigating the root causes of these disruptions to prevent their recurrence. It ensures reliable services by addressing problems both reactively, after they occur, and proactively, to prevent them. For example, while incident management can be applied to restore a failing email server, problem management processes might lead to a recommendation of an upgrade to prevent such failures. Both processes, guided by the ITIL framework, ensure IT services align with business needs. SLAs further define and maintain these service standards.

Effective problem management employs stages from detection to resolution, using tools like known error databases for quicker solutions. ITSM tools are vital, automating incident and problem processes. Considerations for the selection of such tools include cost, complexity, and adaptability. The right tool ensures efficient management and boosts service resilience. ITSM tools and practices are pivotal for aligning IT services with business needs, ensuring quick issue resolution, and enhancing long-term service reliability.

UNIT 3

IT SERVICE MANAGEMENT: ASSET MANAGEMENT

STUDY GOALS

On completion of this unit, you will be able to ...

- understand the principles of IT service management (ITSM) asset management.
- describe the distinct stages of the asset lifecycle.
- understand the integral relationship between ITSM asset management, financial management, and procurement processes.
- discuss the role of configuration management databases (CMBDs) within ITSM asset management.

3. IT SERVICE MANAGEMENT: ASSET MANAGEMENT

Introduction

Think of ITSM asset management as the backstage manager of the grand show called "information technology." It's all about keeping track of each piece of tech equipment and software an organization uses, from the moment it's bought until the time comes to phase it out. Imagine a business like a bustling coffee shop. Computers process orders, keep track of inventory, handle payments, and even play that chill background music. ITSM asset management ensures that each device or software, like the computer system taking your coffee order, runs smoothly, gets updated when needed, and is replaced when it starts slowing things down. Beyond just keeping tabs, it's also about making sure everything fits the business goals and follows the rules. For instance, if a software update ensures credit card details are safer, ITSM asset management is on it. It's the component responsible for ensuring everything in the IT world of a business runs like a well-oiled machine.

3.1 Overview

IT service management asset management, which is often referred to as IT asset management (ITAM), is a set of practices that combines financial, inventory, contractual and risk management responsibilities. It helps manage the lifecycle and costs of IT assets of the entire organization. IT assets encompass all software and hardware components present within the business setting. ITAM is an important part of an organization's strategy. It involves the tracking and management of every component in the organization's IT infrastructure. It ensures that both the tangible and intangible assets are properly catalogued, valued, located, maintained, upgraded or disposed of when the appropriate time comes.

The two main objectives of ITAM are to identify and control all items of interest and to manage and protect the integrity of assets (Sansbury, J. et al, 2016). Asset management practices are concerned with overseeing service assets throughout their entire lifecycle. They help an organization to maintain a comprehensive inventory of assets, documenting the person accountable for their management. These practices encompass the holistic management of IT and service assets, beginning from the acquisition phase, passing through maintenance, and eventually leading to their disposal.

ITAM is closely linked with configuration management, which goes a step further by providing information about how different parts are connected to each other. This information forms the foundation for other processes, particularly incident management, problem management, availability management, and change management, making it a crucial aspect of effective service management solutions. When a change is suggested, detailed configuration information allows for a swift and precise evaluation of how this change will

influence services and components. The process responsible for both asset management and configuration management is called service asset and configuration management (SACM) and it is an integral part of the ITIL framework. In the ITIL framework, the objective of ITAM and configuration management practices is to maintain information about the configuration items necessary to providing IT service, including their relationship.

Now, let's explore a key aspect of this process - configuration items (CIs). There is one main difference between CIs and assets. An asset is any valuable item in an organization, such as hardware or software, while a CI is a specific type of asset managed within a configuration management system (due to its influence on service delivery). For instance, a laptop is an asset, but when detailed records of its configurations and relationships to other items are maintained, it becomes a CI. So, all CIs are assets, but not all assets are CIs. The complexity, size, and type of a CI can vary significantly, ranging from entire systems like a mainframe computer to minor hardware components like modems. As part of the configuration management process, data associated with these CIs is stored in the **configuration management database (CMDB)**, ensuring a structured and organized approach to their tracking within the IT landscape. CIs play a central role in configuration management, and they are directly link to ITAM. However, while ITAM focuses on the tracking and control of physical and virtual components of the IT infrastructure, configuration management zeroes in on a specific category of these components: CIs.

Configuration Management Database (CMDB)

This is a repository that acts as a central source of information on the IT infrastructure's hardware, software, and service assets. It keeps track of these assets and their relationships, aiding the management and control of IT services.

It's advisable to initiate the configuration management process by establishing high-level CIs. Additionally, pinpointing essential services and their components is a strategic starting point for effective configuration management. Examples of critical configuration items include:

- Essential PCs, laptops and servers
- Key LAN switches, routers, and telecommunications equipment
- List of all important business applications and corresponding documentation
- Significant databases and their connections
- Standalone devices, including firewalls, security equipment, modems, and printers

Critical CIs are the essential components of an IT infrastructure that enable maintaining the functionality, performance, security, and availability of an organization's IT services. These CIs usually have a significant impact on the delivery of critical services, so their performance and health are closely monitored. Critical CIs can vary widely depending on the organization, its business model, and its IT infrastructure. The identification and management of critical CIs are essential to maintaining the availability and performance of IT services, mitigating risks, and ensuring effective service management.

3.2 Using a Configuration Management Database

To explain the use and importance of a configuration management database (CMDB) let's imagine a situation where you're managing the IT operations for a large organization with numerous servers, applications, and databases. In the event of a problem, such as a critical business application performing poorly, it's pivotal to understand the various components involved and their interrelations. This is where a configuration management database becomes essential. A CMDB holds detailed information about all configuration items (CIs) in your IT environment, their configurations, and the relationships between them. With this, your team can promptly identify potential sources of the issue and target their troubleshooting efforts effectively. Furthermore, a CMDB can guide strategic decision-making by revealing the dependencies of various services on certain components. This is crucial when planning updates or changes, helping to minimize potential disruptions. Hence, a CMDB goes beyond a simple inventory of assets. By providing a clear picture of the relationships and dependencies among assets, it enables efficient management of the IT environment, supporting service delivery and risk management. It is a necessary tool that supports both IT asset and configuration management processes.

In essence, a CMDB serves as a detailed inventory of an organization's IT environment. It tracks and stores data on all CIs, which can range from entire systems, like a mainframe computer, to minor hardware components, like modems. This includes information about the hardware and software components (assets and services) in an IT infrastructure and details about their configurations and interdependencies. The scale and complexity of a CMDB can range widely depending on the organization's size - it may be a small tool to capture such information about CIs or a complex asset management system providing a holistic perspective of all IT assets and their interrelations globally. The stored data can be presented in several forms, like a network diagram illustrating the entire infrastructure accompanied by textual descriptions of each CI.

Using a CMDB in ITSM asset management bring several benefits, as outlined in the list below.

- **Change impact analysis:** Before implementing a change, the CMDB can be used for analysis of its potential impact on services and components and can assist in decision-making about the proposed changes.
- **Improved incident and problem management:** A CMDB helps to understand the relationships between various assets and configurations, thereby speeding up the resolution of incidents and problems by identifying which components are impacted
- **Enhanced risk management:** A CMDB allows IT teams to gain a better understanding of the dependencies within their IT environment, helping to identify potential vulnerabilities and risks.
- **Asset lifecycle management:** By providing comprehensive information about all the assets, a CMDB assists in managing the entire lifecycle of assets, from acquisition through to maintenance and disposal.

- **Support for auditing and compliance:** A CMDB helps with the maintenance of an accurate record of assets, their configurations and their relationships, which can be used to demonstrate compliance during audits.

Using a CMDB, therefore, significantly enhances an organization's ability to manage its IT environment effectively, mitigate risks, and deliver high-quality IT services.

The fundamental purpose of configuration management is to enable better IT decision-making, and the use of the data accumulated in the CMDB is the best way to enhance those decisions. The primary use of this data involves preparing reports produced from configuration management information. Examples of uses for configuration management data include:

- Single points of failure
- IT charge back
- Equipment refresh
- Software licensing

The reporting process starts by creating a set of standard reports. These are defined as reports that are run periodically with minimal alterations. There are two categories of routine reports for configuration management. The first focuses on CIs and the relationships between them. They generally present a snapshot of a segment of the environment at a specific moment in time. The second category includes reports that deliver measurement data related to the configuration management service and process. It predominantly revolves around events and procedures that modify or use the configuration data in one way or another.

The first reports that come to mind are what can be called “allocation reports”. This category of reports breaks down CIs according to different segments of the business. From a technical perspective, this may include a report that organizes all CIs according to the responsible party, thus illustrating who is accountable for each element of configuration.

Table 2: Example of an Allocation Report

Domain	Service	Owner	Configuration item	Attributes		
				Name	Serial number	Provider
Communication support	Email	John Smith	SMTP server	SMTPServer01	SN1234567890	Cisco
				IMAP server	IMAP-Server01	SN0987654321
		
	Elec-tronic messages

Network-ing

Source: Rafał Włodarski (2023).

From a business standpoint, a typical standard report lists CIs according to financial responsibility. It allows each department or business owner to obtain a listing of the items they are paying for. This report can be used to create line-of-business or division-level reports. With the appropriate data, it might be beneficial to generate business benefit reports, which illustrate which business units each IT component serves, irrespective of who funds that component.

3.3 Asset Lifecycle

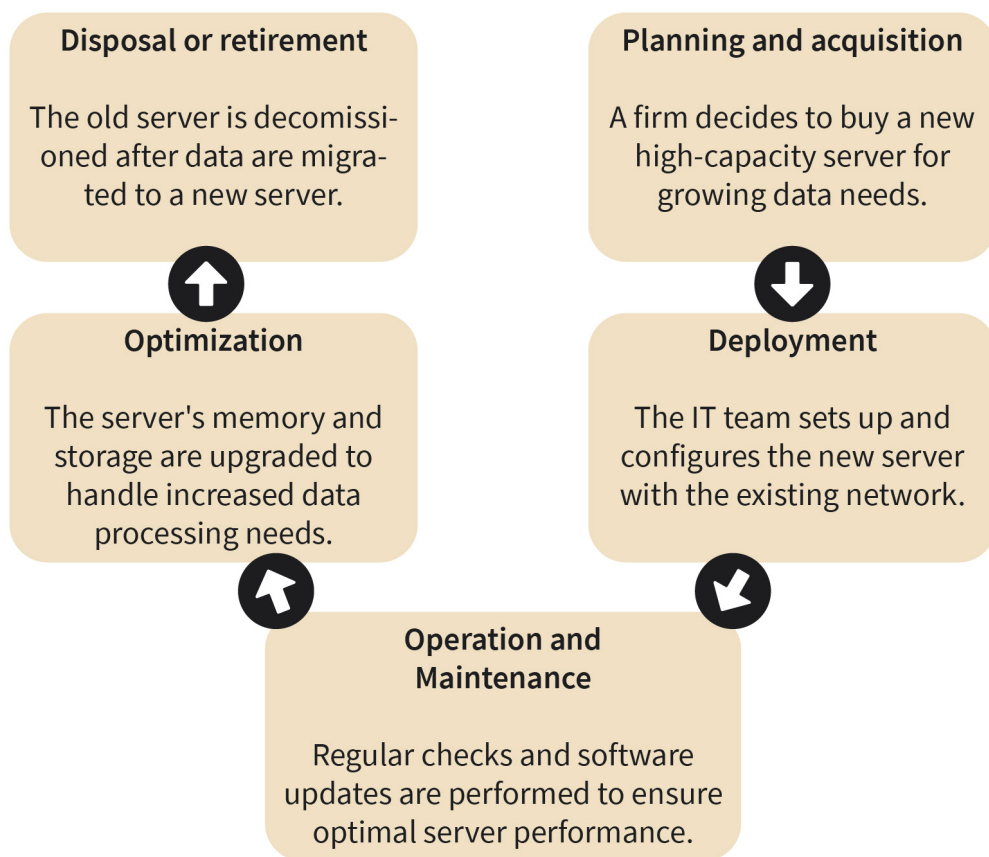
The asset lifecycle, in the context of IT service management (ITSM), refers to the various stages through which an IT asset passes from the time it's conceived to the time it's retired or disposed of. Understanding and managing this lifecycle is crucial for organizations to get the most value from their IT assets while minimizing costs and risks. The asset lifecycle typically consists of several key stages:

1. **Planning and acquisition:** In this phase, the organization identifies the need for a new IT asset, such as a server, a software application, or a piece of hardware. This phase involves assessing requirements, exploring different options, choosing the most suitable solution, and finally purchasing or leasing the asset.
2. **Deployment:** After acquisition, the next step is to install, configure, and integrate the new asset into the existing IT infrastructure. This might include setting up hardware, installing software, and ensuring the asset works properly with other elements of the infrastructure.

3. **Operation and maintenance:** This is the longest phase of the asset lifecycle and involves the regular use of the asset for its intended purpose. It includes all the activities necessary to keep the asset functioning optimally, such as patching, updating, troubleshooting, and regular maintenance.
4. **Optimization:** Over time, assets might need to be optimized to continue meeting the organization's needs effectively. This could involve upgrading software, expanding hardware capacity, or reconfiguring settings. An effective asset management strategy helps to identify when such optimization is necessary.
5. **Disposal or retirement:** Eventually, an asset will reach the end of its useful life and will need to be retired. This could be because it's no longer needed, it's outdated, or it's too expensive to maintain. The disposal or retirement phase involves safely decommissioning the asset, ensuring any sensitive data is securely erased, and recycling or disposing of the asset in an environmentally responsible manner.

These stages aren't the only possible examples of the phases of an IT asset lifecycle, as different organizations might have variations or additional steps based on their specific needs and operational frameworks.

Figure 5: Example of Asset Lifecycle



Source: Rafał Włodarski (2023).

The above example demonstrates the systematic progression of an IT asset through its lifecycle, each stage marked by distinct objectives and activities. This orderly flow ensures that the asset continually aligns with the evolving needs of the organization while maintaining operational efficiency and compliance. Each stage is interconnected, and proficient management across these stages contributes to maximizing the value of assets while minimizing associated risks and costs. Engaging in a structured asset lifecycle management approach enables organizations to drive better resource allocation, informed decision-making, and ultimately, achieve a higher return on investment on their IT assets.

Importance of Security and Compliance in IT Asset Lifecycle

Whether it's a software application or a physical device like a server, every IT asset has access to or contains sensitive information. The aim of IT asset security is ensuring that this information remains confidential and unauthorized personnel don't have access to it, throughout the entire asset lifecycle. This is not just for the sake of business confidentiality but also to protect things like customer information and trade secrets.

A CMDB can also play a crucial role in compliance. The detailed inventory contained in a CMDB ensures that assets adhere to regulatory standards. For instance, if a certain data protection regulation requires particular encryption standards or data residency requirements, the CMDB can track which assets comply with these standards.

3.4 Asset Analysis and Risk Management

In the realm of ITSM, asset analysis is about understanding the details of an organization's IT assets. This process provides organizations with an in-depth examination, weighing the utility and cost-efficiency of assets against the overarching organizational objectives. Asset analysis can provide value on multiple levels, for example, to assess the operational functionality of assets but also to take a deep dive into the economic implications of each asset. We can look at IT asset analysis in terms of the following dimensions:

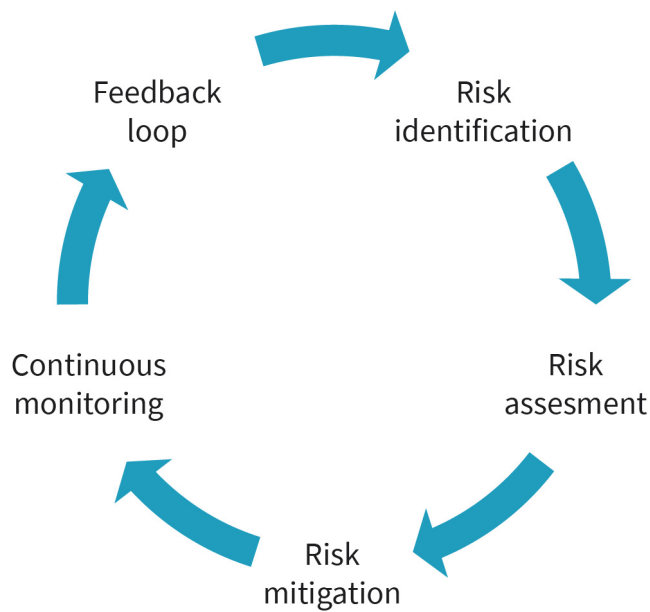
- Scope
 - Covers tangible assets like hardware
 - Includes intangible entities such as software licenses and cloud resources
 - Involves understanding the volume and reach of assets and their roles within the organization
- Functionality
 - Assesses operational functionality of assets
 - Provides insights into performance levels
 - Identifies potential bottlenecks
 - Reveals opportunities for system optimization
- Economic viability
 - Evaluates beyond just functional performance
 - Scrutinizes economic implications of assets
 - Considers costs from initial acquisition to ongoing maintenance
 - Aims to ensure assets bolster the financial health of the organization

- Lifespan estimation
 - Involves the projection of asset lifecycles
 - Guides decisions on timely upgrades
 - Informs maintenance scheduling
 - Helps to determine the right time for asset decommissioning to optimize ROI

Understanding each of these dimensions leads to informed decision-making regarding IT assets, which in turn, contributes to better IT asset management and alignment with organizational objectives.

The analysis is the basis for risk management, a discipline that is paramount to the health and sustainability of any IT strategy. By identifying potential threats, vulnerabilities, or other challenges that could negatively affect the value or functionality of IT assets, organizations can be better prepared. This might mean recognizing technological obsolescence before it becomes critical, pinpointing security vulnerabilities in the early stages, or identifying potential challenges when trying to integrate new assets with older systems.

Figure 6: Phases of Risk Management



Source: Rafał Włodarski (2023).

Following the identification of these risks, a deeper evaluation must be undertaken. Each risk is scrutinized, determining both the potential severity of its impact and the likelihood that it will occur. Such assessments are invaluable as they allow organizations to differentiate between immediate, high-priority risks and those that might be considered more long-term concerns or less dangerous. It allows organizations to take adequate measures.

Recognition and assessment are just the start. The core of risk management lies in the proactive steps taken to mitigate these risks. This can encompass a broad spectrum of actions, for example technological interventions like software patches, process changes, and even intensive employee training sessions. The key is to tailor the mitigation strategy to the specific nature and context of the risk.

There are several approaches to risk management, for example ITIL 4 emphasizes a general collaborative approach. The approach to risk management is more holistic and is integrated across various management practices. On the level of specific roles, the risk manager is responsible for the day-to-day running of risk management. This involves crafting a risk management policy that outlines the organization's approach to handling risks, maintaining the risk register, updating key stakeholders of risk remediation activity, and supporting other processes with risk mitigation activity. The risk assessment team works on identifying, assessing, and analyzing the risks that the organization may face and assists the risk manager in their duties. The risk owner is responsible for managing a particular risk faced by an organization. This person is usually tasked with ensuring that the risk is properly identified, assessed, and managed in a way that aligns with the organization's risk management framework and overall business objectives. The risk owner is responsible for developing and implementing risk mitigation strategies, monitoring the status of the risk, and reporting to relevant stakeholders on the risk and the effectiveness of the mitigation strategies in place.

Given the ever-evolving landscape of technology and associated risks, continuous monitoring is a necessity. By keeping a vigilant eye on both the IT assets and the broader environment in which they operate, organizations can spot and address emerging risks. This ongoing process also ensures that lessons from past risk management experiences (both the successes and the challenges) are integrated into future strategies, fostering an environment of continuous improvement.

