ITIL®

ITIL®4: High-velocity IT

ITIL® OFFICIAL PUBLISHER

AXELOS GLOBAL BEST PRACTICE

Explore the entire ITIL® 4 Managing Professional Range

Browse the range now

- Various formats to suit your learning style
- Gain skills for the digital era
- Improve career prospects
- Maximize exam success

tiny.cc/itil4official ITIL Official Publisher @ITILPublisher

ITIL®

ITIL®4: High-velocity IT

AXELOS
GLOBAL BEST PRACTICE

tso
a Williams Lea company

Published by TSO (The Stationery Office), part of Williams Lea,
and available from:

Online
www.tsoshop.co.uk

Mail, Telephone, Fax & E-mail
TSO
PO Box 29, Norwich, NR3 1GN
Telephone orders/General enquiries: 0333 202 5070
Fax orders: 0333 202 5080
E-mail: customer.services@tso.co.uk
Textphone 0333 202 5077

TSO@Blackwell and other Accredited Agents

AXELOS
Full details on how to contact AXELOS can be found at:
https://www.axelos.com
For further information on qualifications and training accreditation, please visit:
https://www.axelos.com/certifications
https://www.axelos.com/archived-pages/becoming-an-axelos-partner/training-organization-and-trainer-accreditation
For all other enquiries, please email: ask@axelos.com

First edition 2020

ISBN 9780113316403

Printed in the United Kingdom for The Stationery Office
Material is FSC certified and produced using ECF pulp, sourced from fully sustainable forests.

P003017691 c20 02/20

Contents

List of figures

List of tables

Foreword

At this new stage in the development of the IT industry, AXELOS is delighted to present ITIL 4, the latest step in the evolution of IT best practice. By building on our experience and bringing fresh and forward-looking thinking to the marketplace, ITIL 4 equips your business to deal with the challenges currently faced by the industry.

The adoption of ITIL as the most widely used guidance in the world on IT and service management will continue with ITIL 4. It ensures continuity with existing ways of working (where service management is already successful) by integrating modern and emerging practices with established and proven know-how. ITIL 4 also provides guidance on these new methods to help individuals and organizations to see their benefits and move towards using them with confidence, focus, and minimal disruption.

ITIL 4's holistic approach raises the profile of service management in organizations and industries, setting it within a more strategic context. Its focus tends to be on end-to-end product and service management, from demand to value.

ITIL 4 is the result of a great deal of global research and development work across the IT and service management industries; this work has involved active practitioners, trainers, consultants, vendors, technicians, and business customers. The architect team has collaborated with the wider stakeholders and users of ITIL to ensure that the content meets the modern requirements of continuity, innovation, flexibility, and value.

ITIL training provides individuals with a structured approach for developing their competencies in the current and future workplace. The accompanying guidance also helps organizations to take advantage of the new and upcoming technologies, succeed in making their digital transformations, and create value as needed for themselves and their customers.

As demand for digital technology grows, practitioners are under increasing pressure to design, develop, run, and support digital systems and services. *ITIL® 4: High-velocity IT* aims to help readers understand digital transformation and to guide them and their organizations towards a more integrated state between business and technology. By discussing organizational best practice and useful mental models from a practitioner's perspective, it provides invaluable guidance in the practical application of high-velocity IT.

Welcome to the new generation of IT best practice!

Mark Basham

CEO, AXELOS Global Best Practice

Preface

People depend on digital technology. The societal and economic impact of digitally enabled organizations is unprecedented, and these organizations demand a significantly different way of working. We need to strike a new balance between managerial control and professional judgement. IT service management often leans towards control, but digitally enabled organizations are based on complex and intrinsically unpredictable systems. These systems are uncontrollable at lower levels of granularity, so traditional management instruments such as specialization, prescriptive processes, and performance targets do not work. Action at the sharp end has to be based on professional judgement and an understanding of the nature of the systems in which the practitioner works. Practitioners therefore need appropriate 'high-velocity IT' ways of thinking and of working.

As the reader, you care about this not only because you want to remain valuable and relevant in a changing workplace, but also because you have the privilege of contributing to the prosperity and wellbeing of people by providing them with better digital experiences. You have taken the trouble to look at industry guidance such as ITIL, so you are presumably committed to your professional development. You are aware of the considerable challenges and responsibilities involved in managing digital technology. You are up for the challenge and are prepared to review and unlearn old approaches and integrate new concepts in your way of working. You take your work and yourself seriously.