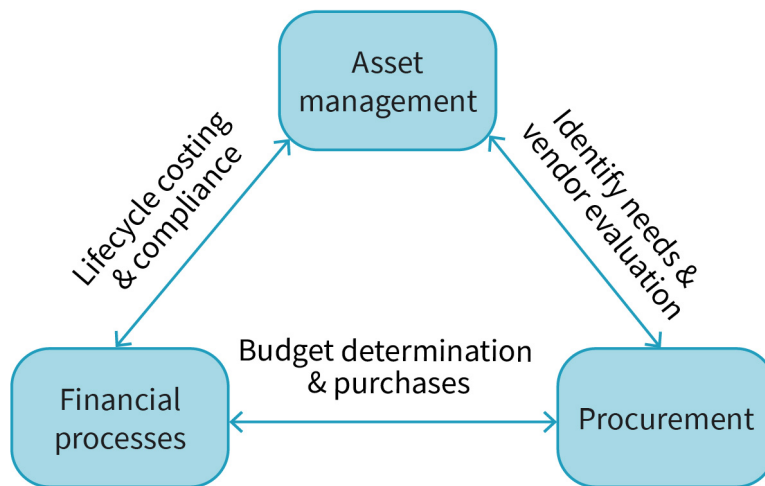
3.5 Interrelation with Procurement and Financial Processes

Asset management doesn't stand alone in the broader domain of ITSM. It is interrelated with procurement and financial processes, in the best case, creating an environment that ensures a seamless flow of operations and maximized value derived from IT assets. This synergy helps to minimize costs and risks associated with IT operations.

The process often starts with ITSM asset management identifying the needs of an organization. Whether due to growth or a shift in business objectives, there is a need for new assets, upgrades, or replacements. Once these needs are identified, those responsible for the financial processes determine the budget for the impending requirements. Here, the role of asset management is pivotal as it provides essential input about estimated costs associated with asset acquisition, deployment, operation, and eventual retirement or disposal.

Moving ahead in this cycle, the procurement processes come into action. The emphasis here is on selecting the right vendors, negotiating favorable contracts, and finalizing purchases. Those involved in asset management remain key stakeholders, ensuring that procured assets align with the organization's needs. Additionally, asset management processes are used to evaluate of vendor reliability and consider various other critical factors.

Figure 7: ITSM Asset Management Interaction with Procurement and Finance



Source: Rafal Włodarski (2023).

The financial implications of assets aren't limited only to their procurement. There is a broader picture, which is the entire lifecycle cost of the asset. Asset management goes beyond the immediate expenses, bringing into focus the operational expenditures, routine maintenance, recurring licensing fees, and sometimes even the costs of potential downtime or disruptions. Such details and nuanced insights become cornerstones for financial planning, enabling the organization to budget and forecast with greater precision.

The importance of tracking cannot be understated, particularly as more assets are procured. Knowing the whereabouts and operational status of assets is essential. This tracking aids key asset management processes but is also crucial for financial reconciliations, especially when considering aspects like asset depreciation, taxation, or even insurance.

When the asset lifecycle comes to an end, a specific set of challenges and responsibilities emerges. Decisions concerning disposal must be made, whether through sale, recycling, or decommissioning. The role of asset management is to aid and guide the financial processes, aiming to reconcile the book value of the asset and extract any residual value from it.

Another layer of complexity is added by the continuous need for audit and compliance. There are constantly changing regulatory frameworks and standards that must be adhered to. Both asset management processes and financial processes jointly ensure that the organization remains compliant, whether it's in terms of software licensing, data disposal methods, or demonstrating the efficient and ethical use of assets.

In essence, the relationship between ITSM asset management, procurement, and financial processes is not just about coexistence but about collaboration. They intertwine and interact at multiple levels, ensuring that an organization achieves the maximum benefits from its IT investments while staying financially prudent and compliant.

Asset Management Systems and Asset Traceability

An asset management system is a centralized system that tracks and manages an organization's assets, particularly IT assets. The following key factors should be considered when it comes to selecting an asset management system:

- **Integration with other systems:** Asset management systems can be integrated with vendor and contract management systems. Sometimes, the functionality of vendor and contract management systems can be expanded to include asset management.
- **Interoperation:** Asset management systems may need to be interoperable with other corporate systems, such as fixed assets systems. This interoperability ensures seamless management and tracking of assets across different departments or functions.
- **Digital invoicing:** For efficiency and accuracy, when assets like IT equipment are procured, their invoices should be digitized. These digital invoices should detail the purchased products specifying their type, model, and serial number.
- **Database population:** Digital invoices can directly populate an asset management database. This automation minimizes manual data entry, ensuring data accuracy and saving time.

An ID traceability matrix can be useful for understanding the challenges that come with asset management. The table below shows a representation of an asset lifecycle in terms of a series of key identifiers.

Table 3: IT Asset Traceability Matrix

	Demand or continuous improvement ID	Project or other funding ID	Requisition ID	Request for Quote ID	Contract ID	Purchase order ID
Record of demand or improvement process	X	X	X			
Request for requisition or acquisition service		X	X			
Quote request			X	X		

Purchase order			X	X	X	X
Advanced shipping notice						X
Invoice						X
Bill of landing						X
Bill of parcel						X
Installation service request or work order		X	X			
Fixed asset record						X
IT asset record						
Actual asset						
CMDB record						

Source: Rafał Włodarski (2023), based on Betz, C. T. (2011).

Table 4: IT Asset Traceability Matrix

	Advance shipping Notice ID	Invoice ID	Stated Shipping Serial ID	Shipment ID	Fixed Asset ID	Installation Service Request ID
Record of demand or improvement process						X
Request for requisition or acquisition service						
Quote request						
Purchase order						
Advanced shipping notice	X					
Invoice		X	X			
Bill of landing				X		
Bill of parcel			X			
Installation service request or work order				X		X
Fixed asset record		X	X		X	
IT asset record					X	

Actual asset						
CMDB record						

Source: Rafał Włodarski (2023), based on Betz, C. T. (2011).

Table 5: IT Asset Traceability Matrix

	IT Asset ID	Configuration Item ID	Plate Serial #	BIOS Serial #	Asset Location	Assigned Host Name	Assigned DNS Name(s)
Record of demand or improvement process							
Request for requisition or acquisition service							
Quote request							
Purchase order							
Advanced shipping notice							
Invoice							
Bill of landing							
Bill of parcel							
Installation service request or work order					X		
Fixed asset record					X	X	X
IT asset record	X		X	X	X	X	X
Actual asset	X		X	X	X	X	X
CMDB record	X	X	X	X	X	X	X

Source: Rafał Włodarski (2023), based on Betz, C. T. (2011).

The IT asset traceability matrix is a tracking tool that monitors assets throughout their lifecycle. It maps each IT asset and provides a comprehensive snapshot of an asset's status. The matrix fosters accountability by offering a detailed audit trail and aids the identification of inconsistencies. Furthermore, by showcasing the details of each asset, it supports informed decision-making about upgrades, replacements, or decommissioning. Automation helps maintain the matrix in real-time, reducing errors. It is not only instrumental to operations: The matrix also ensures compliance with regulatory standards, making it crucial for both management and transparency.



SUMMARY

The ITSM asset lifecycle consists of the various stages an IT asset undergoes from conception to retirement. These stages include planning and acquisition, deployment, operation and maintenance, optimization, and disposal or retirement. Effective management of this lifecycle can ensure maximum value extraction from IT assets while also minimizing costs and risks. Tools such as the configuration management database (CMDB) are essential in tracking these assets. At the same time, IT asset security has to be ensured throughout the lifecycle (especially when disposing of assets) to make sure that sensitive data is not compromised.

Asset analysis and risk management are interlinked. Asset analysis helps organizations understand the operational and economic implications of their IT assets. Risk management involves identifying and addressing potential threats to these assets, including recognizing, evaluating, mitigating, and continuously monitoring risks to ensure that IT assets are protected and contribute value in the long run.

The main goal of asset management in ITSM is to ensure that organizations maximize the value derived from their IT assets throughout their lifecycle. Asset management is closely intertwined with the organization's financial and procurement processes. This interconnection is crucial for identifying needs, setting budgets, procuring assets, and tracking them throughout their lifecycle. The interaction ensures that the organization utilizes its IT investments optimally while staying financially sound and compliant.

UNIT 4

IT SERVICE MANAGEMENT: SUPPLIER MANAGEMENT

STUDY GOALS

On completion of this unit, you will be able to ...

- understand the pivotal role supplier management plays within the wider scope of IT service management.
- align supplier offerings with company goals and derive real value from contracts.
- identify and tackle risks that come with working with external entities and learn how to minimize potential setbacks.
- recognize and describe general sourcing approaches in IT service management (ITSM).
- identify the essential components that make up a robust service level agreement (SLA).

4. IT SERVICE MANAGEMENT: SUPPLIER MANAGEMENT

Introduction

In today's technology-driven world, the smooth operation of businesses often hinges on IT infrastructure, making ITSM a cornerstone of success. Supplier management plays a pivotal role within ITSM, ensuring that providers of essential IT services consistently meet the necessary standards in terms of quality and cost-effectiveness.

Take the example of an e-commerce giant like Amazon. Its operations, spanning numerous countries and millions of products, depend on a complex IT network. From servers to cybersecurity to logistics software, they rely on various suppliers. A hiccup from even one supplier, like a server issue during peak sale days, can lead to major financial and reputational repercussions. This highlights the importance of effective supplier management. Through regular monitoring, performance reviews, and proactive communication, ITSM professionals can make certain that every supplier is in sync with the company's overarching objectives in order to deliver optimal results.

4.1 Overview

ITSM supplier management is about more than simply liaising with external vendors. It's an intricate dance that involves aligning the offerings of these suppliers with the specific needs and ambitions of the organization. Supplier management revolves around ensuring that contracts with vendors do not just exist on paper but come alive in practice, aligning seamlessly with the business objectives.

At its heart, this process is about value. Every cent spent on an external service should return tangible benefits to the organization. But value isn't the only goal. It's equally crucial to monitor how these suppliers perform. Are they living up to their promises? Are the service levels agreed upon being met consistently? Beyond monitoring, there's the dimension of risk. Each external engagement carries with it a certain level of dependency. Supplier management seeks to identify these risks, weighing the benefits against potential pitfalls.

The process covers a broad spectrum of day-to-day activities: from strategizing about which types of suppliers align with organizational needs, to the meticulous task of selecting them based on stringent criteria. Even once the suppliers are on board, the work doesn't end. Contracts need continuous oversight, ensuring terms are adhered to while also being flexible enough to evolve with changing needs. Regular reviews punctuate this relationship, measuring the supplier's performance against set benchmarks. Sometimes, tough decisions have to be made—whether to renew a relationship or bid farewell.

Within the intricate framework of ITSM, supplier management is a pivotal element that focuses on the broader strategy. It is instrumental in ensuring that external collaborations effectively bolster the service delivery mechanism, thereby elevating the organization's pursuit of service excellence and operational efficiency. At its core, supplier management is dedicated to forging and maintaining robust, mutually beneficial partnerships with suppliers to guarantee consistent and superior delivery of IT services.

Today, many IT services involve more than a single IT service provider. For instance, services dependent on networks might rely on a telecommunications provider to connect distant sites. Hardware maintenance is typically managed by an external party, and commercial software is often maintained by external suppliers, which might include the original software vendor. In the context of service level management, the dynamic between these external entities and the primary IT service provider is shaped by foundational contracts. These contracts are established between the IT service provider and the third-party entities, ensuring service support. The goal behind supplier management is to guarantee the consistent quality and cost-effectiveness of IT services. It is much more than merely brokering an initial support agreement.

The main objectives of supplier management are:

- Creating and maintaining a supplier strategy
- Creating and cultivating positive relationships with suppliers
- Negotiations of contracts with suppliers that match business requirements and overseeing these contracts throughout their duration
- Collaborating with service level management to ensure that contracts supporting other services meet the standards set in service level requirements and service level agreements
- Monitoring supplier output to guarantee cost-effective delivery
- Establishing and sustaining a comprehensive supplier and contract management database

Supplier management consists of the following sub-processes (Thejandra B. S., 2014):

- Providing the supplier management framework
- Evaluation of new suppliers and contracts
- Establishing new suppliers and contracts
- Processing of standard orders
- Supplier and contract review
- Contract renewal or termination

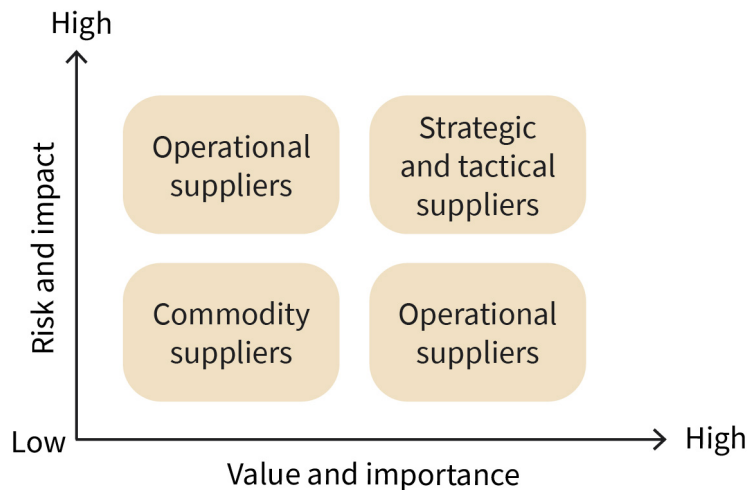
Supplier management involves overseeing suppliers and the services they deliver to ensure maximum value is derived from each supplier throughout the duration of the partnership. Considering the complexity of modern IT services, it's common for a single service to be furnished by a blend of both internal and external providers. The challenge of supplier management lies in coordinating these multifaceted relationships with external suppliers, ensuring alignment in objectives, and facilitating service delivery that meets the

standards set in SLAs without compromising on cost-effectiveness. The ultimate goal is not putting pressure on suppliers to give the lowest possible price but to nurture a relationship that yields sustained, long-term value for the organization.

Let's look at an example that illustrates the importance of good supplier management. Imagine a renowned healthcare institution that has pushed their primary equipment vendor on costs so aggressively that signs of potential service disruptions become evident. The institution prides itself on its assertive negotiation skills, overlooking the looming risks. The situation escalates to legal proceedings, with each party pointing fingers at the other. Regardless of the resolution of the lawsuit, the equipment services fail, leaving the institution scrambling to address the aftermath.

To get the most out of working with a supplier, the IT team should clearly know where they're headed and what the plan for future is. This means understanding their long-term plans in light of the business they support and figuring out the services they need to create to reach those goals. By comparing what they can do with what they need to achieve, they'll see where they might need an extra pair of hands from outside. This self-check, along with considering any risks, sets the stage for deciding how to work with suppliers. It makes sense to spend more time with key suppliers and less with the ones that aren't as critical. This means we should sort suppliers into categories. A useful way to do this might be looking at how much they impact services, the risks they might bring, and how valuable they are. For example we might group suppliers according to the following categories: strategic, tactical, operational, commodity (as shown in the image below).

Figure 8: Supplier Categorization



Source: Rafał Włodarski (2023), based on Sansbury, J., Brewster, E., Lawes, A., Griffiths, R. (2016).

It's essential to note that supplier categorization isn't a one-time task. Regular reviews are necessary as services, organizational priorities, and suppliers themselves can change over time. A supplier considered tactical today might become strategic tomorrow based on evolving needs.

Let's imagine CloudTech, an IT firm that initially categorized ServerWare, a server hardware provider, as a tactical supplier. As the company shifted focus to cloud-based services and started building its data centers, ServerWare's hardware became crucial to their core operations. During a routine supplier review, CloudTech recognized the increased importance of ServerWare's services. Consequently, they reclassified ServerWare from a tactical to a strategic supplier, leading to the benefits listed below.

- Increased collaboration: initiating strategic planning sessions with ServerWare for future hardware needs
- Improved SLAs: renegotiating service agreements for better support and faster response times
- Risk assessment: strengthening risk management strategies to ensure ServerWare's supply chain reliability

By updating ServerWare's categorization, CloudTech ensured their supplier management remained responsive to changing business needs and service dependencies.

Supplier management acts as the essential backbone of ITSM, seamlessly integrating external services into an organization's framework for optimal performance. To fully leverage this, organizations must have a clear direction and choose suppliers that align with their overarching vision. Regularly reviewing and classifying suppliers based on their importance, potential risks, and overall value is vital. Ultimately, supplier management goes beyond mere contracts and costs – it's about establishing strategic partnerships that drive organizational progress.

4.2 General Sourcing Approaches in ITSM

Sourcing is an important aspect of ITSM as it determines how an organization procures and manages its IT services. Sourcing decisions can heavily influence the quality, cost, and delivery of IT services. There are several general sourcing approaches, each with their own advantages, challenges, and strategic implications.

In-house or insourcing

The terms “in-house” or “insourcing” refer to an organization delivering IT services using its own internal resources. This approach offers full control over resources. Organizations have a direct influence over priorities, quality, timelines, and security. Moreover, sensitive information remains in-house, potentially reducing data leakage or security risks. There's a stronger alignment with organizational culture and objectives since everyone is part of the same entity. However, it can require significant investment in infrastructure, technology, and personnel. There is also risk of limited scalability, slower technological adoption, and potential difficulties in attracting top-tier talent if the organization is not primarily tech-focused.

Outsourcing

Outsourcing involves an organization contracting an external provider to deliver certain IT services or functions. This can lead to potential cost savings, especially if the provider has economies of scale or specialized expertise that the organization lacks. It can also provide access to specialized expertise and broader talent pool and allow the organization to focus on core business functions. On the downside, there's a potential loss of control over service quality and risks related to vendor management. Communication barriers can also arise, and there might be a misalignment between the vendor's goals and the organization's objectives.

Multi-sourcing

When an organization uses multiple suppliers to deliver its IT services, it is commonly known as “multi-sourcing”. This approach reduces dependency on any single vendor and can optimize costs and quality by selecting the most suitable providers for different service areas. It reduces dependency on a single provider. However, it also brings challenges like complex vendor management, potential integration issues, and a need for robust coordination.

Partnership or joint venture

Partnership or joint venture sourcing involves two or more organizations collaborating to deliver IT services. This leverages the strengths of each party, leading to shared risks and investments and access to complementary skills. It can also foster innovation and provide access to markets or expertise that one party lacks. However, it requires a strong alignment of objectives and can introduce complexities in governance. Potential disputes or differences in organizational culture may occur, leading to friction.

Cloud sourcing

The procurement of IT services from cloud providers is known as “cloud sourcing”. These services can range from infrastructure to software. This approach is scalable, can be cost-effective, and provides access to advanced technologies without a significant upfront investment. Concerns can arise around data security, a reliance on external networks, and the potential for vendor lock-in when migrating to another service becomes challenging or costly.

Shared services or collaborative approach

The shared services or collaborative approach involves multiple parts of an organization (or even multiple organizations) sharing IT services through a centralized unit. This leads to cost savings through economies of scale and standardized processes. However, there's a potential for reduced flexibility in meeting specific needs and it can introduce governance complexities when trying to satisfy diverse stakeholders.

When choosing a sourcing approach, organizations should weigh up factors like cost, control, flexibility, risk, and strategic alignment. It's worth noting that the best sourcing model might evolve over time, influenced by changing business needs, technological advancements, and market shifts. A sourcing approach is not locked in forever – it can be changed.