We, the authors, reviewers, and other contributors, each in our own way, are committed to making our industry a better place to work. We believe that people come first. We work in complex human systems in which professionals often have to take gambles. This is defensible: there is limited data available so their actions are based on well-reasoned hypotheses that they calibrate when they observe the results of their actions. In toxic workplaces they could easily be blamed (with the luxury of hindsight) for taking the wrong decisions. This is unacceptable. When your people have to bet, you had better bet on your people. In parallel with the more 'technical' guidance that this publication offers, we draw your attention to the importance of psychological safety, wellbeing, and ethics in the workplace. Many organizations not only have accrued technical debt in their software as a result of taking conscious or unconscious shortcuts, they also have a considerable social debt that needs repayment. This is why we care. This is a pivotal moment. The time is right to share our thoughts and to hopefully be of service.

I have witnessed the concern, engagement, and generosity of those who have contributed to this publication. We believe in its potential. It is now up to you.

Mark Smalley

Lead editor, ITIL® 4: High-velocity IT

About the ITIL 4 publications

ITIL® 4: High-velocity IT addresses the specifics of digital transformation and helps organizations to evolve towards a convergence of business and technology, or to establish a new digital organization. It is one of four ITIL 4 publications, which build on the concepts introduced in *ITIL Foundation*. Each of these publications focuses on a different aspect of service management.

ITIL® 4: Create, Deliver and Support addresses the cultural and team management aspects of product and service management; provides an overview of the tools and technologies which support service management; and demonstrates how to integrate management practices into end-to-end value streams.

ITIL® 4: Direct, Plan and Improve helps to align product and service management with modern business requirements; drive successful organizational transformation; and embed continual improvement into an organization's behaviour at every level.

ITIL® 4: Drive Stakeholder Value provides guidance on establishing, maintaining, and developing effective service relationships at appropriate levels. It leads organizations on a service journey in their service provider and consumer roles, supporting effective interaction and communication.

The ITIL 4 publications are supported by the ITIL management practice guides, which contain pragmatic, hands-on guidance that can be applied in the context of all ITIL 4 publications. Practices that are particularly relevant to *ITIL® 4: High-velocity IT* include architecture management, availability management, business analysis, capacity and performance management, deployment management, information security management, infrastructure and platform management, monitoring and event management, portfolio management, problem management, relationship management, risk management, service continuity management, service design, service desk, service validation and testing, and software development and management. The practice guides can be accessed online at www.axelos.com/my-axelos/my-itil.

About the ITIL story

The guidance provided in this publication can be adopted and adapted for all types of organization and service. To show how the concepts of ITIL can be practically applied to an organization's activities, *ITIL® 4: High-velocity IT* follows the exploits of a fictional company on its ITIL journey.

This company, Axle Car Hire, is undergoing a transformation to modernize its services and improve its customer satisfaction and retention levels, and is using ITIL to do this. In each chapter of the text, the employees of Axle will describe how the company is improving its services, and explain how they are using ITIL best practice to do this.

ITIL storyline sections appear throughout the text, separated by a distinct border.

The story so far

Axle Car Hire is undergoing a digital transformation.

Axle is headquartered in Seattle, with branches across Europe, the US, and Asia-Pacific. Before its transformation, Axle faced a downturn in business and a decrease in customer satisfaction. It lost customers to disruptive enterprises offering innovative services, including car-sharing and driverless cars, through online platforms and mobile apps.

Consequently, Axle hired a new CIO, Henri, who was chosen for his experience in large-scale IT transformations, balancing approaches such as design thinking, DevOps, and Agile with management frameworks such as ITIL, ISO, COBIT, and IT4IT. He understands the importance of embracing IT and digital innovation in modern business. He was tasked with increasing customer satisfaction, attracting and retaining customers, and improving the company's bottom line.

Henri prioritized the digital transformation of Axle and used ITIL as a foundational source of best practice on which to build other approaches. This enabled the change he knew the business needed. The adoption and adaptation of ITIL helped Henri to deliver the high-quality services that co-created value for Axle and its customers. He examined the ways that Axle could manage the four dimensions of service management, adopt the service value chain, and utilize the seven ITIL guiding principles in the continual improvement of its services.

Under Henri's direction, new services were introduced, such as the advanced driver assistance system and biometric access to vehicles. These new services were widely adopted by Axle's customers. As a result, the company gained a reputation for fast and reliable service. Customer loyalty improved and repeat bookings increased. The Axle Green improvement initiative was also introduced to help Axle achieve its vision to be an environmentally friendly organization. Many of the company's environmentally friendly targets have already been achieved, with plans for new developments underway. A project to ensure that half the Axle fleet runs on sustainably generated electricity is making progress.

After a period of strong growth, Axle is experimenting with new service models in response to the changing business climate. In different locations around the world, Axle is looking for solutions to the new challenges it faces. If the new service models are successful, they can be deployed in Axle's branches worldwide.