However, the transitioning between sourcing approaches in ITSM can be complex and requires meticulous planning and execution. This could be due to cost inefficiencies, a change in organizational goals, the need for more flexibility, or various other reasons.

Let's look at this in practical terms by exploring an example. Imagine a company, TechFlow Inc., that initially had an insourced ITSM model but has decided to transition to an outsourced model due to cost inefficiencies and the need for advanced technological expertise. So that the company is prepared for the change, it's key that they follow three main steps, as outlined in the following paragraphs.

Step one: assessment

Before considering a transition, it's vital to understand why the current sourcing strategy is no longer optimal. The decision to reassess the IT sourcing approach typically lies with senior IT management, often in conjunction with procurement specialists and key stakeholders from the relevant business units. The assessment is grounded in various data points that help to illustrate the full picture of the current sourcing strategy's performance and its alignment with the organization's goals.

Let's look at what this step entails in terms of our TechFlow scenario: Senior IT management at TechFlow, alongside procurement specialists, have assessed their current ITSM model. They have realized that the in-house team is struggling to keep up with the latest IT trends and the cost of maintaining the team is high. They have collected data on the performance of their IT services, employee productivity, and costs involved.

Step two: setting clear objectives

Once the need is identified, clear objectives must be for the transition. This could range from achieving cost savings, gaining access to better technology or expertise, or improving service quality. In our TechFlow scenario, the primary goal is to reduce costs and gain access to cutting-edge technology and expertise, thereby improving service quality and efficiency.

Step three: transition roadmap

A comprehensive roadmap should be created, detailing each phase of the transition. This includes milestones, expected challenges, resources needed, and timelines. In our TechFlow example, the company will need to develop a roadmap outlining the transition to an outsourced model, detailing milestones like vendor selection, contract negotiations, staff retraining, and timelines for each phase.

These steps can help to prepare for transition. However, during the transition period, the company could still face many obstacles and challenges. Some key examples are outlined in the list below.

- **Operational disruption:** Any change in sourcing can disrupt regular operations. This is particularly true when transitioning from insourcing to outsourcing or vice versa. In the TechFlow example, regular operations could be disrupted during the initial stages of transition, as employees adjust to working with the new external team.
- **Data migration & security:** If the transition involves changing IT service providers, there's a risk associated with migrating sensitive data. Ensuring data security during this phase is paramount. For example, the transition planned at TechFlow will involve moving sensitive data to the vendor's systems, necessitating stringent data security measures.
- **Cultural & organizational challenges:** Changing sourcing approaches can affect organizational culture, especially if it involves insourcing a previously outsourced function or vice versa. Employees may feel threatened or unsure of their roles, leading to resistance. In the TechFlow example, the shift might also require adjustments to the organizational culture to align with the outsourced model.
- **Contractual impediments:** Existing contracts with service providers might have clauses that make transitioning challenging or costly. In the case of TechFlow, existing contracts with in-house staff and vendors will have to be revisited, leading to legal complexities and negotiations.

Best practices can facilitate the change and make the transition period smoother. Examples of these practices, broken down in to six key areas, are outlined in the list below.

1. **Stakeholder communication:** Clear and regular communication of the reasons for the transition, its benefits, and its progress to all stakeholders, including employees, management, and external partners can help in managing expectations and reducing resistance.
2. **Pilot testing:** Before a full-scale transition, organizations should consider running pilot tests to identify potential issues. For example, when transitioning to a cloud sourcing model, it might be advisable to start with one department or function, assess the results, and then roll it out more broadly. In the TechFlow example, it could make sense to initially transition the IT support for a single department to the outsourced model. The success of this pilot could then inform the broader rollout strategy.
3. **Dedicated transition team:** A team should be assigned to oversee the transition. This team should have representatives from IT, operations, HR, and finance, ensuring a holistic approach to the transition.
4. **Continuous monitoring & feedback loop:** Implementation of mechanisms to continuously monitor the transition's progress is crucial. Regular feedback loops can help to quickly identify and address challenges as they arise.
5. **Knowledge transfer & training:** If the transition involves new tools, technologies, or processes, it may be advisable to ensure that there's a structured knowledge transfer process in place. This is pivotal when transitioning from outsourcing to insourcing, where the in-house team might need to understand systems or processes previously

managed by an external provider. In terms of the TechFlow example, it could be helpful to organize training sessions for the in-house team to familiarize them with new processes and tools introduced by the vendor.

6. **Exit strategy:** Especially when transitioning away from a particular vendor or sourcing model, it is vital to have a clear exit strategy in place. This involves understanding contract terminations, data retrieval, and ensuring there is no remaining dependency on the previous model or provider.

In ITSM, choosing the right sourcing strategy is not a one-size-fits-all solution. Organizations should evaluate their unique requirements, financial capabilities, appetite for risk, and strategic objectives to select the best approach. As the business landscape and technology shift, it might be necessary to switch to a different sourcing strategy. While this can be a complex undertaking, with thorough planning, an understanding of potential challenges, and the use of best practices, organizations can navigate these transitions effectively.

4.3 Evaluating and Selecting Suppliers

Evaluating and selecting suppliers is a vital process that determines how effectively an organization can meet its IT service requirements. The first step is understanding the requirements. This may entail comprehensively determining the IT needs, specifying features, and setting performance expectations. Service level requirements should also be defined, as laying out the metrics and performance indicators that suppliers should meet is crucial for effectiveness. This is followed by pre-qualification of suppliers, where potential suppliers are assessed based on preliminary criteria. It's essential to ensure they are financially stable and have a good market reputation. This initial assessment provides a picture of their past performance, client feedback, and any industry recognitions. Once the preliminary assessments are done, organizations typically send out a **request for information (RFI)** to gather basic data about suppliers' capabilities, which is followed by a **request for proposal (RFP)**. This RFP lays out the organization's requirements in detail, and suppliers respond with how they plan to meet these needs and the associated costs.

Evaluation is a crucial stage where the proposals are studied carefully. This includes a technical assessment to ensure the supplier has the required tools, technology, and skilled workforce. Costs are scrutinized, ensuring they align with the budget and offer adequate value. References and case studies from the supplier's past projects also offer valuable insights into their capability and reliability.

After a supplier is chosen, contract negotiations can start. The contract should clearly define the service scope, costs, and timelines. It's also crucial to set out the service level agreements and the penalties or rewards associated with them. Once the contract is ready, the onboarding process begins. Transfer of knowledge is a significant part of this phase. The supplier's systems and processes are integrated with the organization, and tools for communication and collaboration are set up. Lastly, even after the supplier starts

Request for information (RFI)

A preliminary document used by organizations to gather general information about products, services, or suppliers before making a purchasing decision.

Request for proposal (RFP)

A formal document that outlines specific requirements and invites suppliers to submit proposals offering solutions and pricing for products or services.

their service delivery, the service must be continuously monitored and reviewed. This ensures that they meet the standards agreed upon and offers an opportunity to identify and implement areas of improvement.

It's important to remember that the ITSM environment is constantly evolving. Supplier contracts and relationships should be flexible, so that they can be adapted to changing needs and requirements. The way organizations interact with their suppliers can significantly influence their competitive positioning, operational efficiency, and innovation potential. Supplier relationship models provide a structured approach to categorization, management, and optimization of supplier partnerships. The purpose of these models is to ensure that businesses can strategically align their expectations, investments, and engagement strategies with the right suppliers, leading to mutually beneficial outcomes. Nine supplier relationship models (including classifications in terms of performance and strategy) are outlined in the list below.

1. **Harvest:** *High performance, low strategic potential.* This model aims at optimizing the benefits from the supplier without binding the client. It focuses on short-term gains, flexible contracts, and minimal obligations for the client. For example, a retail company uses a harvest model for short-term relationships with suppliers of seasonal items like holiday decorations, opting for flexible, season-specific contracts to capitalize on the best prices and terms without long-term commitments.
2. **Influence:** *High performance, medium strategic potential.* This model is designed to harness the supplier's technology to collaboratively create and introduce products and services. It puts emphasis on joint research & development, shared intellectual property, and co-branding opportunities. In terms of a practical example, imagine a smartphone manufacturer that forms a relationship with a camera tech supplier to co-develop an exclusive camera module, sharing expertise and marketing to create a co-branded product that helps them to get an edge over competitors.
3. **Integrate:** *High performance, high strategic potential.* This model promotes engagement in a long-term exclusive partnership aiming to influence the market. It focuses on long-term contracts, high dependency on each other, and shared market strategies. For example, an automotive company might partner with a battery manufacturer in an integrated model, signing a long-term contract to use exclusive batteries and jointly market their electric technology as a new standard.
4. **Improve:** *Medium performance, low strategic potential.* This model supports the supplier in addressing deficiencies, given a favorable business justification. Its main objectives are continuous improvement processes, regular feedback loops, and possibly shared training sessions. For example, a software firm might adopt an improve model with a cloud provider to enhance service performance through infrastructure upgrades and shared training, ensuring better service for the firm's applications.
5. **Sustain:** *Medium performance, medium strategic potential.* The goal of this model is to track supplier performance and enhance value by aligning strategies. It emphasizes performance tracking, regular reviews, and strategic alignment sessions. To illustrate this model, let's imagine a coffee chain employs a sustain model with their bean supplier, holding quarterly reviews to ensure alignment with quality, delivery, and ethical sourcing standards.

6. **Invest:** *Medium performance, high strategic potential.* This model focusses on setting mutual goals to foster expertise and allocate resources from both supplier and client. It means mutual investments in training, technology, or market penetration strategies. For example, let's imagine a biotech firm that uses an invest model with a lab, pooling resources and expertise in shared facilities to collaboratively develop and market new drugs, sharing both costs and profits.
7. **Mitigate:** *Low performance, low strategic potential.* This model adopts a decisive "progress or exit" approach. It focuses on strict performance benchmarks, clear consequences for underperformance, and potential exit strategies. For example, a construction company might implement a mitigate model with a material supplier, setting strict monthly delivery benchmarks and clear penalties for underperformance to ensure timely deliveries.
8. **Develop:** *Low performance, medium strategic potential.* In this model, minimal resources are allocated to prepare the supplier for business engagements with the client. That may be capacity building, mentorship programs, and gradual onboarding processes. To illustrate this model let's imagine a financial corporation that engages in a develop model with a cybersecurity firm, offering mentorship and a phased integration with the aim of building the firm into a key security services provider.
9. **Bail out:** *Low performance, high strategic potential.* This model assigns a swift response team to ensure a steady supply. The core of this model is crisis management, rapid response teams, and contingency planning. For example, an airline might activate a bail-out model with a parts supplier during a fleet crisis, forming a rapid response team to secure immediate parts supply and minimize operational downtime.

The nine supplier relationship models underscore the diverse approaches organizations can adopt to optimize their interactions with suppliers. These models not only offer a structured way to categorize and manage supplier partnerships but also provide a roadmap to ensure aligned expectations. The key to an effective supplier strategy lies in recognizing the nuances of each relationship and adopting the right model to foster growth, collaboration, and adaptability.

Evaluating and selecting the right suppliers is an aspect of supplier management that directly impacts an organization's operational efficiency, competitiveness, and innovation capabilities. As the IT environment evolves, it's essential for businesses to maintain agile and adaptable supplier relationships. The nine supplier relationship models serve as a valuable guide, enabling organizations to strategically navigate their supplier partnerships. By understanding and leveraging these models, organizations can cultivate supplier relationships that not only meet their immediate needs but also position them for future success.

4.4 Contracting and Service Level Agreements

In the world of ITSM, the relationship between organizations and their suppliers has grown increasingly intertwined. To navigate this, clear agreements, such as contracts and service level agreements (SLAs), are becoming essential. In ITSM terms, a contract is a legally binding arrangement that outlines the terms of service delivery, including aspects like pricing. SLAs define the expected standards, deliverables, and performance metrics.

The significance of these tools for supplier management lies in their ability to offer clarity and transparency. They provide a shared understanding of roles, responsibilities, and expectations, which in turn ensures consistency in service delivery. With well-defined standards, organizations can hold suppliers accountable, addressing any deviations from benchmarks both parties have agreed upon. In addition, by delineating conditions and expectations, these agreements act as a protective measure, helping organizations mitigate potential risks associated with outsourcing.

When an organization uses external vendors to support or maintain critical equipment, or outsources certain services, a contract (agreed upon and signed by both parties) is essential. The contract should be prepared in detail and cover the following:

- Scope of work
- Exclusions from the contract
- Hours of service
- Roles and responsibilities
- Timeframe of contract
- Spares support
- Required reporting
- Terms of payment
- Penalties

Drawing up contracts requires collaboration between technical, financial, and legal teams to ensure comprehensive coverage and precise wording. It is crucial for contracts to be robust enough to hold up under legal examination or even in court if the need arises. Apart from that, a thorough technical service level agreement (SLA) is indispensable for maintaining appropriate support. While some businesses believe that outsourcing relieves them of internal oversight, this is a misconception. Continuous monitoring of the outsourced work is vital to ensure it aligns with expectations. Delegation of tasks doesn't absolve an organization of its duty to oversee and ensure smooth progression. If the outsourcing partner performs as expected, the task of overseeing is straightforward. However, any shortfalls on their part can escalate into more significant challenges. It's essential to understand that any issues with vendors inevitably lead to greater challenges for an organization. For example, let's imagine a healthcare company that has outsourced its IT infrastructure management to a vendor. The contract lacks specific clauses on cybersecurity measures, leading to a data breach. The breach not only exposes sensitive patient data but also results in significant legal ramifications and loss of trust among patients.

As the above example demonstrates, it's vital to have written agreements with the vendors, service providers, and consultants who help keep things running for a business. Without a clear agreement in place, there's no way to be sure that the required assistance will be provided by external parties when needed. A service agreement usually covers the points listed below. Each of these needs to be explained in simple and clear terms, so everyone involved knows what is expected.

- **SLA details**

- Name of the project or area of support
- Contract number or reference number, with date
- Start date and end date for contract
- Parties to the agreement, including authorized persons, departments and workplace addresses
- Description of the project or work expected
- Detailed scope of work

- **Obligations and requirements**

- Common obligations of both parties
- Out of scope (both parties)
- Assumptions, constraints, risks, and limitations
- Hardware, software, spares, and other requirements

- **Operational procedures**

- Incident and problem management procedures
- Escalation procedures
- Change management procedures
- Help desk or support procedures, turnaround times for response, resolutions, and workarounds
- Standard working hours or service windows covering the number of hours per day and holidays

- **Legal and financial aspects**

- Legal aspects, jurisdiction, and non-disclosure clauses
- Financial aspects including budgets, payment terms, penalties, additional costs, extra charges, taxes, and billing methods
- Project termination clauses, notice periods for closure
- Signatures of authorized representatives from both parties

Over time, the needs of an organization may evolve, and the services provided by the supplier might require adjustments. Periodic reviews of contracts and SLAs allow for necessary amendments, ensuring that they remain relevant and in alignment with the current business landscape. This process involves checking whether the metrics and deliverables originally agreed upon are still pertinent, and if the supplier's capabilities have grown or changed.

What is also important to remember is that, while the aim is to strictly adhere to SLAs, unforeseen circumstances might lead to deviations. It's crucial for organizations to approach such scenarios with a degree of flexibility. This doesn't mean compromising on

standards, but rather understanding that factors outside of a supplier's control can impact service delivery. In such situations, a collaborative approach to finding solutions will be more effective than a punitive one.

Contracts and SLAs are more than mere documents. They are the backbone of the relationships between organizations and suppliers, ensuring clarity, consistency, and accountability. They are the blueprints that guide the intricate workings of ITSM, from outlining the specifics of services, to safeguarding against potential hiccups in outsourcing.

4.5 Monitoring and Controlling Suppliers

As organizations often heavily rely on various suppliers for essential IT services, it's important to ensure they're consistently meeting the standards that have been agreed upon. A well-rounded approach is needed to effectively track and evaluate supplier performance, ensuring it aligns with the broader business objectives.

Understanding the importance of monitoring and controlling suppliers comes down to a few pivotal factors. Firstly, continuous supplier oversight offers a means to proactively pinpoint and tackle issues before they become bigger challenges. Moreover, it guarantees consistent delivery of the expected services or products, and again at the quality and cost that was agreed upon. This continuous assessment and feedback loop ensures that the service quality remains up to the mark.

At the heart of monitoring and controlling lies a series of operational facets. Firstly, there's the critical task of defining explicit performance metrics. Metrics such as response times, resolution times, and uptime (among other vital KPIs) serve as a benchmark. Then, there is the practice of holding regular reviews to weigh supplier performance against the set SLAs. Such reviews can shed light on performance trends, emerging challenges, and potential improvement zones. Creating channels for feedback, both from the core IT team and end-users, can also offer nuanced insights into supplier performance. Coupled with this is the importance of regular audits, ensuring that suppliers remain compliant with contractual, legal, and industry norms. Lastly, using monitoring insights can spur initiatives geared towards continuous improvement with suppliers. Current ITSM tools are highly useful for this purpose. With their specialized features, they can closely monitor performance metrics, create detailed reports, give warnings if an SLA isn't met, and provide an overview of all suppliers at a glance.

That said, it's not always smooth sailing. Keeping data accurate, working closely with suppliers, and managing multiple suppliers can be challenging. Nevertheless, adopting certain best practices can significantly streamline the process. Clear and open communication is pivotal, ensuring suppliers are always in the know about expectations, metrics, and feedback. A balanced stance, where good performance is praised as much as non-compliance is flagged, can foster a positive relationship. When issues arise, a collaborative problem-solving approach can yield better results than a confrontational one.

These actions are called supplier relationship management (SRM) and are evident, for example, in the framework TrueSRM. TrueSRM is a specific approach to SRM but the concepts and practices of SRM are broadly applicable and evident across various frameworks and methodologies in business and supply chain management. Over time, the position of suppliers can change and an organization should control that. The dynamic nature of the framework can be used by companies to their advantage and is tied to the concept of primary and secondary interaction models. The secondary model indicates the position a supplier could potentially move to based on their performance.

In terms of the operating-model elements for SRM, top-down decision-making determines a supplier's strategic potential, whereas bottom-up input helps to evaluate a supplier's performance. Governance models need to be differentiated based on the type of interaction model they are associated with, and resources should be allocated according to supplier positioning.

Post-implementation, the focus of SRM should shift to gaining a competitive advantage instead of merely measuring benefits. The dynamic nature of suppliers means that their performance can alter, leading them to move within the SRM framework. Organizations are usually cautious with new suppliers and it is advisable for new suppliers with limited strategic potential to be placed in the "Mitigate" category until they prove themselves. Exceptional new ones might be put in the "Develop" category, whereas those that have high potential but fall short on performance might be put in the "Bail Out" mode. Certain moves within the SRM framework, both vertically and horizontally, are deemed feasible while others are not, as illustrated in the following figure.

Figure 9: Supplier Moves in Nine Supplier Relationship Models

		Strategic potential		
		Low	Medium	High
Performance	High	Harvest ↑	Influence ↑	Integrate →
	Medium	Improve ↑	Sustain ↑	Invest ↑
	Low	Mitigate ↙	Develop ↑	Bail out ↖

Source: Rafał Włodarski (2023), based on Easton, S., Hales, M. D., Schuh, C., Strohmer, M. F. (2014).

So far we have emphasized the fact that the framework isn't fixed, which might lead you to ask: Does this mean a supplier can simultaneously fit into multiple interaction models? In essence, the SRM framework's purpose is to represent the supplier as a single unit, without diving into the category level. Instead, the framework acts as a general guideline. The flexible approach of the SRM framework sets the stage for a more dynamic and encouraging interaction. This concept can be likened to how a company might approach employee development. Just as it's important to communicate potential growth and

advancement opportunities to an employee during a review, similarly, it's essential to convey to suppliers the possibility of evolving their status within the SRM framework based on their performance and contributions.

For SRM decision-making, the structure should ideally use existing cross-functional teams, thus avoiding the establishment of new committees. Decisions come in two main types: those made bottom-up for performance evaluations and those made top-down for assessing strategic potential. These cross-functional teams are responsible for bringing both the top-performing and the under-performing suppliers to their leaders' attention. In diversified companies, the ranking of supplier performance should be conducted at the individual business level.

Governance models differ based on the supplier type. Those in the "Harvest," "Improve," and "Sustain" categories will have periodic performance assessments. However, the focus for suppliers in the "Mitigate," "Develop," and "Bail Out" categories will be on rectifying performance or addressing other business-critical issues. Suppliers under the "Influence," "Integrate," and "Invest" tags will undergo a considerable shift in governance. Their treatment will include more proactive communication, the development of roadmaps, account planning, and the establishment of joint steering committees.

For the majority of suppliers, it's vital to pay attention to both the planning and facilitation of meetings. Without adequate care and planning, quarterly meetings can deteriorate into mere rituals, resulting in wasted chances. Bringing key stakeholders from both sides together demands substantial time and financial resources. Typically, relationship managers from both entities should agree on the primary subjects to address over the coming year's series of meetings, all based on the account strategy. Subsequently, functional teams from both parties should be assigned to compile insightful updates and forecasts concerning the advancement in relation to the account strategy. Taking this proactive approach during supplier meetings can greatly enhance the overall partnership.

Effectively monitoring and managing suppliers is crucial to the overall success of ITSM. The TrueSRM framework offers a dynamic approach to supplier relationship management, highlighting the importance of clear performance metrics, regular reviews, and open communication. While managing suppliers poses challenges, adopting best practices like proactive engagement and strategic planning can foster a beneficial relationship. The key lies in a tailored approach, recognizing the dynamic nature of supplier positioning, and maintaining constructive communication. In essence, adept supplier management ensures not only operational efficiency but also a lasting and evolving partnership.

 **SUMMARY**

In the context of ITSM, supplier management helps to align the goals of an organization with the services offered by suppliers. It goes beyond contractual arrangements, fostering beneficial partnerships that enhance operational efficiency and service excellence. The core objective is to derive tangible value from investments in external services,

with a focus on diligent monitoring of supplier performance and assessing associated risks. This includes crafting strategies, nurturing positive relationships, negotiating contracts carefully, and maintaining ongoing oversight to ensure favorable business outcomes.

Sourcing is crucial in determining the procurement and management of IT services, impacting the quality, cost, and delivery of these services. Different approaches to sourcing include insourcing (in-house), outsourcing, multi-sourcing, partnership or joint venture, cloud sourcing, shared service, and the collaborative approach. Transitioning between different sourcing strategies is a complex process necessitating careful planning and execution. Key steps in a successful transition include assessing the current strategy's shortcomings, setting clear objectives, and developing a detailed roadmap.

Meeting an organization's IT service needs requires effective supplier evaluation and selection. This process begins with understanding and detailing IT requirements, including features and performance expectations, followed by supplier pre-qualification based on financial stability and market reputation. Subsequently, organizations should issue a request for information (RFI) and then a request for proposal (RFP), where suppliers detail their capabilities and associated costs. The evaluation phase is critical, encompassing technical assessments to verify a supplier's capabilities and ensuring cost-effectiveness. Following selection, contract negotiations take place and the service scope, costs, and timelines (including the stipulation of service level agreements and associated penalties or rewards) are outlined. The onboarding phase involves the facilitation of knowledge transfer, the integration of supplier and internal systems, and the setting up of any tools needed for collaboration. Continuous monitoring of the service is essential for maintaining standards and identifying areas for enhancement.

The role of consistent supplier monitoring and management cannot be overstated. Given that organizations lean heavily on various suppliers for key IT services, establishing mechanisms to vigilantly track and assess their performance is imperative. Effective supplier surveillance facilitates early identification and the resolution of issues before they escalate into significant problems. This sustained scrutiny ensures that suppliers uphold the quality and cost parameters that have been agreed upon, fostering a culture of ongoing quality assurance.

UNIT 5

DEVOPS: CONNECTING DEVELOPMENT AND OPERATIONS

STUDY GOALS

On completion of this unit, you will be able to ...

- describe the differences between software products and services and understand how they are developed.
- explain the shortcomings of a separation between development and operations teams in the context of software delivery.
- contrast the DevOps lifecycle with traditional software development.
- explain the automation practices that underpin DevOps and the relationships between them.
- understand how to configure service management processes to account for DevOps.

5. DEVOPS: CONNECTING DEVELOPMENT AND OPERATIONS

Introduction

In traditional IT companies, the creation and delivery of software was confined to two separate teams: development (Dev) and operations (Ops). On the surface, this setup meant that the Dev team would write code and then hand it over to the Ops team, who were responsible for deployment and support. However, that separation ran much deeper, impacting a number of dimensions. Different teams usually meant distinct organizational units or departments. This translated into separate objectives, ways of working, tools, knowledge, and even mindsets. At times, these differences became a major roadblock to effective collaboration and in certain circumstances fostered a culture of blame and finger-pointing.

In DevOps, the Dev and Ops teams share the same core objective – fast delivery of high-quality software to clients. It is present across the entire lifecycle – from strategy, to design, transition, operations, and continual improvement. Instead of them versus us, it's just us - one highly collaborative DevOps team.

The overhaul of collaboration between the development and operation teams explains the cultural side of DevOps (arguably the most important one), but there are also many methodological and technological aspects to be considered. One goal of DevOps is shift to small and frequent releases of new software features and fixes – a tenet of Agile development that often complements DevOps. Another is a high level of automation of processes with the aim of reducing human errors, lowering operational costs, and increasing quality. In this unit, we will explore some of the primary practices in this area, including continuous integration (CI), continuous delivery (CD), continuous deployment (CD), and infrastructure as code. While this disruptive approach might seem at odds with IT service management (ITSM), it is possible to adapt DevOps principles and methods in a company that embraces Information Technology Infrastructure Library (ITIL) practices. In this unit you will learn the basics of the software development and delivery process, how DevOps impacts it and explore the links with IT service management in the new technological era.

5.1 Development and Operation of Software in the Context of IT Management

In order to grasp the concepts covered in this unit, you will need a basic understanding of each keyword that appears in the title. Let's begin with the broadest definition of IT. Traditionally speaking, IT has two major constituents: products and services. Digital products

(like their physical counterparts) can be considered tangible — a deliverable is produced that can be then used by certain customers. Services, however, are more like the experience of consuming a product and can be considered intangible.

To make things less abstract, let's consider some real-life examples. Products include software like applications and computing systems, hardware like servers or networking equipment, and even something as simple as a document. All these products can be purchased at any time and consumed immediately or at a later point in time. On the other end of the IT spectrum are services, which can include maintaining the software and/or hardware elements that were bought beforehand. Other examples of IT services are risk assessments, trainings, and the broad field of IT consulting. All these services can be experienced only in the present.

What both products and services have in common is that for most part, they are the outcome of a project — either a development or a maintenance/operations one. For both types of projects, a certain management approach or framework is needed, as well as a team that executes it. However, the people involved in development and operations projects require different skills and perform different types of activities. These underlying activities were traditionally separated in terms of time, the teams that took care of them, and even their location. The main moment the corresponding teams got to communicate was a project handover.

Now that you understand the difference between products (software) and services (operations), let's explore how they both contribute to a traditional software development lifecycle (which requires management at every stage).

Traditional Software Development

The creation of software historically followed a sequential process. Distinct phases that a project needed to go through to yield working software (analysis, design, implementation, testing and deployment) occurred one after another and had well-defined exit criteria. This way of working relied on the assumption that requirements could be clarified upfront and would not evolve over time. As a result, there would be no need to revert to previous project phases. We know today that this is not a realistic scenario. Accordingly, modern approaches to software development embrace changing requirements and aim for regular delivery of working pieces of software. Nevertheless, many of the notions of traditional software development remain relevant — particularly the type of activities (represented as distinct project phases) that are necessary to create software.

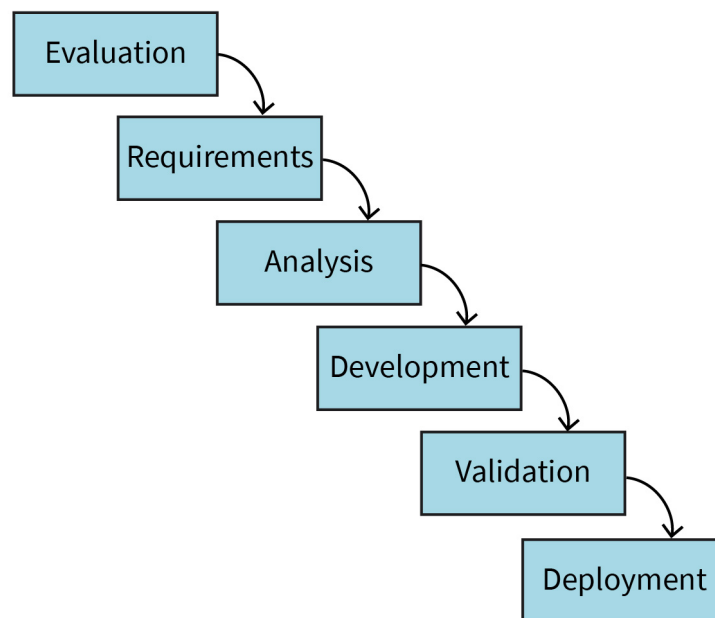
Everything begins with a business need that is expressed as requirements a product is designed to deliver. These can come from the people that will consume the product once it is delivered (end-users) or stakeholders that are knowledgeable about the problem the software will solve. With the project scope clear and set, the next stage is the solution design. Its architecture and other design artefacts are created. The project then moves to the development stage. The product team (usually developers) will work with the requirements and turn them into working

code — a tangible deliverable. The software artifacts need to meet a certain quality threshold before being released. This is ensured by various types of testing and validation. The result should be a production-ready product.

Once the testing stage is considered complete, the product is ready for delivery to end users. The software artifact is deployed to the production environment where its consumers can access and start using it. Whenever a large set of users start extensively employing a software, numerous bugs, issues, or even security breaches can be discovered and need to be addressed. This is when a maintenance team intervenes and fixes the problems detected and a new version of software is prepared and deployed into production.

The software development lifecycle we have just described is based on a waterfall model, as depicted in the following figure.

Figure 10: Waterfall Model



Source: Rafal Wlodarski (2023)

After any deployment, the live product needs to be operated on. This is what IT operations stands for and includes all the processes that support hardware and software that are used by the enterprise but also products that deliver services to customers of the company.

The main operational processes of can be distinguished as follows:

- Request fulfillment
- Incident management
- Problem management (postmortem)
- Configuration management
- Change management

The Role of Operations in the Software Development Lifecycle

In the traditional way of working, there are two common set-ups used to handle IT operations. Firstly, whenever companies do not have dedicated human resources or the project team or scope is relatively small, the development team also takes on any operations tasks. In larger organizations, it is common to see professionals who specialize in only one area of software creation – hence there are dedicated development, testing and operations teams. In the latter scenario, the operations team simply receives a new release of a software product and decides when it should be deployed so that there are no disruptions to the existing services.

Let's consider a simple practical example. Imagine an e-commerce web application that allows customers to order certain goods. Such a solution is underpinned by the working code (usually regrouped into separately deployed components – front-end and back-end in a simple client-server architecture). Its creation involves requirements engineering, design, development, and testing. This part of the software development lifecycle is usually the responsibility of a dedicated project team. Once the solution is considered production-ready, it can be deployed and maintained. This requires hardware and infrastructure resources (such as a web server and a database). IT operations have a major role in assuring the functionality and quality of these assets.

5.2 Characteristics and Shortcomings of a Separation Between Software Development and Operations

Before the proliferation of DevOps, the task of building software products involved managing the hardware needed to run them as well. In practice, that meant rooms dedicated to servers, which needed to be set up, monitored and updated to ensure their proper functioning. Whenever a large amount of such hardware needed to be managed, dedicated personnel would take care of it – the IT operations team.

On top of the hardware-related tasks, IT operations teams can be responsible for putting in place and ensuring the smooth operation of software. This includes deploying new software features, bug fixes and updates across environments as well as managing the release process. Safeguarding running software involves monitoring its performance and availability, responding to any issues in that area, and maintaining security measures.

Development of software was confined to a separate team of engineers who performed completely different types of tasks. Once a product was ready, they would hand it over to the operations team for hardware set up, deployment of the software (this mostly involved manually running commands on a server) and technical maintenance. Silos formed with little communication between the disparate teams.

While there is nothing wrong with such an approach per se, whenever the company experiences significant growth and products rapidly increase in number (leading to expansions in underlying hardware and infrastructure as well), things get out of hand. Each new software release takes more time and becomes less predictable. Given that humans make errors whenever manual work is involved, problems arise more often, which leads to less frequent releases and longer time to market. Teams grow unhappy and so do customers, who do not obtain their bug fixes in a timely manner.

Shortcomings of Separation

The shortcomings that are associated with a separation between software development and operations teams can be grouped according to four main themes, each of which we will explore in the following paragraphs.

Lengthy processes

When software development and operations are separated, a handover between the two teams is needed. As the work of operations and development is considered separate and loosely connected, a knowledge transfer is required for the operations team work independently. The technical handoff between the teams calls for a coordination for packaging, testing, and deployment of artefacts, which contributes to longer release cycles. Additionally, knowledge sharing with regard to functional aspects of the software system can be necessary (how to use it to obtain the desired functionality).

Another set of activities that takes more time than if the teams were working together is bug investigation and fixing. If the development team does not have direct access to the deployed systems, it can result in delayed identification and resolution of operational issues. Equally, the operations team might not be able to reproduce certain behaviors and made need to seek the development team's input and help. This needs to be formally planned and arranged as the teams are likely to have different ways of working and different schedules. As a result of the prolonged processes, the time to market of products and new features is longer, which can negatively impact the overall customer satisfaction and the competitiveness of a company.

Siloed responsibilities and lack of shared understanding

From a hierarchical point of view, having different teams intervene at different stages of a project lifecycle (the developers building new features and operations taking care of deployment and maintenance) naturally translates into separate sets of responsibilities. These usually come with differing priorities and perspectives. What is crucial to one team (stability and reliability for operations) can be considered a roadblock by others (reluctance to change hinders innovation and the creation process of the development team). Furthermore, developers may not understand operational considerations, and the operations team may have very limited insights into the software and how it was built.

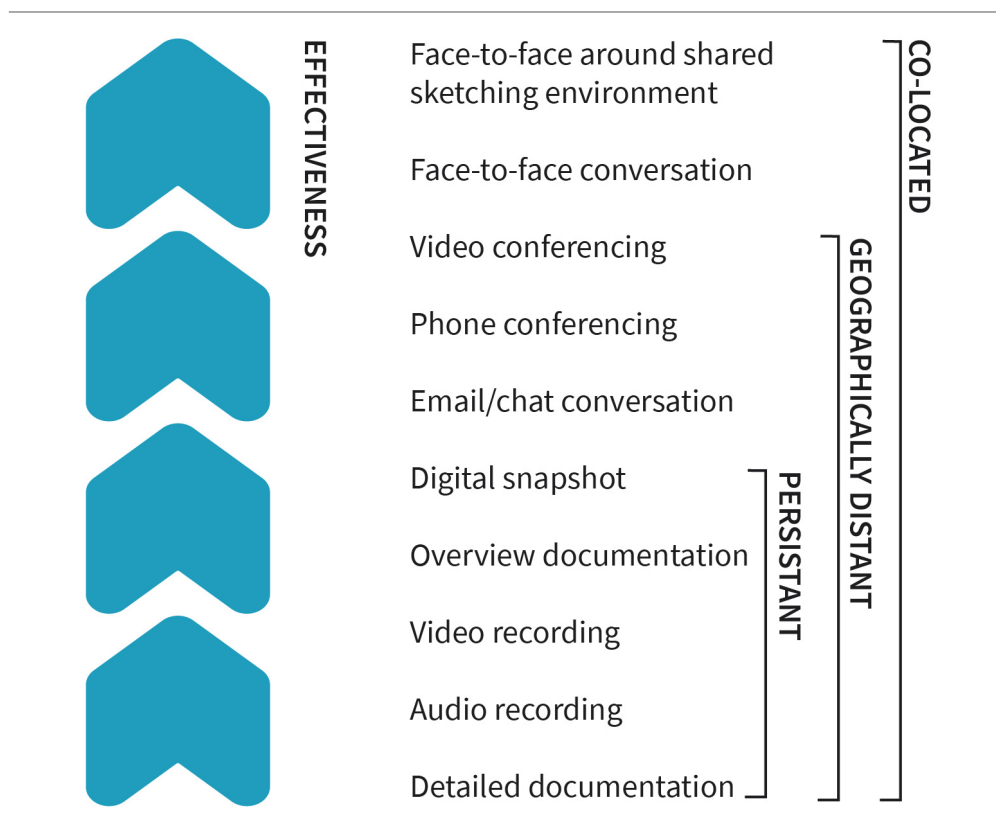
Such a disconnect can cause misalignment in goals of the respective teams, taking them away from what should be a shared goal: the success of the company or product. In the long run, an even more impactful consequence of this can be a blame culture. Whenever

difficult issues occur between the development and operations teams, it is easy to start finger-pointing about who's responsible for the error or omission that caused repercussions. Such a culture of blame can create a negative working environment and hinder effective problem resolution.

Less effective collaboration

In a traditional set-up, it is likely that development and operation teams work in different offices or departments, rather than working in the same space. Such an approach hinders collaboration as more effort is put into communication among employees as opposed to in face-to-face exchanges, as shown in the diagram below.

Figure 11: Effects of Proximity on Effectiveness of Different Communication Methods



Source: Rafal Wlodarski (2023), based on (Cockburn, A., 2000).

Developers may not have direct access to the production environment, making it difficult to receive timely feedback about how their code is performing in real-world conditions. Similarly, operations cannot just call out a name and let a team member know that there is a potential issue that needs to be analyzed. In both scenarios, the teams would end up sending an email or opening a ticket in a platform that was set up to facilitate collaboration. This is an inefficient means of communication. The less direct interaction takes place, the more delay there is in getting any results.

Longer feedback loop and improvement cycles

The competitive nature of the market puts pressure on teams to strive towards short turn-around times, fast deliveries, and short feedback cycles. Unfortunately, the separation between development and operations teams negatively impacts the feedback loop. Customer feedback cannot be continuously evaluated as product fixes and new increments take time to test and deploy. The release cycle is often long, which results in the development team making assumptions before the application deployment finally takes place.

Separation can also lead to a lack of knowledge-sharing between teams. Often, developers need to adopt innovative technologies and experiment with new approaches. In contrast, the operations team is responsible for maintaining the stability of hardware and software. When there are limited opportunities to spontaneously exchange information and to learn from each other, each team continues business as usual without any fresh insight and ideas for improvement. This results in a learning culture that hinders innovation and the ability to implement improvements effectively.

The aim of DevOps is to alleviate all the shortcomings we have just discussed by removing the barrier between development and operations and fostering close collaboration and short feedback loops. The promise of DevOps is that the quality of products will also improve since DevOps usually brings automations that enhance quality assurance, testing, and security practices. DevOps has become a buzzword that means many different things to many people. No two definitions are alike, but most of them do share one thing: culture. DevOps is a cultural transformation that brings people from disciplines that once used to be separate together.