Meet the Axle employees

Here are five key employees of Axle Car Hire:

Henri *Is the new CIO of Axle Car Hire. He is a successful business executive who's prepared to shake things up. He believes in an integrated approach to IT and service management.*

Marco *Is the Axle Car Hire IT delivery manager. He is process-driven and continually references the ITIL framework to help him manage positive service relationships. However, Marco has had little exposure to a blended or collaborative approach to service management.*

Radhika *Is the Axle Car Hire IT business analyst, and it is her job to understand the user requirements of Axle Car Hire staff and customers. She is inquisitive and energetic, and strives to maintain a positive relationship with all her customers, both internal and external. Radhika works mostly on discovery and planning activities, rather than in IT operations. She asks a lot of questions and is great at spotting patterns and trends.*

Solmaz *Is Axle's business transformation manager. She is passionate about customer satisfaction for existing and potential customers, and is focused on supplying appropriate services to meet their needs. To complement her role, she also specializes in human-centred design, making design decisions based on how people can, need, and want to perform tasks, rather than expecting users to adjust and accommodate their behaviours to a product. Solmaz is warm, collaborative, and likeable.*

Su *Is the Axle Car Hire product manager for travel experience, and has worked for Axle for the past five years. Su is smart, meticulous, and passionate about the environment.*

ITIL Foundation recap

This section provides a brief recap of the concepts introduced in *ITIL® Foundation: ITIL 4 Edition*.

The key components of the ITIL 4 framework are the ITIL service value system (SVS) and the four dimensions model.

The ITIL service value system

The ITIL SVS represents how the various components and activities of the organization work together to facilitate value creation through IT-enabled services. The structure of the ITIL SVS is shown in Figure 0.1.

The core components of the ITIL SVS are:

- the ITIL service value chain
- the ITIL practices
- the ITIL guiding principles
- governance
- continual improvement.

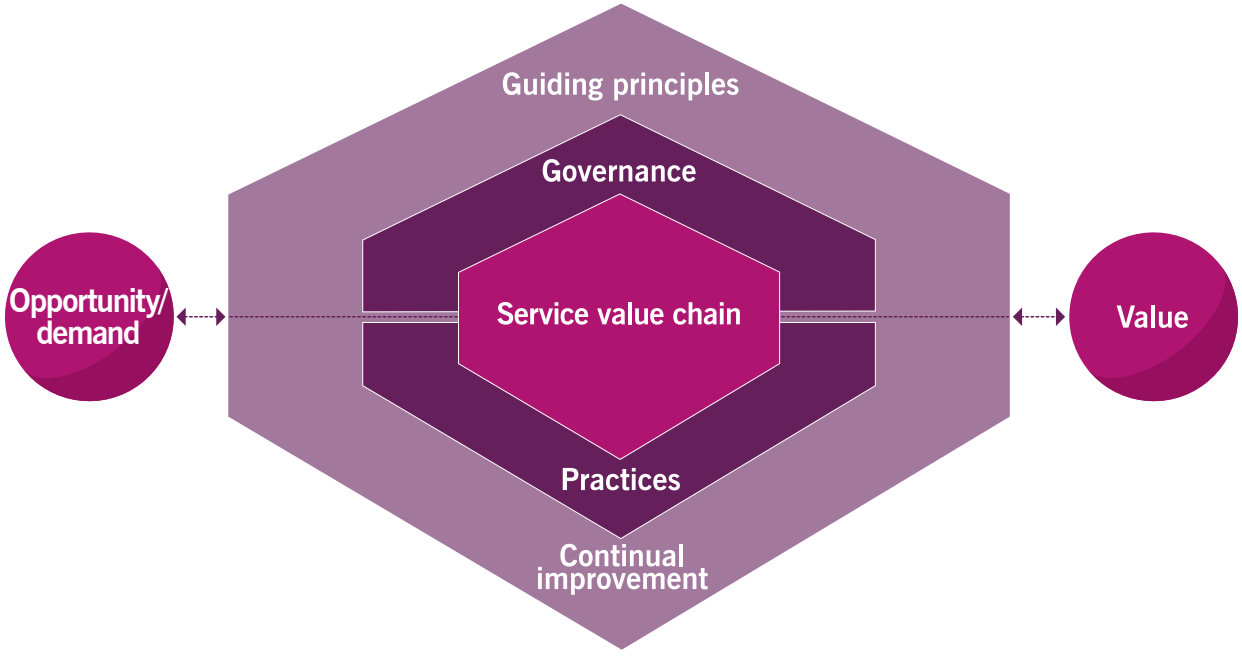

Figure 0.1 The service value system

The ITIL service value chain

The central element of the SVS is the service value chain, an operating model which outlines the key activities required to respond to demand and facilitate value realization through the creation and management of products and services. The service value chain is shown in Figure 0.2.

The ITIL service value chain includes six value chain activities which lead to the creation of products and services and, in turn, value. The activities are:

- plan
- improve
- engage
- design and transition
- obtain/build
- deliver and support.

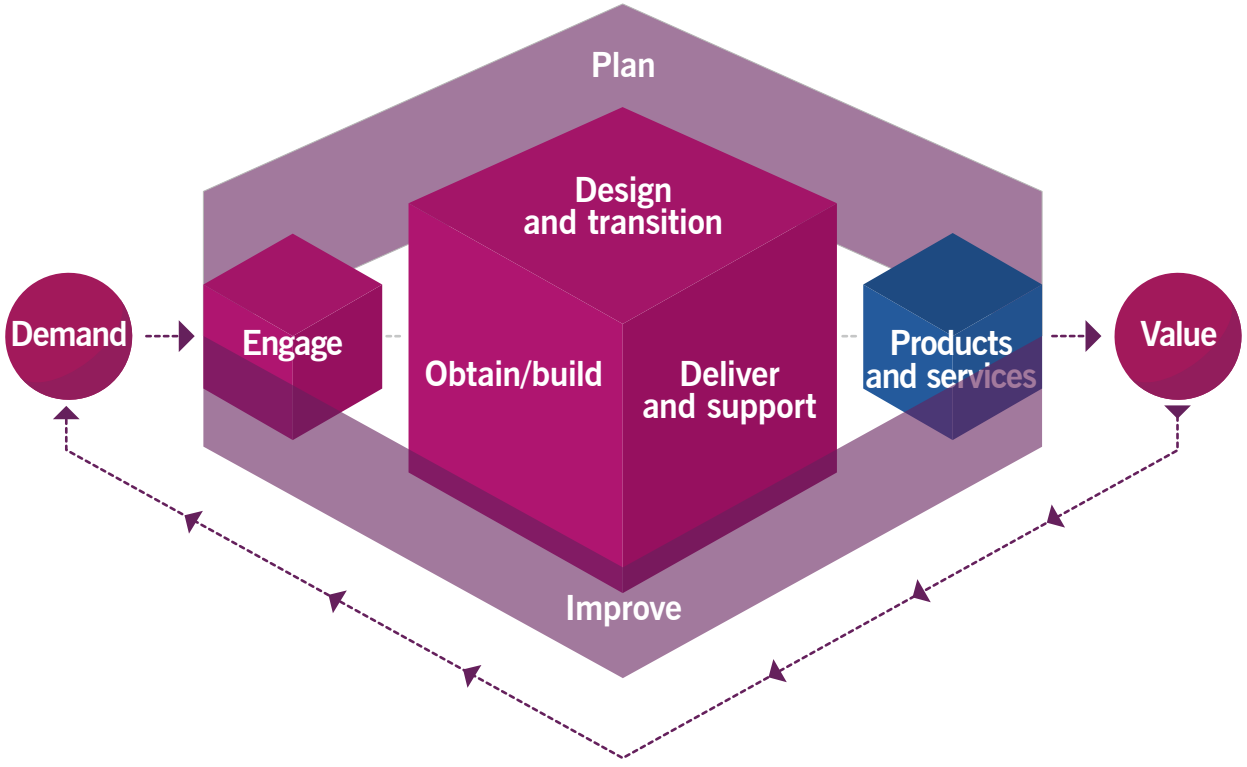

Figure 0.2 The ITIL service value chain

The ITIL practices

Practices are sets of organizational resources designed for performing work or accomplishing an objective. The ITIL SVS includes 14 general management practices, 17 service management practices, and three technical management practices. These are outlined in Table 0.1.

Table 0.1 The ITIL management practices

General management practices	Service management practices	Technical management practices
Architecture management	Availability management	Deployment management
Continual improvement	Business analysis	Infrastructure and platform management
Information security management	Capacity and performance management	Software development and management
Knowledge management	Change enablement	
Measurement and reporting	Incident management	
Organizational change management	IT asset management	
Portfolio management	Monitoring and event management	
Project management	Problem management	
Relationship management	Release management	
Risk management	Service catalogue management	
Service financial management	Service configuration management	
Strategy management	Service continuity management	
Supplier management	Service design	
Workforce and talent management	Service desk	
	Service level management	
	Service request management	
	Service validation and testing	

The ITIL guiding principles

The ITIL guiding principles are recommendations that can guide an organization in all circumstances, regardless of changes in its goals, strategies, type of work, or management structure.