5.3 The DevOps Idea: An Overview of the Concept and its Elements

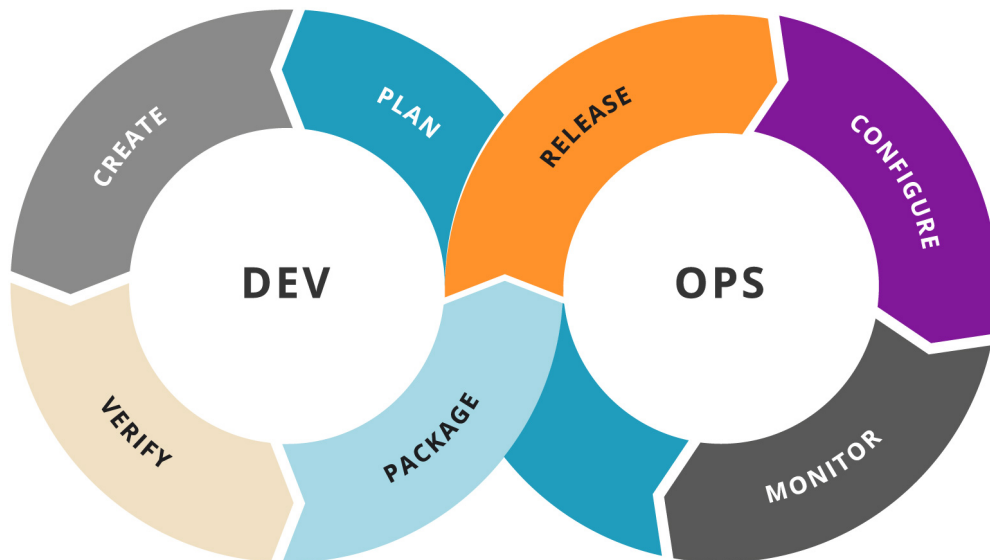
The DevOps approach means having the development and operations teams collaborate rather than working in silos. Dev-Ops is not a tool or a technology but a cultural and organizational dimension of product development and delivery. It relies on automation, short feedback loops, and release cycles to build and ship software and technology projects faster, more efficiently and more reliably. One of the factors that leads to the success of DevOps is its human aspect, which manifests itself through close collaboration within the team and the fostering of multi-disciplinary skills. How is this different from the blame culture we discussed earlier? In DevOps, the responsibility for completing a task is shared among the whole team rather than one person being accountable. Although there will likely be a particular individual working on the task in question, it's the entire team that ensures its success.

Apart from the cultural impact, DevOps completely changes the work processes of the people involved. Firstly, the lifecycle of a project is adapted. Secondly, a set of technical practices that enable automation of the lifecycle's underlying activities are introduced. We will now explore these aspects of DevOps in more detail.

The DevOps Lifecycle

DevOps impacts the whole life span of a product: from development through to delivery and maintenance. Due to its continuous nature, the DevOps lifecycle is often depicted as an infinity loop composed of eight phases, as shown in the figure below.

Figure 12: DevOps Lifecycle: Underlying Phases



Source: Kharnagy (2016), CC BY-SA 4.0

The eight phases represent the processes, capabilities, and tools that both the development (on the left side of the loop) and operations (on the right side of the loop) teams need for DevOps to succeed. Through each of the underlying phases, teams communicate, collaborate, and seek improvement.

Discover

The discover phase is when gathering inputs and feedback from key stakeholders takes place. In this highly interactive phase, the focus is on understanding the current processes and tools by capturing information in templates and checklists. This input leads to the compilation of a list of DevOps methods and practices that address existing challenges and constraints.

Plan

For the team to be effective, upcoming work needs to be identified, prioritized, and then its progress tracked. For each of these activities, an Agile approach to work should be applied and carried out according to one of the methods or frameworks, for example. **Scrum**. Common deliverables for this phase include a project roadmap and milestones, which are an outcome of collaboration between development and operations team as well as other stakeholders.

Scrum

This is an iterative and incremental Agile framework for software development. In contrast with traditional approaches, Scrum embraces changing requirements and close collaboration with the client. A Scrum team is self-organizing and multidisciplinary.

Build

This is when the actual solution is developed. The architecture is conceived, design artefacts produced, and the source code written. The outcomes of this phase are executable software artifacts or binaries. In terms of technical processes, this phase relies heavily on version control (managing source code changes and tracking different version to facilitate collaboration among team members) and continuous integration.

Test

The DevOps test strategy is about “shifting-left”. Simply put, this means testing early in the project lifecycle. The test should be written at the lowest level possible (for example, unit tests) and run automatically before any new feature is integrated into the code repository. Different types of automated testing should be carried out in order to ensure the software functions properly and meets quality standards in terms of security and performance.

Deploy

In the DevOps lifecycle, the application code moves through the release pipeline to the production environment continuously. Before reaching end users, each project increment typically involves a successful continuous integration build, followed by automated tests of different levels (unitary, integration, system tests) and executed in multiple environments (development, test, staging). Automation can be leveraged during this phase to push changes to production quickly and safely.

Operate

After a successful delivery of the solution, operations activities include maintaining, monitoring, and troubleshooting it in the production environment as well as managing all IT infrastructure.

Observe

DevOps focuses on observing the impact of the solution on its users and infrastructure. This requires collecting appropriate data (response times, memory utilization, network traffic) so that product uptime, speed, and functionality can be safeguarded. Other key elements that enable efficient monitoring of application’s health include automated alerting and notification mechanisms (which are activated whenever predefined limits are exceeded) as well as logging and tracing capabilities (which help with troubleshooting and debugging issues).

Continuous feedback

DevOps teams need to evaluate each release so that they can improve. Incorporating customer and developers feedback in terms of the solution, and delivery process is a key aspect of DevOps and Agile approaches. Short feedback loops help to ensure that teams continuously learn and incorporate information from end users and incidents.

As DevOps implies a multitude of tools and methods, it can be highly challenging to roll it out for the whole organization simultaneously. Implementing DevOps may require reorganizing teams, which can be challenging in large organizations with complex and/or hierarchical structures. Additionally, many enterprises outsource most of their IT. In these cases, an external company takes care of the operations activities, hence there is no corresponding structure or talent within the organization that can use DevOps tools and practices effectively. Other obstacles include legacy systems (whose replacement can be time-consuming and costly) and humans' resistance to change. It's advisable to start small – think a supporting application or service where the team could experiment with this new way of working. Adopting DevOps culture, practices, and tools will gradually help increase the team's confidence and ultimately achieve business goals faster.

DevOps Practices

The success of DevOps is strongly linked to the advanced automation mechanisms that underpin this new way of working. DevOps was not born overnight: It was an evolving process aimed at addressing certain difficulties of software development. Software systems are complex, and a simple change to a single file can have unintended side effects on the overall system. Therefore, when a team of developers work on related systems, the risk of breaking other's code increases significantly. What's more, as software-driven solutions involve a large number of files, including code, tests, and external dependencies and libraries. Coordinating artefact updates can be a daunting and costly task especially in large-scale or complex systems that involve multiple teams contributing to the project. To address such issues, a mechanism of nightly builds (creating software artefact from source code) followed by a set of automated tests was introduced back in the 90s. These were the underpinnings of continuous integration (CI).

So, what has CI become today? Put simply, it's a practice that automates a part of the project development lifecycle. Full automation of a project development lifecycle is achieved by putting in place what is called a continuous integration/continuous delivery (CI/CD) pipeline. It is essentially a runnable specification of steps performed to build a new version of software solution. This means that instead of doing it manually, the process is written programmatically and can be re-run whenever needed. Let's explore the pipeline further by exploring each of its elements.

Continuous integration

CI is the first part of any CI/CD pipeline. "Integration" describes the process of pulling pieces together, making sure different project artefacts bundle correctly and that the result is a functioning deliverable. In practice, this translates into combining the right version of libraries, specific toolkits, data sources and running automated tests that verify the outcome is as expected. With CI, this process is automated (thanks to an automation server or a cloud-based solution) and executed frequently (usually a couple of times per day but sometimes as often as upon every change to the code repository). The objective of the practice is early detection of issues and a fast feedback loop.

Continuous Delivery and Deployment

The CD part of the pipeline refers to either continuous delivery or continuous deployment, depending on how advanced an organization is in terms of the automation process. The next step after integration is releasing a project. This activity requires incorporating all necessary project files (source code, libraries, and configuration files) and preparing a runnable solution that the end-users or other systems can consume. This means a specific format is expected, which makes it possible to install the solution on a designated environment.

Continuous deployment is the process of automatically replacing an existing solution in a given environment (usually production), with a newly generated artefact (as the outcome of the continuous integration and delivery steps). Continuous delivery enables the release of a new version of the solution at any time, but the deployment still requires manual steps (like transferring artefacts from repository to a production server). Continuous deployment builds on the automated release process and automatically (without human intervention) takes care of the deployment of project artefacts to the target environment.

Infrastructure as Code

Now let's explore a further technical building block of DevOps. Infrastructure as code (IaC) is an IT infrastructure management practice that enables automated provisioning and handling of technical resources necessary to run software solutions. That encompasses anything from setting up servers, virtual machines, databases, networks to application configuration and deployment steps. The idea behind IaC is to write a (most commonly text) file containing code to define, deploy, update, and destroy a project's infrastructure. It is then executed and takes care of the system administration without human intervention. Before IAC was born, this was a core task of a dedicated operation team, and it was performed manually. This represents an important technological and organizational shift where handling of hardware is done in a programmatic, and thus software-oriented way. Some of the benefits of IaC include consistency across environments, rapid provisioning and configuration processes and easily achievable scalability of resources.

5.4 DevOps vs. IT Service Management: How to Connect the Approaches

Now that you understand what the software development process entails, including the underpinning of the DevOps approach and how it enables cross functional collaboration between the development and operations teams, let's turn our attention back to ITSM. Firstly, it's important to note that ITSM processes such as those prescribed by ITIL do not deal with the phases of software development lifecycle prior to the system deployment. However, **release management** and operating a running system or software can certainly be provided as a service, hence they fall into ITSM territory.

So, what is the link between DevOps and ITSM? There are five life cycle stages in ITIL: service strategy, service design, service transition, service operation and, continual service improvement. Of these stages, two are relevant to this discussion because their underlying activities overlap with the concerns of DevOps. Please see the tables below for an illustration of which DevOps activities (that are part of the observe phase) intertwine with relevant ITIL processes. The two approaches complement each other, as DevOps provides insights into the health and performance of software and IT services. Such data is important for effective ITSM as it enables informed decisions and effective problem-solving.

Release management
This is a process that encompasses planning, coordinating, and deploying new software versions, fixes and changes to the production environment. It is an integral part of ITSM because it ensures changes to IT services are rolled out in a systematic way and the results are aligned with business objectives.

Table 6: Overlap Between ITIL’s Service Design, Service Operations Stages and DevOps’ Monitoring Phase: Service Design

Service design	
<p>Availability management Ensuring that IT services are available and reliable by identifying and addressing points of failure.</p>	<p>Real-time monitoring Keeping track of the application's behaviour in production by means of metrics.</p> <p>Alerting and notification Use of automated alerting systems (which can be integrated with incident management tooling) when metric thresholds are exceeded or certain breaches (like security) take place.</p>
<p>Capacity management Ensuring that IT services have the resources needed to meet demand.</p>	<p>Performance monitoring Collecting and tracking of performance related-metrics in order to identify bottlenecks and potential optimizations.</p>
<p>IT service continuity management Ensuring that IT services can be rapidly restored quickly whenever a major disruption takes place.</p>	<p>Error monitoring Use of tools to track errors, exceptions, and crashes of a running system.</p> <p>Logging and tracing Collection and analysis of system and infrastructure logs historizing events and errors.</p>
<p>Information security management Ensuring proper security measures and assessments to protect confidentiality of assets.</p>	<p>Alerting and notification</p>

Source: Rafal Wlodarski (2023)

Table 7: Overlap Between ITIL’s Service Design, Service Operations Stages and DevOps’ Monitoring Phase: Service Operation

Service operation	
<p>Event management Monitoring events that take place in an IT environment.</p>	<p>Logging and tracing Alerting and notification</p>
<p>Incident management Restoring regular service operations rapidly following an incident or service disruption</p>	<p>Real-time monitoring Logging and tracing Alerting and notification</p>

Problem management

Identifying root causes of repeated incidents and taking counteractive measures.

Logging and tracing**Error monitoring**

Source: Rafal Wlodarski (2023).

Service level management is one of the processes in the ITIL service design phase that provides the most structure. Its objective is defining service level agreements (SLAs) that establish performance and quality targets for the services offered. These agreements need to account for the monitoring capabilities of the DevOps era and be enriched with new KPIs. Metrics related to performance (response time, resource utilization, throughput), availability (uptime and downtime, latency) and incidents (error rate, incident count) will likely remain similar to in traditional practices. However, due to the incorporation of DevOps practices their source or monitoring tools used can change.

The ITIL area that is more impacted by DevOps is release management, as use of automation practices around software delivery and deployment changes significantly. When drafting SLAs for services around software created and delivered with DevOps, five key performance metrics or key performance indicators (KPIs) should be considered. Each of these metrics will be outlined in the list below.

1. **Frequency of deployments:** Nowadays, the development of software is dominated by Agile methods (KPMG, 2017). These methods entail working in iterations, which are a time-boxed periods of time (typically 2-3 weeks; Sutherland, Schwaber 2020), within which teams create new features or bug fixes. A set of these are delivered to end users as part of the next release of a product. It's very important to release on a regular and frequent basis to minimize the scope of changes, and hence, potential maintenance and operations efforts. Releases can be scheduled on any daily basis (weekly or daily) but enterprises that fully capitalize on DevOps and its automation mechanisms such as continuous deployment, can see hundreds of releases of their product per day.
2. **Deployment time:** This metric measures the time needed to fulfill the whole deployment process, including the ready state of the infrastructure. This corresponds to the "deploy" part of the DevOps lifecycle discussed earlier. With a CI/CD pipeline in place, the execution time required for each step can be tracked (integration and artefacts preparation, testing, deployment, and provisioning of resources). Such granularity of information enables continuous improvement of the pipeline. As, over time, solutions grow, entailing more lengthy deployment processes (for example, due to a number of new tests being added) optimization of this metric is crucial.
3. **Deployment failure rate:** This metric refers to the rate of outages that take place due to deployments of a solution. While in a perfect world this metric should be equal to zero (as such downtime means the solution is not available to the end users), this is not very realistic and the target here is to keep the deployment failure rate as low as possible.

Even the most robust CI/CD pipelines fail from time to time for random reasons such as test flakiness (inconsistent or unreliable behavior of programmatic tests). With a high rate of changes deployments, the probability of failure naturally increases.

4. **Deployment failure detection time:** This metric is strongly related to the deployment failure rate and is considered one of the most crucial metrics in DevOps (Kaiser, 2023). Some failures will always occur: the key is to detect them as soon as possible and take mitigating actions to resolve the underlying issue and minimize the downtime. This KPI is sometimes referred to as mean time to recovery (MTTR).
5. **Change lead time.** This metric measures how much time elapses between the detection of an issue and the moment it is addressed and deployed to production. Changes usually mean bugs and issues that require code changes which must go through the whole CI/CD pipeline before being released. Shorter lead times are a good indication of an effective collaboration and workflows.

While the above list is a good starting point of metrics to include in ITSM processes in the DevOps era, it is by no means exhaustive. Enterprises should look for the most appropriate metrics for their context and potentially adapt the proposed set or introduce new KPIs. Nevertheless, the overarching advice here is to keep things simple. Each metric that makes it to a contractual agreement entails monitoring and reporting efforts that can become a pain point later on.



SUMMARY

Long gone are the days when development teams were in charge of creating software solutions or new features and the main responsibility of operations teams was setting up servers, rolling out new products to production, and maintaining the related infrastructure. The premise of DevOps is the integration of processes related to product development, delivery, and operations. DevOps is a portmanteau of development and operations (teams) and represents a combination of culture, practices, and tools that enable faster and more streamlined delivery of business value.

One practice that makes DevOps possible is the automation of processes. Manual and error prone tasks are replaced with a programmatic approach that is repeatable and reliable. Key practices in this area include continuous integration, continuous delivery/deployment, and infrastructure as code. In order to connect the practices and processes of service management and DevOps, both approaches need to be aligned. One of the key areas to pay attention to is service monitoring and continuous improvement. These areas are driven by KPIs and customer feedback.

UNIT 6

IT APPLICATION PORTFOLIO MANAGEMENT

STUDY GOALS

On completion of this unit, you will be able to ...

- describe the value of an application portfolio to IT and the many entities that participate in the optimization and growth of an organization's technologies.
- identify the key components to be collected when developing an application portfolio.
- define common metrics to determine the value of applications.
- name the common lifecycle disposition designations and their purpose.

6. IT APPLICATION PORTFOLIO MANAGEMENT

Case Study

Since its inception, ABC Company has grown through mergers and acquisitions. ABC has many applications, although the exact number is unknown. The business identifies gaps in automation while identifying redundancies in data entry and business process steps. The information technology (IT) organization struggles to provide the necessary level of service to operate and maintain the applications. Each year's budget planning process references the expenses of the previous year without a true understanding of what could be optimized and improved. The expenses continue to grow because the technologies underlying the applications are at the end of their lifecycle, and support resources are unavailable.

The current technology landscape complicates ABC Company's strategic plan for improvements. The complexity of the application portfolio puts significant demands on organizational resources. Gaps and overlaps in applications create a technological environment that does not facilitate the integration of new applications and data.

In addition to improving the planning for future development, many of the ITIL practices will benefit from a comprehensive understanding of the business applications that support the business capabilities and business processes of ABC Company.

6.1 Overview of IT Application Portfolio Management

Organizations use business software, or applications, to perform a business purpose. Applications gather, update, and manage data, producing valuable information for the operation of the business. An application may consist of one to hundreds of individual software programs or components assembled to provide the expected automation (Van Haren Publishing, 2018).

History

Over the past few decades, organizations have faced many new business requirements due to new business processes, new regulatory requirements, mergers and acquisitions, and more. IT organizations rapidly addressed these requirements, by acquiring or building new software and modifying existing software.

As with all things, creating small software patches or larger software bolt-ons added significant complexity to the software. As time passed, modification of existing software was further complicated as those familiar with the software, e.g., the original architects and authors, left the organization. In addition, emerging technologies and new software languages, such as Python, were more attractive to software developers, causing older programming languages, such as COBOL, to fall out of favor. However, many organizations' older software applications are written in older programming languages (Axelerate, n.d.).

As a result, IT would build or purchase a new application rather than modify an existing related application. This often resulted in multiple applications automating the same business function. The business groups also contributed to the uncontrolled growth of business applications. As business users became more technologically knowledgeable, business departments acquired software applications to automate their specific needs, resulting in duplicate applications (O'Quinn, 2018). This was further exacerbated by the increase in mergers and acquisitions that have taken place across industries.

Organizations will each have a full complement of business applications. Depending on the operational model of the new organization, there will be a varying degree of overlap (O'Quinn, 2018). Other common scenarios include software applications that are acquired and abandoned or become unable to integrate with existing software applications or incompatible with newer infrastructure (Axelerate, n.d.).

Over time, business users may transition away from the older applications. However, the applications may still be installed and operational, consuming disk space and processing power. Further, software service contracts may still be in existence, resulting in an item of unnecessary expenditure for a software application that is unused (Axelerate, n.d.).