The seven ITIL guiding principles are:

- **Focus on value** Everything that the organization does needs to map, directly or indirectly, to value for the stakeholders.
- **Start where you are** Do not start from scratch and build something new without considering what is already available to be leveraged.
- **Progress iteratively with feedback** Do not attempt to do everything at once.
- **Collaborate and promote visibility** Working together across boundaries produces results that have greater buy-in, more relevance to objectives, and increased likelihood of long-term success.
- **Think and work holistically** No service, or element used to provide a service, stands alone.
- **Keep it simple and practical** If a process, service, action, or metric fails to provide value or produce a useful outcome, eliminate it.
- **Optimize and automate** Resources of all types, particularly HR, should be used to their best effect.

Governance

Governance is the means by which an organization is directed and controlled. The role and position of governance in the ITIL SVS will vary depending on how the SVS is applied in an organization.

Continual improvement

Continual improvement is a recurring organizational activity performed at all levels to ensure that an organization's performance continually meets stakeholders' expectations. ITIL 4 supports continual improvement with the ITIL continual improvement model, outlined in Figure 0.3.

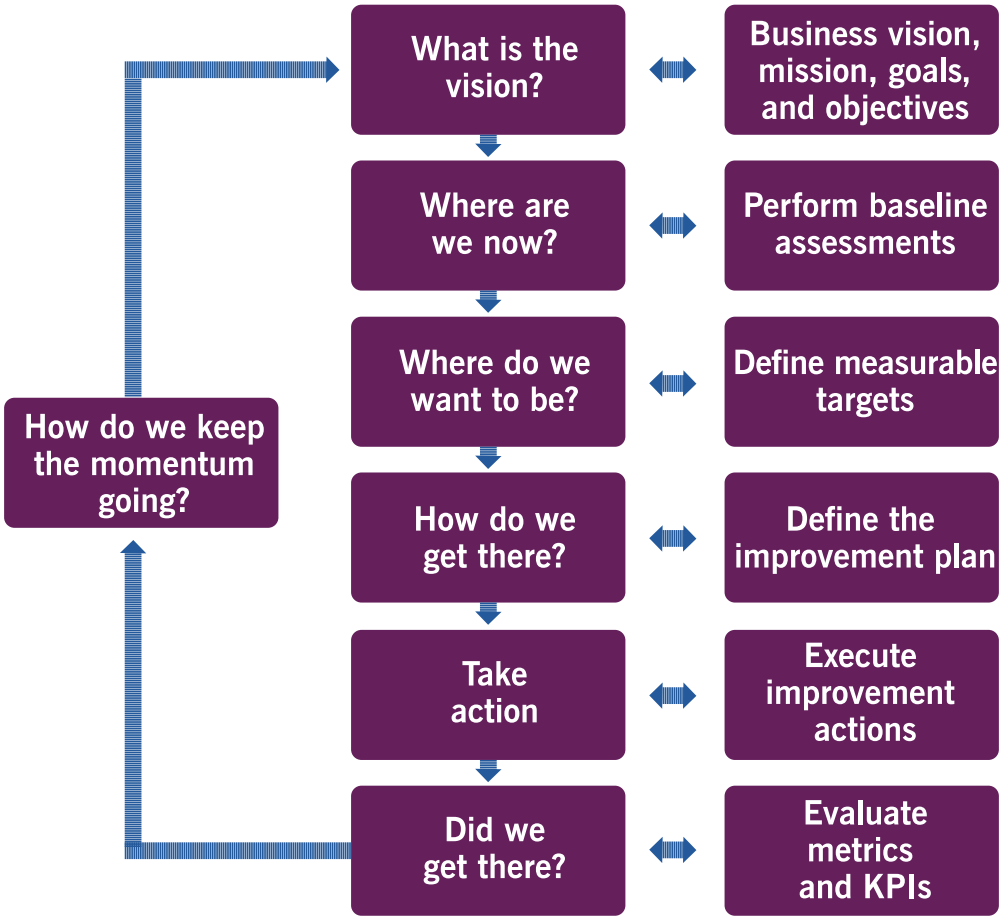

Figure 0.3 The continual improvement model

The four dimensions model

To support a holistic approach to service management, ITIL defines four dimensions that collectively are critical to the effective and efficient facilitation of value for customers and other stakeholders in the form of products and services. The four dimensions (shown in Figure 0.4) are:

- organizations and people
- information and technology
- partners and suppliers
- value streams and processes.

The four dimensions represent perspectives which are relevant to the whole SVS, including the entirety of the service value chain and all ITIL practices. The four dimensions are constrained or influenced by several external factors that are often beyond the control of the SVS.

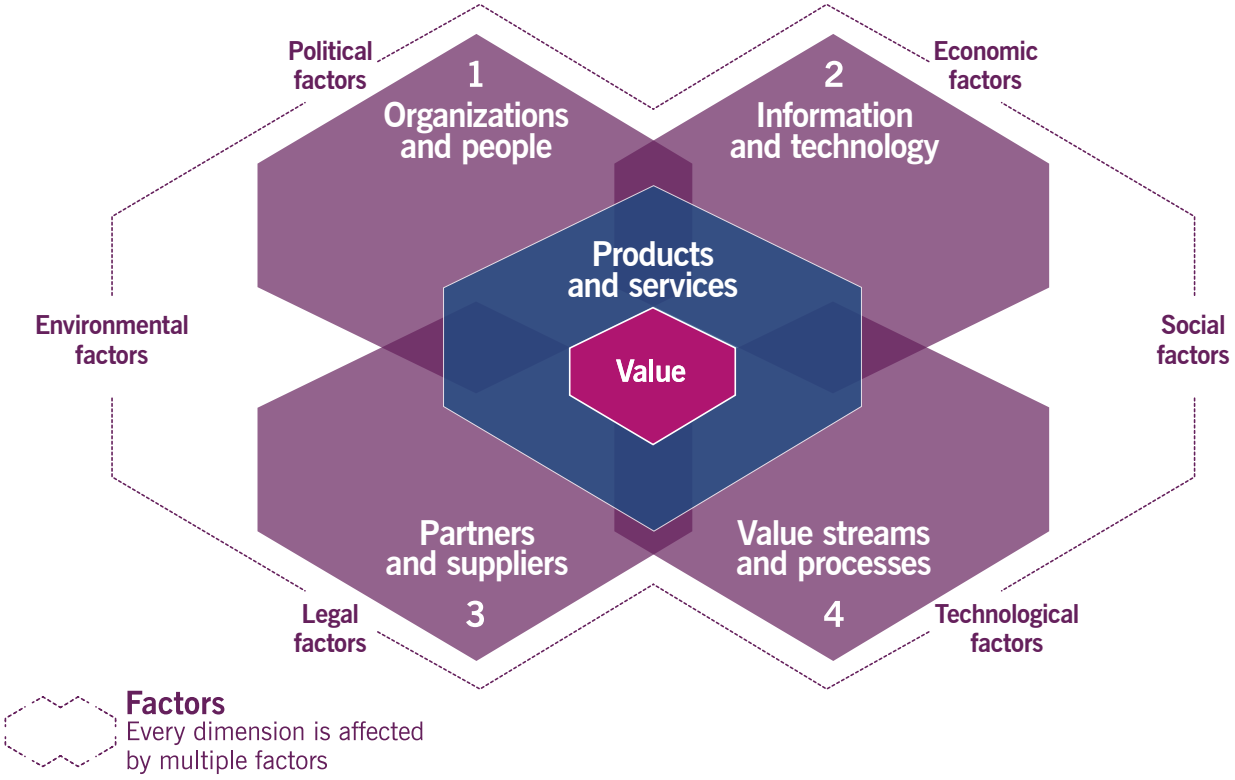

Figure 0.4 The four dimensions of service management

CHAPTER 1

INTRODUCTION

1 Introduction

Digital technology is increasingly important. Its economic, societal, and political impacts are unprecedented. At the same time, it is increasingly challenging for digital practitioners to design, develop, run, and support the systems and services that fulfil this demand. *ITIL® 4: High-velocity IT* focuses on digital products and services, including digital customer experiences, covering good organizational practices and mental models, all from a practitioner's perspective.

1.1 Audience and scope

This publication is aimed at IT and service management practitioners who work in organizations that are becoming more digitally enabled. It will help those who are familiar with traditional IT and service management concepts to discuss 'digital' confidently, develop practical competencies, and integrate new concepts, techniques, and technologies into their ways of working.

You should use this publication as a tool to improve how you and your co-workers:

- provide products and services
- continually raise your standards of work
- trust and are trusted
- accept ambiguity and uncertainty
- commit to continual learning.

This publication's scope includes the primary activities in the digital value chain, including what the practitioner does and the resources they use across the lifecycle of digital products to:

- make the right digital investments
- realize and deliver resilient digital products and services quickly
- ensure that the service consumer realizes value from those products and services
- assure the conformance of activities with governance, risk, and compliance requirements.

1.2 Background and context

Technological progress, combined with innovatively applied digital technology, has resulted in systems that are complex and unpredictable. Systems now comprise a multitude of components produced by parties who act with varying degrees of dependability, leading to intrinsic fallibility and flaws. Continual changes to a system and its environment mean that these flaws also continually change. Most are too small to cause significant issues, either because of redundancy and other forms of resilience in the system, or because of human intervention.