In addition to the logistical concerns of a growing technological landscape, the business needs to control operating expenditures while expecting IT services that meet current business needs and position the organization for future growth (Axelerate, n.d.). Application development and support can consume a third of an IT operating budget. Yet, many companies accept the expense because of the need to automate their business processes; however, they do not have a strategy for overseeing its ongoing value (Gruia, 2014).

Application Portfolio Management

Using concepts from project and investment portfolio management, application portfolio management (APM) came into existence. Adopted in the late 1980s, organizations recognized the need to describe their inventory of business applications (Van Haren Publishing, 2018). This requirement was crucial as organizations approached the year 2000 and the concern about the failures that potentially would result from software and its associated data based on a two-digit year rather than a four-digit year (Van Haren Publishing, 2018). During the time that these software applications were written, disk storage was a significant concern. Disk drives held megabytes, not terabytes of data, and were expensive. For example, IBM's first disk drive that exceeded 1GB was released in 1980, could store 2.5GB of data, weighed 550 pounds (250 kg), was the size of a refrigerator, and cost USD 81,000 (Solarwinds Pingdom, 2019).

Organizations understood the need to create a complete inventory of business applications with metadata that identified essential aspects of each application (Van Haren Publishing, 2018).

Y2K
The year 2000 was feared to cause problems with software applications written in the 1900s, as a two-digit year was commonly used to store dates. Program logic was built to assume that the two-digit year reflected a 19xx date.

Whether driven by the **Y2K** concerns or general business needs, IT organizations and business groups understood the need to eliminate the inefficiencies of duplicate software and the infrastructure to support it (Upadrista, 2015). In addition to the cost-based benefits, organizations also needed to understand the positive or negative effects of applications on business value, risk, and alignment with strategy (Bodenstaff & Quartel, 2016).

With APM's capabilities, organizations can build a picture of the applications supporting business functions. The portfolio can then be analyzed to make meaningful decisions for application rationalization and future development planning.

Benefits

Organizations that have invested in an APM platform and the discipline accompanying an APM practice have realized many benefits from creating a single source of truth of the organization's applications and continuous lifecycle management. Axelerate (n.d.) identifies many of the benefits achieved with APM:

- application alignment to determine which applications support which business capabilities and which ones do not have a current purpose.
- optimized software inventory to reduce duplicity of applications that serve the same purpose and consolidate software vendors when possible.
- optimized infrastructure by reducing applications, in turn reducing the demand for computer processing power, disk storage, and human resources to manage the environment.
- reduced technical risks. Older applications and applications that have been sunset by software vendors are considered risks when they cannot be maintained to meet security and regulatory requirements.
- reduced costs. The reductions itemized above result in reducing IT's budgeted operating expenses.
- optimization of business processes. Duplicate applications often cause redundant data entry steps. Rationalization of the application portfolio may result in the optimization of business processes.
- identification of gaps in automation. Business capabilities that are not met by present business applications are more easily identified. This also supports better decision-making for obtaining cloud services and the acquisition of new business applications.

APM and ITIL

APM provides input for many of ITIL's practices. Within the general management practices, strategy management provides input to APM to understand the business and technical value of applications. Risk management should include an assessment of application risks. The output of the APM disposition determination provides input to architecture,

portfolio, and project management and supplier and workforce management practices. In the service management and technical groups of practices, the APM portfolio provides an organized, agreed-upon baseline of application inventory data for each of the practices.

6.2 Application Manual

The general process for APM includes the development of an inventory of applications and the **metadata** associated with each application. Technical and business scoring of the application can then be done in preparation for the analysis of the portfolio, covered in the next section.

Metadata
These are data that describe data.

Creating an Inventory

The discovery process for the application inventory should include applications in operation, whether on-premises or in the cloud, applications in development, and applications planned for development or acquisition (Simon et al., 2010). For each application, Simon et al. (2010) recommend capturing the data elements in the following table.

Table 8: APM Data Elements

General information	
	Application name Release version Implementation date Upgrade and patch history
Stakeholders	
	Application owner Business units supported Number of application users
Business domain	
	Business capabilities enabled Business processes supported
Technology domains	
	Data objects create, read, update, delete (CRUD) Operating system Other technical components
If homegrown	
	Cost of development Ongoing internal operational costs
If COTS or SaaS	

	Vendor contact information User groups Cost of application acquisition Key contractual statements Annual support costs
Operations	
	Operational performance Primary application support contact Primary technical support contact

Source: Joan Lawson (2022), based on Simon et al. (2010).

Stakeholders of the application include the application owner and the business users. The process of identifying stakeholders occasionally highlights that an application owner or sponsor no longer exists and that the application has few active users. The stakeholders are the initial source of gathering other information for the inventory.

Data collection approaches

Given the organizational landscape depicted above, assembling a complete list of applications operational throughout the organization can be a challenging undertaking. Collecting data can include a mix of automated and manual efforts based on what is available within the organization.

Automated data collection requires software tools that can intelligently identify business applications operating on organizational servers and gather available data elements (Simon et al., 2010). Automated data collection is best for identifying what applications exist and where they are running. Manual data collection is the bulk of the effort, requiring gathering information about each application from business and technical users (Simon et al., 2010). The deliverable of the data collection is as complete and accurate an application inventory as is appropriate for the organization to conduct a valuable portfolio analysis (described in the next section) and to plan for future growth and development efforts (described in the last section) (Simon et al., 2010).

Grouping applications

Grouping related applications into meaningful portfolios provides context to the inventory. Groupings may be done by the application owner, relationship to a business process or capability, or assigned to a specific technology.

IT Enterprise Architecture and APM

APM is a natural connection between IT enterprise architecture (EA), the chief information officer (CIO), and business units. EA needs to gather information to create a view of the current state architecture across the four domains of business, data, applications, and technical. EA is chartered with an understanding of the current state and the business

direction to develop a strategy and architecture for the future state. EA needs to govern existing technologies in an oversight capacity to ensure consistency with the EA principles, standards, and future architecture (Van Haren Publishing, 2018).

Business domain

The CIO's role is to connect with the business units to ensure that technologies align with business capabilities and needs, enable future business growth, and meet the organization's budget and service levels. Aligning business applications with the business capabilities in the business capability map identifies overlaps and gaps in the automation of business capabilities. The analysis of the portfolio in APM relies on inputs from the CIO and business to assess the strength of each application in the portfolio. Last, the collective knowledge of all three disciplines results in an optimal plan for applications that best meet the needs of the business while doing so cost-effectively and efficiently (Van Haren Publishing, 2018).

Technology domains

EA requires an inventory of all technology components. APM provides the inventory of business applications as a foundation for identifying the data that are used by each application. For each inventoried application, the technical components such as server, operating system, database management system, and **web server**, should be identified.

Web server
This platform software manages processing with HTTP requests.

The further APM processes of portfolio analysis, the resultant application rationalization, and future development planning are also consistent with the objectives of the EA practice.

Application Portfolio Management Maturity

Significant organizational undertakings such as ITIL, EA, and APM, require a considerable investment of dollars and efforts. Assessing the maturity of these practices enables organizational leaders to understand the improvements that have been made and the opportunities for future enhancements (Simon et al., 2010).

Simon et al. (2010) propose an APM maturity model with the following levels:

- Level 0—Application Portfolio Obscurity. This is the starting level for many organizations experiencing the symptoms of a lack of management of the business applications, as described above.
- Level 1—Application Portfolio Understanding is the level of organizations that have built an inventory of business applications, as described in this section.
- Level 2—Application Portfolio Intelligence is the level of organizations that continually evaluate the inventory of business applications to analyze their deficiencies to identify areas for improvements. This is further described in the next section.
- Level 3—Application Portfolio Quality signals an organization achieving the expected benefits from an optimized application portfolio.

- Level 4—Application Portfolio Excellence is the level achieved by organizations that continue APM on a continual basis and use an optimized portfolio to make future software investments. This is further described in the last section.
- Level 5—IT Portfolio Excellence furthers the benefits of APM by incorporating the same APM process throughout the full technology landscape.

6.3 Portfolio Analysis

Once inventoried, applications can be evaluated and scored on factors such as usage, quality, business value, technical value, risk, and cost (Axelerate, n.d.; Capture, n.d.). The result of the scoring can then be used to classify the application and plan for its future lifecycle.

Measurement and Application Scoring

An organization will choose the most important metrics to evaluate the business applications in its application inventory. Each organization will decide what and how it wants to conduct the analysis. The following sections will discuss analyses that are common to organizations.

Business value

As the primary role of business applications is to support the business, understanding the value of the application to the organization should be the initial consideration. The following value points need to be considered:

- Business capabilities describe what a business does operationally to conduct business successfully. Examples of business capabilities include “sell product,” “sell services,” and “manage receivables.” How well does each application automate one or more business capabilities (LeanIX, n.d.-a)?
- Business processes describe how a business conducts business. Examples include “bill customers,” “receive cash payments,” and “reconcile credit card receipts and credits.” How well does each application automate one or more business processes by enabling users to accomplish regular tasks (Simon et al., 2010)?
- Strategic efforts ask, “Is the application consistent with the organization’s strategic goals (LeanIX, n.d.-a)?”
- Regulatory compliance is according to governing bodies, who set forth guidelines or laws. Business applications must contain logic that is current and consistent with existing regulations in the jurisdictions in which the organization operates (Simon et al., 2010).

Technical value

The technical value of an application indicates its technical health and strength. A high technical value indicates that an application that runs efficiently can be maintained and upgraded and meets the service levels of the business users. The following value points need to be considered:

- source code (homegrown software). The size and complexity of the source code need to be understood to evaluate the ease with which future modifications can be made. Also, what technologies underlie the application, are they current, and can support personnel be identified?
- software (COTS, SaaS, and **OSS**). A vendor should be identified and contracted for ongoing upgrades, maintenance, and support of software that is obtained from a third party as commercial-off-the-shelf (COTS) or software as a service (SaaS). Participation in an active user group allows for sharing information with business peers. OSS requires a viable support community for product improvements and informal support. Optionally, a vendor that provides upgrades, maintenance, and support is most beneficial (Simon et al., 2010).
- integration. An organization should understand its requirements for the processes and data that span multiple applications. On what technologies are the applications built, and can they provide an integration capability (Simon et al., 2010)?
- service level. Service data such as the “number of service requests, trouble tickets and failures, response times, and availability [Maizlish and Handler, 2005; Weill and Vitale, 1999]” (Simon et al., 2010, p. 42) should be gathered to determine the level of service realized by business users as compared to the expected service levels.

OSS

Open-source software is often developed collaboratively. The source code is made available to others for modification and use at no cost.

Cost of ownership

An application’s cost of ownership is based on the annual operating expenses for the application beyond the acquisition investment. Simon et al. (2010) identify key costs of ownership to include

- support, maintenance, and upgrade costs for homegrown software include personnel, while operating expenses for SaaS, COTS, and OSS include service contracts with software vendors for the same purpose.
- operational costs for on-premises software include the hardware, software, and storage upon which the application resides and operates.
- depreciation of capital investment from purchasing the software or fees for SaaS-based software (Simon et al., 2010).

Risk

Together with other organizational risk assessments, an analysis of application risks should be conducted to determine the impact that the failure of an application will have on the organization (Simon et al., 2010). For example, an organization that conducts 80 percent of its business through online sales would be far more impacted by the downtime of its **e-commerce** site than an organization that performs 10 percent of its business through online sales.

E-commerce

The buying and selling of goods and services over the internet is called e-commerce.

Analyzing Application Value

Once an inventory has been created and each application assessed, the next step is to analyze the application's value in the organizational landscape and ultimately determine the application's lifecycle within the organization, thereby rationalizing the business applications so that the application inventory best serves the business and meets technology standards (Upadrista, 2015). The Strategic Grid Application Portfolio Matrix by Ward et al. (2002, as cited in Raza, 2021) is one of many approaches to analyzing the value of applications across four quadrants.

Table 9: Application Analysis

	Importance for future business	Importance for current business	Focus
Key operational applications	Low	High	Effectivity
Support applications	Low	Low	Effectivity
Strategic applications	High	High	Speed, features
High potential applications	High	Low	Prototypes

Source: Joan Lawson (2022), based on Raza (2021).

Categorizing applications into four categories driven by the application's importance for future and current business gives insight into the weight that should be given to each of the metrics described previously.

Lifecycle disposition

The results of the analysis lead to decisions on the lifecycle disposition of the application. Axelerate (n.d.) identifies common lifecycle dispositions for applications as follows:

- **invest.** The application is valuable and supports additional investments to enhance its value. Investment is typically triggered by new business requirements.
- **maintain.** The application is valuable and supports necessary investments to maintain its value. Investment can be triggered by business or technical requirements.
- **retire.** The application has been identified as unnecessary and will be retired. This might require the transition of any existing users to other business applications or different processes.
- **replace.** The application has diminished in business or technical value. However, its purpose still exists, and therefore it must be replaced.

Optimization

Optimization of the application portfolio is the phase of APM that puts the analysis and determination into action. A roadmap needs to be developed by the business units and IT to determine a well-coordinated approach to the retirement or replacement of applications and planned investments for those applications that will be enhanced. Optimization should be accomplished with the organization's project portfolio management efforts so that dependencies are well-understood and planned for (Simon et al., 2010).

6.4 Development Planning

APM enables organizations to plan future application investments at a macro level by rationalizing and optimizing current applications, better positioning the organization for the future.

Planning

Organizations use APM as input to the annual planning and budgeting process to understand where organizational dollars are required for technology investment, maintenance, retirement, and replacement efforts (Thinking Portfolio, 2013). APM enhances the complete picture required for planning efforts. An organization's short- and long-term goals, project portfolio, and application portfolio optimization efforts can be balanced to ensure alignment with the business needs and the availability of budget and resources (Joshi, 2020).

Manage Your Portfolio

APM promotes the continual management of the IT portfolio so that thoughtful investment decisions are made based on identifying and analyzing an application's business value, technical value, cost of ownership, and risk (Joshi, 2020). Portfolio management also encourages collaboration between the business and IT to identify and determine the value of business applications.

Forward-Thinking

Once the baseline of APM data has been established, APM is a forward-thinking discipline that enables an organization to manage its current portfolio of applications efficiently and effectively. An optimized portfolio results in the best use of the organization's technology budget to maintain existing environments and build a modern, scalable technology landscape that enables its projects and plans for growth (Capture, n.d.).



SUMMARY

Application Portfolio Management (APM) is a valuable discipline central to IT, EA, and ITIL. Organization application inventories have grown with minimal boundaries as they add the necessary automation to meet business needs. However, that lack of control has resulted in excess costs, excess supporting environments, poor service levels, redundant work, and complex positioning for incorporating new technologies. The APM discipline includes the creation of an inventory of business applications, the analysis and assessment of the applications based on business- and technology-defined metrics, the determination of how the applications can be rationalized to best serve the organization in the present, and how it can best position the organization for technological growth to meet strategic plans for the future.

UNIT 7

IT ARCHITECTURE MANAGEMENT BASICS AND TERMS

STUDY GOALS

On completion of this unit, you will be able to ...

- detail key aspects of IT enterprise architecture management
- describe the difference between IT architecture, IT enterprise architecture, and domain-level architectures.
- discuss business and IT goals of enterprise architecture management.
- identify the intersection of IT enterprise architecture management with other essential business practices.

7. IT ARCHITECTURE MANAGEMENT BASICS AND TERMS

Case Study

ABC Company has been in operation since the early 1970s. ABC expanded quickly and acquired an extensive portfolio of business applications, some operating in the corporate data center, while others operate on personal computers in the business departments. Many of the business applications are not integrated, and, as a result, the automation of business processes often includes redundant steps with data duplicated across the organization. Last, to support this complex environment, the data center houses mainframe and mini-computers, some of which are overburdened, while others are underutilized.

ABC's IT organization has a broad range of skills, excelling in many technical capabilities. IT has not previously been invited to the organization's strategy discussions. Unfortunately, IT has then acted in the capacity of order-taker, reacting to lists of requests from the business leaders and resulting in the implementation of targeted solutions. IT has not been able to contribute to the organization by creating a holistic, scalable, agile architecture that positions ABC's technologies for the future.

7.1 IT Enterprise Architecture

Information technology enterprise architecture (IT EA) is often compared to the building or expansion of a house. The first steps that an architect takes are to understand the current environment (the house as it exists today or an empty spot of land) and to understand the current and future needs of the homeowner. The architect will then design a high-level plan for the new home. Once the homeowner accepts it, the high-level plan will be broken up into lower-level designs for structural engineering, mechanicals, electrical, etc.

In a similar manner, the term IT architecture refers to the plan for technologies that provide support to an organization's business functions, while the term IT EA refers to the alignment of the IT architecture to the organization's business objectives and capabilities (Gambit, n.d.). IT must understand the current environment of the technologies that exist today to serve the business automation needs of the business departments, as well as the current and future needs of the organization and how the current environment addresses those needs. IT architects will then design a high-level future environment, referred to as an IT architecture. This can then be broken up into the various IT sub-architectures required to construct and implement the solutions.

Many sub-architectures comprise the overall architecture. This describes the technology landscape that automates an organization's business needs. Business, applications, data, and technology are four standard sub-architectures of the overall architecture of the

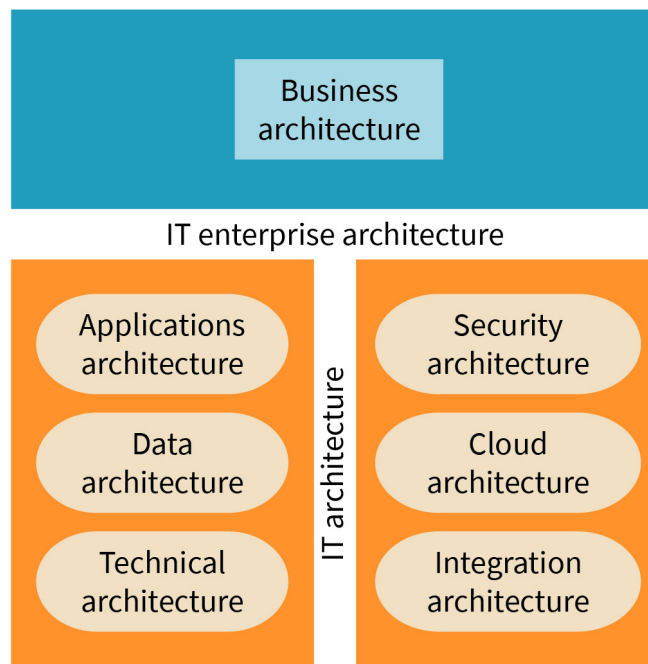
organization. Each of the standard sub-architectures depicts the architecture of the corresponding domain. For example, the applications architecture describes the technologies of the application domain.

Examples of other specific architectures include security architecture, integration architecture, and cloud architecture. IT architecture consists of all the stated sub-architectures except for the business architecture. IT architecture aligned with the business architecture is referred to as the IT enterprise architecture.

The products of the IT enterprise architecture management practice primarily include (Thabit, 2011; Turban et al., 2013)

- standards, principles, policies, models, frameworks, and methodologies to govern how IT components are introduced, designed, built, implemented, managed, and modified.
- a blueprint design for the future state architecture of software and hardware components.
- a roadmap that guides the transition from the technology's current state architecture to the future state architecture.

Figure 13: IT Enterprise Architecture Components



Source: Joan Lawson (2022).

Business Architecture

Business architecture is represented by the design of what an organization does to achieve its business goals. The business architecture is a blueprint of the organization, considering internal (organization, products, etc.) and external (customers, suppliers, etc.) con-

stituents and factors to present a common understanding of the organization and upon which strategic goals and tactical demands can be based. The Business Architecture Guild (2020) defines business architecture as representing the “holistic, multidimensional business views of: capabilities, end-to-end value delivery, information, and organizational structure; and the relationships among these business views and strategies, products, policies, initiatives, and stakeholders” (p. 2). The Guild further identifies the value of business architecture as providing “an abstract representation of an enterprise and the business ecosystem in which it operates” (p. 2). They clarify how business architecture delivers value “as an effective communication and analytical framework for translating strategy into actionable initiatives” (p. 2). The Guild summarizes the value of business architecture as enhancing “the enterprise’s capacity to enact transformational change, navigate complexity, reduce risk, make informed decisions, align diverse stakeholders to a shared vision of the future and leverage technology more effectively” (p. 2).

Ulrich and Kuehn (2015) state that the business architecture provides a detailed understanding of the organization which can be used for IT enterprise architecture to

- provide a context for the definition of business requirements.
- assess IT investments.
- determine cross-program, cross-project impact, and dependencies (pp. 3–4).

In addition to aiding in developing IT enterprise architecture, the business architecture enhances business operations. These business-related benefits include the ability to (Ulrich & Kuehn, 2019, pp. 3–4)

- align strategies and plans across business units.
- determine cross-program, cross-project impact, and dependencies.
- assess the impact of regulatory and policy changes.
- enable strategic business transformation.
- integrate companies during mergers or acquisitions.

Data Architecture

Enterprises are inundated with data that are collected, stored, and manipulated to create useful information. Data are collected from business applications, such as new customer demographics and sales orders, and used throughout the organization for downstream processes, such as order shipments. They are then used to generate valuable information for the necessary business departments, such as sales tracking and order history. A data architecture describes the data and information of an organization, and how they are structured, stored, accessed, and managed. It further specifies how data are aligned to business and application architectures.

The Data Management Body of Knowledge describes data architecture as “an integrated set of specification artifacts used to define data requirements, guide integration and control of data assets, and align data investments with business strategy. It is also an integrated collection of master blueprints at different levels of abstraction” (Mosley & Brackett,

2010, section 4.1). The construct of a data architecture includes “formal data names, comprehensive data definitions, effective data structures, precise data integrity rules, and robust data documentation” (section 4.1).

Technopedia (2017) further explains data architecture as “a broad term that refers to all of the processes and methodologies that address data at rest, data in motion, data sets and how these relate to data dependent processes and applications” (Technopedia explains section). As part of the development of IT EA, an understanding of data architecture is also used to

- depict the flow of information between people and processes;
- drive data quality and data integrity;
- enable integration of data and processes across the organization;
- help enforce enforcement of required security; and
- reduce data redundancy (Mosley & Brackett, 2010, chapter 4).

Application Architecture

Business applications support or automate the functions of the organization. Business applications can be small and self-contained, for example, an application that operates on a personal computer to create a marketing campaign. Business applications can be large and encompass many departments in the organization, for example, an enterprise resource planning (ERP) application that includes modules for accounting, human resource management, order processing, purchasing, and more.

Business applications might be homegrown customer applications written by in-house developers or contractors, applications purchased as commercial off-the-shelf (**COTS**) applications from software vendors, or COTS applications offered as Software as a Service (SaaS).

COTS

This is a standard software application written by a software vendor and licensed for use by customer.

Application architecture specifies how applications (Millares, 2020)

- are aligned to the business architecture,
- are designed,
- interact with one another, and
- interact with the organization’s data.

As part of the development of IT enterprise architecture, The Open Group (n.d.-a) in The Open Group Architecture Framework (TOGAF) details that an understanding of application architecture is used to

- depict processes between people
- enable integration of data and processes across the organization
- help enforce enforcement of required security
- reduce application overlap and redundancy
- ensure that the application portfolio evolves in a manner consistent with the business, data, technology, and other applications

- better understand the detailed application-level design of the services and components to enable coding to begin
- enable the reuse of software services and interfaces to improve developer productivity, application agility, quality, and consistency (Section 35.6.5).

Technical Architecture

Technical architecture describes the hardware, software, and networks that enable the business applications to operate, the data to be stored and used, and interfaces provided to the application users. The Open Group (n.d.-a) in TOGAF details that technical architecture includes

- infrastructure, such as computer servers, personal devices, and storage devices.
- system software, such as operating systems, web servers, application servers, and database management software.
- middleware used to automate the integration of data between databases. This can include messaging and file transfer software.
- networks and communications used for data transmission (Section 35.6.6).

7.2 Goals of Enterprise Architecture Management

IT enterprise architecture management (IT EAM) results in the building of an IT enterprise architecture where technologies are aligned to support the objectives and capabilities of the organization. Therefore, EAM results in benefits to both the business and technology stakeholders. Benefits differ by organization; however, many benefits shared across organizations will be presented in this section.

Alignment

The fundamental premise of EAM is the alignment of technologies to an organization's needs. Internal factors, such as business objectives, goals, current capabilities, and planned business growth, are considered together with external factors, such as mergers and acquisitions, competition, legal requirements, and regulatory requirements. Knowledge of these internal and external factors enables IT to position technologies to provide the necessary automation (Ahlemann et al., 2012).

Alignment within the IT architecture is also necessary to ensure that the applications and data support the business requirements, while the technical components support the infrastructure requirements for the operation of the applications, data storage, and data access. The alignment and progressive improvements of the business and IT architectures need to be described across common timelines to ensure the preparedness of IT for organizational change (Ahlemann et al., 2012).

Agility

EAM encourages the use of “as a service” (**XaaS**) capabilities to offer scalability, resulting in an organization that is more amenable to change, a necessary attribute in current times. XaaS can be leveraged in many areas of the architecture, from SaaS and infrastructure as a service to security and disaster recovery as a service (Ahlemann et al., 2012).

XaaS
This acronym generalizes all cloud-based services.

EAM also includes the awareness of new and emerging technologies that can be used in transformative ways. For example, smart solutions based on the Internet of Things (IoT) are being identified for many industries to add novel approaches based on sensors, actuators, and artificial intelligence.

Standards

Legacy technologies present a variety of complexities to many organizations that have been in existence prior to the 2000s. Application complexities may include homegrown business applications, shadow IT solutions, redundant automation, lack of documentation, and applications no longer supported by the COTS vendor. Data challenges may include redundant data, siloed data, lack of proper data security, and poor data quality. Technical challenges may include underused and overused computer servers, equipment no longer supported by the original vendor, and lack of documentation.

EAM promotes the standardization of the computing environment, resulting in the simplification of the environment, maximization of resource usage, and reduction of operating expenditures. Further, optimization results in a workforce with targeted skills that best serve the environment (Ahlemann et al., 2012).

Improved Insight

Expanding on the alignment, agility, and standards benefits, EAM results in a data landscape positioned to provide valuable information and improved insights to organizational stakeholders. EAM promotes data governance practices to oversee information quality by using elements of “data quality, data management, policy development, business process improvement, and compliance and risk management” (Smallwood, 2014, Chapter 2).

Other Goals

Budgets

EAM supports IT’s need to identify and plan for the life expectancy of necessary technology components. This is used in conjunction with plans for new technologies to provide essential information to an organization’s budgeting and planning cycle for required technology acquisition.

Quality of service

Known as the “-ilities,” EAM results in an improvement in the quality of technology service. This includes quality of service factors such as availability, reliability, maintainability, portability, and durability.

Communication

Documentation of current and future state architecture is a necessary process for EAM. Doing so provides knowledge to stakeholders and promotes a shared context for understanding the current state and envisioning the future state. Overall, this results in an improved communication channel between the business and IT groups.

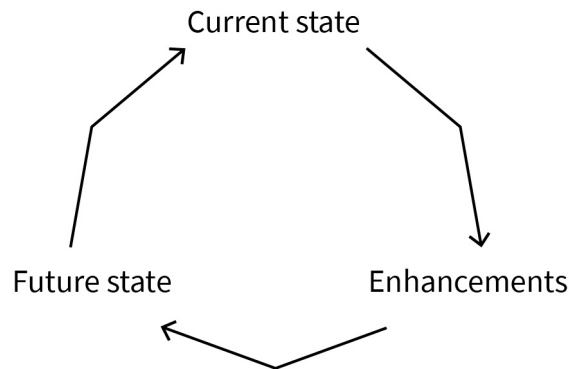
7.3 Processes in the Management of IT Enterprise Architectures

EAM processes begin with those that result in the iterative creation of the future state architecture and roadmap. In the following, the inclusion of the products of the EAM discipline in other technology and business disciplines will be presented.

Creating a Future State Architecture

The goal of EAM is to develop a future state, improved architecture, and a roadmap to implement the future state architecture. However, creating these products requires knowing the IT architecture’s current state, or baseline. The current state architecture provides input to the cyclical process of creating a new, future state architecture. Gaps between the current and future state architectures are addressed when designing a future state architecture with the current state of the sub-architectures providing input and the improved sub-architectures as the output. The future state architecture, now improved, becomes the current state once the technology solutions defined for the future state have been implemented. The current and future architectures include details of all four domains.

Figure 14: Evolving Architecture



Source: Joan Lawson (2022).

Business domain and architecture

To develop a comprehensive understanding of the enterprise, many views of the organization can be created. The creation of current state views is typically driven by the absence of a thorough understanding required for future development efforts. Typical views of the enterprise created as part of a business architecture include (The Open Group, n.d.-a)

- business strategy: “a coordinated set of choices that positions an organization to generate superior long-term financial or social returns” (Pham et al., 2013, Section 2.2).
- business capabilities: identification of what a business does. Business capabilities are implemented through business processes, which define how the enterprise does what it does.
- value stream: how a business delivers value to its key stakeholders through its products and services.
- business knowledge: semantics (customer, order, supplier) within an organization and their relationship (customer name, order number as the relationship established by a purchase made by a customer).
- business vocabulary: the common terminology used by the business together with definitions.
- organization: the organizational structure of the business, including both internal (employees) and external (customers, suppliers, business partners) actors.

Data domain and architecture

The data architecture describes the flow of data through the organization and includes the use of data to create valuable information for the business. Common artifacts created to describe an organization’s data architecture include

- enterprise data models (entity-relationship diagram), which diagram the data entities, attributes of the data entities, and relationships between the data entities.
- data flow, which diagrams the flow of data between processes.
- data definitions, which describe the data entities and attributes.

- physical data stores, which diagram the physical placement of data and data movement architecture (The Open Group, n.d.-a).

Application domain and architecture

The application architecture describes the use of business applications to serve the automation needs of business users. Common artifacts created to represent it include

- catalogs such as the Application Portfolio and Interface Catalog.
- matrices such as the Application/Function Matrix and Application/Organization Matrix.
- diagrams such as Use-Case Diagrams (The Open Group, n.d.-a).

Technical domain and architecture

The technical architecture describes the technologies necessary for the operation of the business applications, the storage and tools required for the acquisition and cleansing of data, and the dissemination of valuable information. Common artifacts created to describe it include

- catalogs, such as Technology Standards and Technology Portfolio.
- matrices, such as an Application/Technology Matrix.
- diagrams, such as Network Diagram and Environments and Locations Diagram (The Open Group, n.d.-b).

Embedding EAM

The investment in EAM anticipates incorporating the EAM practice and its products into the manner in which the organization conducts its business.

Strategic planning

Starting with strategic planning events, EAM's knowledge of new and emerging technologies can help formulate potential technology-enabled business improvements. EA documentation contributes to understanding existing technology and provides a context for enhancements (Ahlemann et al., 2012).

Following the ideation phase, EAM will participate in the overall roadmap development and structuring of the project portfolio related to the technology advancements. Knowledge of the current state and the desired future state enables EAM to define the gap and specify the projects that must be undertaken to bridge it. While doing so, EAM will seek to improve the architecture to reduce complexity and rigidity and encourage components that allow for agility (Ahlemann et al., 2012).

Next, EAM can guide the project's realization to ensure compliance with the desired future state and EA standards. Last, EAM should conduct an evaluation to ensure that the result is consistent with the strategic vision and roadmap activities (Ahlemann et al., 2012).

Project lifecycle

EAM's participation in the project lifecycle is often a topic of debate. The three general levels of project involvement are advisors, participants, or managers of the projects. The preference in most organizations is that EAM remains a strategic practice and that project involvement should be limited to an advisory role (Ahlemann et al., 2012).

As advisors, the EA team will typically assign an architect as a consultant to the project team and provide oversight of the project tasks and deliverables. The architect will ensure that the detailed domain-level design of the solution architecture is consistent with the target architecture. Further, they will verify that the solution utilizes the principles and standards that have been established for technology projects and that all EA artifacts have been modified to reflect the newest technological improvements. Last, should the project team need assistance with piloting or deployment of the new features, the architect can provide guidance (Ahlemann et al., 2012).

Operations

Operations is another IT function that elicits debate on EAM's involvement. While strategic changes are more significant, planned, and with longer horizons, operational changes are tactical, planned and unplanned, with short horizons. On occasion, operational changes are required because of services that no longer function. Given the urgency demanded by operational changes, efficient EAM practices are necessary. EAM interactions can be limited to those that require an impact to architecture or need an exception granted when the change cannot adhere to EA principles and standards (Ahlemann et al., 2012).

Monitoring

The EAM practice needs to monitor various aspects of their contributions to the business and IT so that they can make improvements where and when necessary. The key areas that should be monitored include the following:

- the impact EAM has on the business and IT. Are business goals being realized? Are technologies being optimized and costs contained? Are customer needs being met? Is IT learning and growing to better contribute to business growth? Are internal business processes improved?
- solution/project architectures. Are new projects in conformance with the future state architecture? Are principles and standards being followed?
- EAM adoption. Are the products of the EAM practice available and made available to the broader organization? Are artifacts being kept current? Are they providing value to the organization? (Ahlemann et al., 2012)



SUMMARY

IT EAM is a critical service to be implemented in organizations that value the contribution of technology to their success. IT EAM is a strategic practice that contributes to developing an organization's vision and strategy by identifying how it can leverage technology in its growth plans. Together with the business, a roadmap will be developed to achieve the future state architecture of technologies. Designing a future state architecture and creating a roadmap is a perpetual effort with EAM as a valuable contributor to the definition of organizational strategies. The future state architecture will include detailed architectures for the data, applications, and technical domains to support the business domain.

EAM will continue participation after project identification by advising business and IT during the project lifecycle and as needed once the solution has been implemented into production. Last, EAM must ensure their value to the business and IT by monitoring and governing the currency and use of the EAM assets and standards and contributing to the growth and learning experiences of the IT personnel.

UNIT 8

ARCHITECTURE GOVERNANCE

STUDY GOALS

On completion of this unit, you will be able to ...

- identify the value of architecture governance.
- define the key organizational roles for the successful implementation of architecture governance.
- discuss the principles, standards, and reference architectures that create boundaries for architecture.
- describe how project management and architecture governance mutually support their charters.

8. ARCHITECTURE GOVERNANCE

Case Study

ABC Company has initiated an enterprise architecture (EA) practice. The EA team has gathered portfolios of technologies and designed a current state architecture. They will be creating a future state architecture and roadmap to achieve the initial phases of the architecture.

ABC's IT organization has been in place for many years. The IT staff conducts development efforts with minimal use of standards, patterns, frameworks, and other best practices. ABC's IT leadership understands that implementing new disciplines will not achieve the anticipated benefits of cost reductions, rationalized technologies, and more, without changes in behavior and governance to promote compliance. IT leadership also wants more than compliance. They want to drive a collaborative culture between IT, EA, and the business. Both IT and business leaders believe that improvements and progress will be achieved with a collaborative culture.

8.1 Organizational Structure

Governance is the discipline implemented at organizational levels to ensure that the activities at that level are conducted in the manner in which they were planned. Governance has the following characteristics, as described by Hardasani (2021):

- discipline. Stakeholders commit to and abide by the established processes.
- transparency. Activities can be observed and reviewed by organizational leadership.
- independence. Processes and decision-making is based on established principles that minimize conflict of interest.
- accountability. Governance boards are authorized to make decisions and are held accountable.
- responsibility. Each party is required to be responsible.
- fairness. All decisions need to be equitable and not create an unfair advantage for any one party or stakeholder.

Other Governance Forms

To position architecture governance, it will be described in the context of other forms of corporate governance.

- Corporate governance oversees an organization's adherence to its established policies and practices.
- IT governance ensures that technological services to the organization are aligned with business needs and provide the level of technology capabilities necessary (Hardasani, 2021).

Architecture Governance

Architecture governance is a segment of IT governance, defined by The Open Group (n.d.-b) as “a series of processes, a cultural orientation, and set of owned responsibilities that ensure the integrity and effectiveness of the organization’s architectures” (Architecture Governance Framework section).

Architecture governance frameworks

Frameworks provide a structured guide for organizations to implement a discipline or practice, ensuring that the right processes and resources are in place to be successful. The Open Group Architecture Framework (TOGAF), published by The Open Group, includes an Architecture Governance Framework (AGF) that organizations can use to provide direction to the architecture governance discipline. The main components of the AGF provide an identification of the processes of architecture governance, the related artifacts, and the creation of an artifact **repository** (The Open Group, n.d.-b).

Control Objectives for Information and Related Technologies (COBIT) is the leading governance framework. COBIT emphasizes regulatory compliance to help achieve improved IT and business alignment, assisting organizations in realizing value from their technology investments. One of the many processes in COBIT is “Manage Enterprise Architecture.” COBIT and TOGAF work together, with COBIT guiding the creation and maintenance of the architecture governance process, while TOGAF provides the specifics for the creation and maintenance of an enterprise architecture (Reciprocity, 2019).

Repository

This is the categorization and storage of digital documentation.

Benefits of architecture governance

Governing the organization’s technology architectures provides many benefits, ensuring that the architecture

- links technology resources, processes, and information to organizational strategies;
- is sound and consistent;
- supports the standardization and best usage of technology assets;
- supports the best use of data and information, and skilled resources;
- protects the organization’s digital assets; and
- supports regulatory and other best practices (Behara, n.d.; Hardasani, 2021; The Open Group, n.d.-b).

Architecture governance clarifies organizational roles and responsibilities to deliver the identified benefits and defines a process for exception handling (Behara, n.d.; Hardasani, 2021; The Open Group, n.d.-b).

Governance Roles and Responsibilities

Architectural governance can be implemented with one or more entities. Entities typically include the Architecture Steering Committee and Architecture Review Board. An Architecture Forum is an optional committee. The roles and responsibilities will be unique to each organization and the policies and procedures that guide the activities of the organizational entities.

Architecture Steering Committee

The Architecture Steering Committee provides governance to the enterprise architecture (EA) discipline. The Steering Committee approves architecture standards, principles, and **reference architectures**. The Steering Committee also reviews and approves enterprise architecture plans and roadmaps (Behara, n.d.).

Reference architectures

This is a template for the development of an architecture, typically specific to an industry.

Architectural Review Board

The Architecture Review Board (ARB) has two key responsibilities. First, the ARB will establish architectural standards, EA principles, and reference architectures as components of the framework to be used for the development of solution architectures by project teams. Second, the ARB reviews and approves project solution architectures to ensure that each solution architecture complies with the established framework (Behara, n.d.). Moreover, the ARB typically includes six to twelve members, including architects, IT manager-level resources, and business manager-level resources, representing the technology domains (Ahlemann et al., 2012).