In the past, most people believed that change disrupts stability, and stability controls change; fewer changes meant less risk to stability. In the past decade, a different way of thinking has been adopted. By reducing the

size of change, the risk of disruption is reduced. Smaller, more frequent changes improve the organization's capability to change, thereby lowering the risk of disruption. Recently, the prevailing approach has been to embrace and enable changes.

All of this has changed the environment from 'causal' (if-then-else), predictable models to dispositional (if-then-maybe) ones. Because system behaviour cannot always or easily be predicted, ways of thinking and working shift: from prediction and control to insight and understanding; from large, occasional changes to small, frequent ones; from detailed planning in advance to continually experimenting and learning; and from failsafe to safe-to-fail. These are not trivial shifts.

In addition to this increase in complexity-based ways of thinking and working, breaking down silos, or at least silo-thinking, is a current concern. Homogeneous groups are less resilient and effective than diverse ones. In an organizational context, this means that specialists, teams, departments, and organizations should embrace opportunities to work with a wide range of people.

Fortunately, many things are changing. People are working effectively in small, product- or service-oriented teams. Automation is used more often to support IT processes. The interrelated concepts of immutable servers and infrastructure as code are widely adopted. Holistic systems thinking is more prevalent. Scientific thinking is being adopted. Unpredictability and ambiguity are increasingly accepted as necessary. There is a shift from a mentality of delivering value to one of co-creating value. External services are often acquired and integrated for fast development of digital capabilities, as either an alternative or an addition to the internal Agile approach.

These changes are manifestations of new ways of thinking and working that are becoming mainstream. *ITIL® 4: High-velocity IT* builds on these developments, and contributes to them. Among the topics discussed in this publication are:

- the ethical considerations relevant to digital technology
- design thinking
- working in complex environments
- the ITIL continual improvement model
- Lean culture
- safety culture
- Lean, Agile, resilient, and continuous techniques.

CHAPTER 2

KEY CONCEPTS OF HIGH-VELOCITY IT

2 Key concepts of high-velocity IT

This chapter presents some key concepts that relate to high-velocity IT (HVIT). It explores the nature of digitally enabled organizations and how a significantly different way of working is required, and illustrates how the core concepts of ITIL guidance can be used as building blocks for defining and organizing HVIT work.

The chapter defines several key concepts, including:

- high-velocity IT
- digital technology
- digital organizations
- digital transformation
- high-velocity IT objectives and key characteristics.

2.1 High-velocity IT

Definition: High-velocity IT

The application of digital technology for significant business enablement, where time to market, time to customer, time to change, and speed in general are crucial. High velocity is not restricted to fast development; it is required throughout the service value chain, from innovation at the start, through development and operations, to the actual realization of value.

Just as some digital organizations are more digital than others, the velocity in some organizations is higher than in others. However, an organization with a higher velocity is not necessarily better. The velocity at which an organization should operate depends on the nature of that particular organization, and in some cases a lower velocity may be more beneficial. It is also not necessary, or even recommended, that the whole of an organization's IT should be high velocity. For example, more dynamic customer-facing systems can be managed with an HVIT way of working, whereas back-office legacy systems are better handled in a more traditional way.

High velocity does not come at the expense of the utility or warranty of the solution, and high velocity equates with high performance in general. Increasing velocity within an organization will always involve costs and risks, particularly when there is a steep change rather than gradual improvement. There may be situations where risks are consciously taken in order to gain or retain competitive advantage. Further risks are often unconsciously taken due to a lack of understanding of the context, and dilution of warnings before information has reached decision-making levels. Risks may also be taken when decision-makers are influenced by unbalanced incentives and targets; for example, when decisions are made which are more likely to help meet short-term targets rather than more sustainable ones.

Scientifically speaking, velocity also has a directional component. In HVIT, this is interpreted as doing the right thing. In other words, not only should the requirements of the high-velocity approach be fulfilled, but the right decisions should be made regarding investment and sustainability.

HVIT provides many organizations with higher degrees of digital enablement, but it is not always a prudent investment. For some organizations, it does not make sense to undertake such a transformation, as they have other, higher priorities. Others may choose not to try to increase velocity because they think the amount of cultural change involved would be too difficult to achieve, or unlikely to generate an acceptable return on investment.