Architecture Forum

An organization can establish an Architecture Forum (AF) to encourage collaboration between the business and IT organizational representatives, industry peers, and technology vendor experts to share technical expertise and innovation. The AF is a voluntary, rather than a formal role. An AF might be established during periods of significant organizational growth or technology shifts. Also, an AF might be established in the early stages of EAM to provide unbiased input to the creation of the EAM practice (Ahlemann et al., 2012).

8.2 Policy Development and Enforcement

The purpose of governance is to enforce compliance of architectures with established policies. Policy development is the creation of the guidelines of expected behavior, while policy compliance ensures compliance of organizational actions with the established policy (Online-PMO, n.d.). Specifically, architecture governance includes creating and maintaining architectural artifacts that describe the current state (Wijegunaratne et al., 2014). This includes domain and enterprise-level artifacts, such as the key ones identified below:

- future state enterprise architecture
- business capability map (business)

- entity-relationship diagram (data)
- application portfolio (applications)
- infrastructure portfolio (technical)

Architecture governance includes oversight to ensure the implementation of technology solutions in conformance with the architectural artifacts and roadmap. In addition, governance also needs to enforce compliance of architectures with established policies. “Policy development” is the creation of the guidelines of expected behavior, while “policy compliance” ensures that organizational actions follow the established policy (Online-PMO, n.d.). To this end, the ARB is the organizational unit chartered with the responsibility of governance, assessing, and enforcing compliance. This section will present the policies commonly used by the ARB for architectural assessment (Jimenez, 2020).

Principles

EA principles are established early in the formation of the EA practice. These are rules and guidelines created by consensus, for the long term, that guide how technical decisions are made for the creation, maintenance, and use of architectures (The Open Group, 2018, Chapter 20).

EA principles include the following components:

- a name, an easy-to-remember identifier given to the principle
- a statement that describes the principle clearly and concisely
- a rationale that identifies the reasons for applying the principle to the organization
- implications, which highlight the requirements and impact of the principle

EA principles are typically created and categorized by domain. Therefore, principles will be created specific to the business, data, applications, and technical domains. An example of a fully formed principle is shown below, followed by the names of common principles in each domain.

Principle example

TOGAF (The Open Group, 2018, Chapter 20) lists 21 principles used in whole or in part by organizations. The wording of principles selected by an organization should be modified to reflect their intentions and culture. Principle 20: Control Technical Diversity below exemplifies a well-constructed principle.

Table 10: Control Technical Diversity

Name	Principle 20: Control Technical Diversity
------	---

Rationale	<ul style="list-style-type: none"> • There is a real, non-trivial cost of infrastructure required to support alternative technologies for processing environments. There are further infrastructure costs incurred to keep multiple processor constructs interconnected and maintained. • Limiting the number of supported components will simplify maintainability and reduce costs. • The business advantages of minimum technical diversity include standard packaging of components; predictable implementation impact; predictable valuations and returns; redefined testing; utility status; and increased flexibility to accommodate technological advancements. Common technology across the enterprise brings the benefits of economies of scale to the enterprise. • Technical administration and support costs are better controlled when limited resources can focus on this shared set of technology.
Implications	<ul style="list-style-type: none"> • Policies, standards, and procedures that govern the acquisition of technology must be tied directly to this principle. • Technology choices will be constrained by the choices available within the technology blueprint. • Procedures for augmenting the acceptable technology set to meet evolving requirements will have to be developed and put in place. • The technology baseline is not being frozen. • Technology advances are welcomed and will change the technology blueprint when compatibility with the current infrastructure, improvement in operational efficiency, or a required capability has been demonstrated.

Source: The Open Group (2018, Chapter 20).

The following chart identifies principles commonly used by organizations.

Table 11: Common Principles

Domain	Sample principle 1
Business	Compliance with law Maximize benefit to the enterprise Business continuity
Data	Data are shared. Data are an asset. Data are accessible. Data trustee Common vocabulary and data definition Data security
Applications	Common-use applications Technology independence Ease of use
Technical	Cloud-first Requirements-based change Interoperability

Source: Joan Lawson (2022), based on The Open Group (2018, Chapter 20).

Standards

Technology standards describe specific technology usage, identifying which technologies and technology approaches are standard or acceptable for use and which are not. Typically, standards are specified for three categories:

1. Solutions. These standards specify hardware and software products that can be used and how they should be acquired.
2. Configuration. These standards specify how hardware and software can be installed, configured, and made available to the end-user.
3. Utilization. These standards specify how software applies to an identified business need (ITtoolkit, n.d.).

Standardization limits the variety of products in use by IT departments. This allows IT to focus on the defined standards, developing product expertise, reducing compatibility and integration concerns; facilitating **disaster recovery planning**, licensing, and security management; and reducing acquisition costs as a result of contractual advantages (ITtoolkit, n.d.).

Disaster recovery planning

This is a formal plan for responding to unplanned and disruptive incidents, such as cyberattacks.

Reference Architectures and Architecture Patterns

Architecture patterns are generic technical approaches used in the design of an architecture. Architecture patterns describe how components behave and communicate, the underlying technical layers, and the tools used to create the components (Singh, 2019).

Reference architectures are generic architectures that can be used to construct, standardize, and evolve the architecture of systems for a specific domain, such as an industry. A reference architecture will specify the components of the architecture and how the components address business requirements (Guerra & Nakagawa, 2008). Reference architectures will include a common business vocabulary, reusable designs, industry best practices, architecture principles, building blocks (architecture patterns), and standards (LeanIX, n.d.-b). Reference architectures are often used to evolve a future state architecture for the organization. LeanIX (n.d.-b) lists the following common reference architectures:

- eTOM (telecommunications industry)
- ISA-95 and SCOR (manufacturing and supply chain)
- BIAN (banking industry)
- ACORD (insurance industry)
- US FEAF, Australian AGA (government)
- NAF, DoDAF, MoDAF (defense industry)

Architectural Review Board

The Architectural Review Board (ARB), as previously described, is the organization chartered with the implementation of policies and compliance. The ARB will define and execute the process for determining compliance and resolving exceptions. To support the success of the efforts, the ARB will also provide guidance and communications that encourage the use of the established policies (Jimenez, 2020).

8.3 Project Support

Often governance is perceived as a hurdle to progress. Architecture governance should be perceived as support to project teams, enabling the success of the project and the solution that is being developed. Leganza (2003) states clearly, “the key to enterprise architecture (EA) effectiveness is governance, and the key to governance is intervention in IT projects” (para. 1). When project governance and architecture governance operate together, project governance will enforce architecture compliance, and architecture governance enables the project’s success and the ability to achieve business goals, control costs, maximize resources, and provide full IT value (Leganza, 2003).

EA as Consultants

The review of solution architectures can occur at various stages of the project. If resource availability allows, it is recommended that architects from the EA team be assigned to project teams to function in a consultative capacity. By participating in this manner, consultants can engage early, review requirements, participate in the solution design and technical component selection, and identify opportunities to reuse existing technical building blocks, standards, and reference architectures (Leganza, 2003). Consultants best serve an organization when they participate in an advisory manner rather than with an autocratic approach.

Review Process

Reviews of approved projects should be conducted by the ARB in conjunction with appropriate project team members at a few points in the project cycle. These points can include, as needed, the initiation of the project, complete design review after initial architecture design, and major design changes, when developed and tested, and prior to implementation (The Open Group, n.d.-b).

The ARB will define a process by which one or more design reviews will be conducted on approved projects. Most common is an initial review, a full design review when the design is complete, and a final review when development and testing are complete and before deployment into production. The full design review is the most critical, where a full inspection is conducted on all design elements (Jimenez, 2020).

The Open Group (n.d.-b) suggests that the ARB’s review should be conducted with the following intentions:

- Identify any errors in the solution architecture so that they can be remedied earlier rather than later in the project, when change is costly and has a more significant impact on the project delivery.
- Encourage and ensure the use of standards, architecture patterns, and other best practices.
- Communicate technical readiness of the project to leadership.

Project participation enables the architects to learn from the project teams. Often, architects are perceived as being distanced from the reality of the enterprise's technical landscape, resulting in a lack of knowledge of the daily IT challenges. Architects have the opportunity to learn (The Open Group, n.d.-b)

- where standards and patterns need modification.
- services specific to a limited number of applications that may be used more broadly.
- opportunities to encourage collaboration, resource sharing, and synergies with other business and IT groups.
- to identify criteria to be used in future software and hardware procurement efforts.
- to identify architectural gaps to be resolved in a future state architecture

Review process

When a project complies with established policies, the project will be approved to proceed. When a project is not in compliance, the non-compliant architectural aspects will be detailed. Either the project will be returned to the design phase so that the architecture can be reworked, or the team will request that the ARB consider exceptions to one or more of the areas of non-compliance.

The ARB needs to consider the organization's desire to balance flexibility and control. This balance begins with the development of standards. The typical advice is that principles and standards are universal and allow for interpretation and determination of applicability. Both can be refined as the organization matures and a need for greater specificity or additional policies arises (Ahlemann et al., 2012).



SUMMARY

Many organizations have governance practices in place that govern their enterprise-wide activities. As a subset of corporate governance, IT organizations often have IT governance practices to oversee IT services delivery to the enterprise. One of the many ITIL Practices is architecture management. Architecture management is delivered via an IT EA practice in the organization.

Encouraging adherence to the valuable contributions of IT EA practice requires a shift in the culture of IT organization. Collaboration between the business, EA, and IT needs to be encouraged and rewarded. Further, governance mechanisms need to be established and administered to

realize the benefits of a planned technology landscape. Governance will additionally lead to collaboration between the groups so that the best policies can be implemented, and the necessary components created, improving productivity for all stakeholders.

BACKMATTER

LIST OF REFERENCES

- Ahlemann, F., Stettiner, E., Messerschmidt, M., & Legner, C. (Eds.). (2012). *Strategic enterprise architecture management: Challenges, best practices, and future developments*. Springer
- Axelerate. (n.d.). *Application portfolio management 101: Method and benefits*. <https://www.axelerate.com/post/application-portfolio-management-101-method-and-benefits>
- AXELOS Limited (2019). *ITIL Foundation: ITIL 4 Edition*. The Stationery Office Ltd.
- AXELOS Limited (2020). *ITIL® 4: Digital and IT Strategy*. The Stationery Office Ltd.
- AXELOS (2024). What is service management? Axelos Ltd. <https://www.axelos.com/certifications/itil-service-management/what-is-it-service-management/>
- Behara, G. (n.d.). *A definitive guide to enterprise architecture governance*. LeanIX. <https://www.leanix.net/en/wiki/ea/enterprise-architecture-governance>
- Betz, C. T. (2011). *Architecture and Patterns for IT Service Management, Resource Planning, and Governance* (2nd ed.). Morgan Kaufmann.
- Bodenstaff, L., & Quartel, D. (2016). Application portfolio management: Gaining control of large landscapes [White paper]. BiZZdesign. <https://go.bizzdesign.com/ApplicationPortfolio-Management>
- Business Architecture Guild. (2020). *A guide to the business architecture body of knowledge* (BIZBOK Guide). https://cdn.ymaws.com/www.businessarchitectureguild.org/resource/resmgr/bizbok_8_5/bizbok_v8.5_final_part1.pdf
- Cambridge University Press & Assessment (2024). Information Technology. In *Cambridge Dictionary*. https://dictionary.cambridge.org/dictionary/english-german/information-technology?q=Information+Technology+#google_vignette
- Capture. (n.d.). Lost track of your application landscape? [White paper] <https://ppm.capture.eu/site/down-link?id=637>
- Cockburn, A. (2000). Selecting a project's methodology. *IEEE Software*, 17 (4), 64-71. <https://ieeexplore.ieee.org/document/854070>
- Easton, S., Hales, M. D., Schuh, C., Strohmer, M. F. (2014). *Supplier Relationship Management*. Apress.
- Gruia, M. (2014). Is it time to renew your application portfolio? *ABA Banking Journal*, 106(10), 22-27.

- Guerra, E. & Nakagawa, E.Y. (2008). *Relating patterns and reference architectures*. Hillside. <https://www.hillside.net/plop/2015/papers/proceedings/papers/guerra-2.pdf>
- Hardasani, D. (2021, September 23). Architecture governance. *BPD Zenith*. <https://www.bpdzenith.com/architecture-governance/>
- Harmer, G. (2014). *Governance of Enterprise IT based on COBIT®5*. IT Governance Publishing.
- Jimenez, V. A. (2020, November 9). COBIT resource optimization mapped to TOGAF's architecture review board. *ISACA*. <https://www.isaca.org/resources/news-and-trends/industry-news/2020/cobit-resource-optimization-mapped-to-togafs-architecture-review-board>
- Joshi, S. (2020, September 30). How IT portfolio management can save your team money. *LearnHub*. <https://learn.g2.com/it-portfolio-management>
- Kaiser, A. K. (2023). *Reinventing ITIL and DevOps with Digital Transformation: Essential Guidance to Accelerate the Process*. Apress.
- Kharnagy (2016), *Devops Toolchain* [Image] CC BY-SA 4.0. Wikimedia. <upload.wikimedia.org/wikipedia/commons/0/05/Devops-toolchain.svg>
- Klosterboer, L. (2010). *Implementing ITIL Configuration Management*. IBM Press.
- KPMG (2017). *Driving business performance: Project management survey 2017*. KPMG. <https://assets.kpmg.com/content/dam/kpmg/nz/pdf/July/projectmanagementsurvey-kpmg-nz.pdf>
- LeanIX. (n.d.-a). The definitive guide to application portfolio management [White paper]. https://citadelgroup.com.au/wp-content/uploads/2020/09/LeanIX_whitepaper_Definitive_Guide_to_APM.pdf
- Leganza, G. (2003). *Project governance and enterprise architecture go hand in hand*. Giga Research. <https://www.dragon1.com/downloads/giga-governance-ea.pdf>
- Mosley, M., & Brackett, M. (2010). *The DAMA guide to the data management body of knowledge* (DAMA-DMBOK guide). Technics Publications.
- Mulder, J. (2023). *Modern Enterprise Architecture : Using DevSecOps and Cloud-Native in Large Enterprises*. Apress
- Online-PMO. (n.d.). Governance and organization. <https://online-pmo.com/purposedriven-pmo/governance-and-organization/>
- The Open Group. (2018). *The TOGAF standard: Version 9.2*. The Open Group Architecture Framework. <https://pubs.opengroup.org/architecture/togaf9-doc/arch/>

- The Open Group. (n.d.-a). *Architectural artifacts*. The Open Group Architecture Framework. <https://pubs.opengroup.org/architecture/togaf91-doc/arch/chap35.html>
- The Open Group. (n.d.-b). *Architecture governance*. The Open Group Architecture Framework. <https://pubs.opengroup.org/architecture/togaf8-doc/arch/chap26.html>
- O'Quinn, L. (2018, November 6). Reducing cost and complexity with application portfolio management. Plainview. <https://blog.planview.com/reducing-cost-and-complexity-with-application-portfolio-management>
- Pham, T., Pham, D. K., & Pham, A. T. (2013). *From business strategy to information technology roadmap: A practical guide for executives and board members*. CRC Press.
- Raza, M. (2021, October 5). Introduction to application portfolio management. BMC. <https://www.bmc.com/blogs/application-portfolio-management/>
- Reciprocity. (2019, December 10). What's the relationship between COBIT and TOGAF? <http://reciprocity.com/resources/whats-the-relationship-between-cobit-and-togaf/>
- Ruparelia, N. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35 (3), 8-13. https://www.researchgate.net/publication/220631422_Software_development_lifecycle_models
- Sansbury, J., Brewster, E., Lawes, A., Griffiths, R. (2016). *IT Service Management* (3rd ed.). BCS, The Chartered Institute for IT.
- Simon, D., Fischbach, K., & Schoder, D. (2010). Application portfolio management: An integrated framework and a software tool evaluation approach. *Communications of the Association for Information Systems* (26), 36-56. <https://aisel.aisnet.org/cais/vol26/iss1/3>
- Smallwood, R. F. (2014). *Information governance: Concepts, strategies, and best practices*. John Wiley & Sons.
- Solarwinds Pingdom. (2019, July 10). *Amazing facts and figures about the evolution of hard disk drives*. <https://www.pingdom.com/blog/amazing-facts-and-figures-about-the-evolution-of-hard-disk-drives>
- Sutherland, J., Schwaber, K. (2020). *The Scrum guide: The definitive guide to Scrum*. Scrum-guides. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Technopedia. (2017). *What is data architecture?* <https://www.techopedia.com/definition/6730/data-architecture>
- Thabit, I. (2011). *Architecture management body of knowledge: AMBOK guide for information technology*. Architecture Management Institute.

- Thinking Portfolio. (2013). IT portfolio management to support annual IT budgeting. <https://thinkingportfolio.com/it-portfolio-management-to-support-annual-it-budgeting/>
- Thejandra, B. S. (2014). *Practical IT service management* (2nd ed.). IT Governance Publishing.
- Turban, E., Volonino, L., & Wood, G. R. (2013). *Information technology for management: Advancing sustainable, profitable business growth* (9th ed.). Wiley.
- Ulrich, W., & Kuehn, W. (2015). Business architecture: Setting the record straight. In *Business and dynamic change: The arrival of business architecture* (pp. 21–36). https://www.omg.org/news/meetings/tc/ca-15/special-events/Business_Architecture_Setting_the_Record_Straight_Ulrich-Kuehn_07-2015.pdf
- Upadrista, V. (2015). *The art of consultative selling in IT: Taking blue ocean strategy a step ahead*. CRC Press.
- Wijegunaratne, I., Fernandez, G., & Evans-Greenwood, P. (2014). *Enterprise architecture for business success*. Bentham Books.
- Van Haren Publishing. (2018). Application portfolio management: An introduction. <https://www.vanharen.net/blog/application-portfolio-management-an-introduction/>

LIST OF TABLES AND FIGURES

Figure 1: The Evolution of IT Services Over Time	16
Figure 2: Main Domains of IT Architecture Management	20
Figure 3: ITIL Practices	24
Figure 4: Problem Management Process Flow	36
Table 1: Service Management Software Tools	39
Table 2: Example of an Allocation Report	48
Figure 5: Example of Asset Lifecycle	49
Figure 6: Phases of Risk Management	51
Figure 7: ITSM Asset Management Interaction with Procurement and Finance	53
Table 3: IT Asset Traceability Matrix	54
Table 4: IT Asset Traceability Matrix	55
Table 5: IT Asset Traceability Matrix	56
Figure 8: Supplier Categorization	62
Figure 9: Supplier Moves in Nine Supplier Relationship Models	73
Figure 10: Waterfall Model	80
Figure 11: Effects of Proximity on Effectiveness of Different Communication Methods ..	83
Figure 12: DevOps Lifecycle: Underlying Phases	85
Table 6: Overlap Between ITIL's Service Design, Service Operations Stages and DevOps' Monitoring Phase: Service Design	89
Table 7: Overlap Between ITIL's Service Design, Service Operations Stages and DevOps' Monitoring Phase: Service Operation	89

Table 8: APM Data Elements	97
Table 9: Application Analysis	102
Figure 13: IT Enterprise Architecture Components	107
Figure 14: Evolving Architecture	113
Table 10: Control Technical Diversity	121
Table 11: Common Principles	122



IU Internationale Hochschule GmbH
IU International University of Applied Sciences
Juri-Gagarin-Ring 152
D-99084 Erfurt



Mailing Address
Albert-Proeller-Straße 15-19
D-86675 Buchdorf



media@iu.org
www.iu.org



Help & Contacts (FAQ)
On myCampus you can always find answers
to questions concerning your studies.