The ITIL story: High-velocity IT

Henri: *Axle's vision is to be the world's most recognized environmentally friendly car hire company. This means we embrace new and green technologies in the pursuit of our business goals. Adopting and using these technologies according to the ITIL guiding principles shows the board of directors that judicious investment in the right technologies, correctly deployed, is an effective way to grow our company.*

Marco: *The pace of modern technological evolution means that tools and solutions can quickly become outdated. Technology that was optimal five years ago is not necessarily the best way to optimize our services today. Our tools need ongoing review and the swift development and deployment of any necessary updates to ensure they are still effective.*

Radhika: *Our research shows that our website has become outdated. For example, it is not optimized for the latest smartphones. Our competitors' platforms can be more rapidly tailored to react to the demands of a market that is increasingly doing business on devices rather than computers. We need to improve our platform, or our customers will find an easier way to book their travel.*

Solmaz: *We are competing against transformational business models that utilize technology and address changing customer requirements in new and interesting ways. In response, Axle is asking several of our branches to experiment with innovative business models. In my role as business transformation manager, I am responsible for ensuring that the business has the agility to adapt to change.*

2.2 Digital technology

Definitions

- **Digital technology** Technology that digitizes something or processes digital data. Digital technology refers to information technology (IT) and the parts of operational technology (OT) that have been digitized.
- **Digitization** The process of transforming something (e.g. text, sound, or images) from analogue to digital form by expressing the information in binary digits.

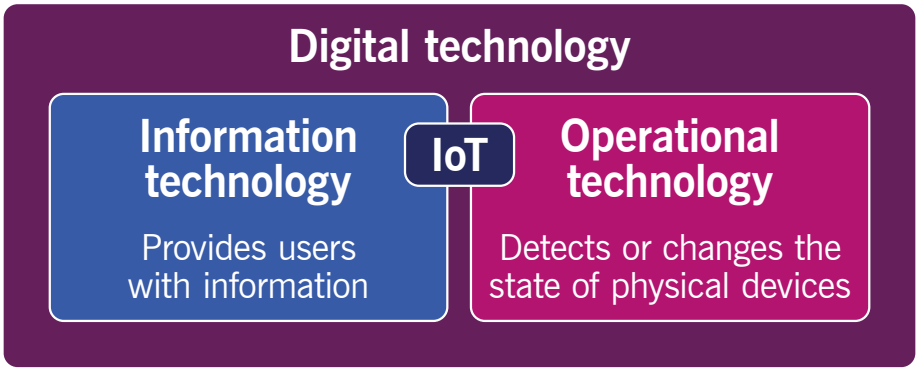

Figure 2.1 Digital technology

Digital technology is made up of both IT and OT. IT provides users with data and information, whereas OT detects or causes changes in physical devices (see Figure 2.1).

2.2.1 Information technology

IT exists as information systems that are made up of hardware, system software, data, and applications that are used for the purpose of data processing. Figure 2.2 shows an information system technology stack in more detail.

Information is data that is useful in a particular context. In IT, making information available to end users is the end goal. This can be presented in the form of numbers or text on a screen, or in other ways, for example, as a moving location on a map.

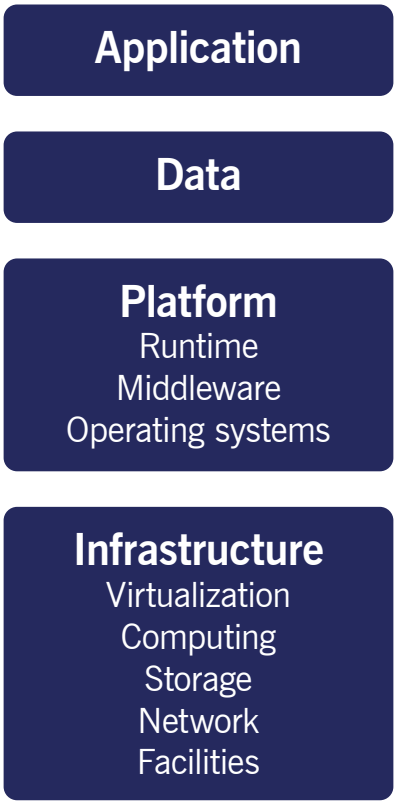

Figure 2.2 Information system technology stack

Definition: Information technology

The application of digital technology to store, retrieve, transmit, and manipulate data (data processing), often in the context of a business or other kind of organization.

The value of IT has traditionally been perceived as increased efficiency, because it leads to automated information systems that process and provide data more quickly, more reliably, and at a cheaper cost than can be done by humans. Increasingly, artificial intelligence (such as machine learning) is applying IT not only to process and provide data, but also to create new information.

Organizations mainly use the information gathered from traditional, automated information systems for internal decision-making. In this way, they help to reduce uncertainty, which is the primary function of information. The value of information is only realized when decisions are acted upon which have been informed and improved by that information. As such, a return on IT investments can often only truly be realized when decision-making takes into account the information that information systems provide. If no action is taken, no value is created. In some instances, this action could also be to do nothing, for instance, in order to avoid an identified risk.

Figure 2.3 demonstrates how the IT stack contributes to the creation of value through informed decision-making.

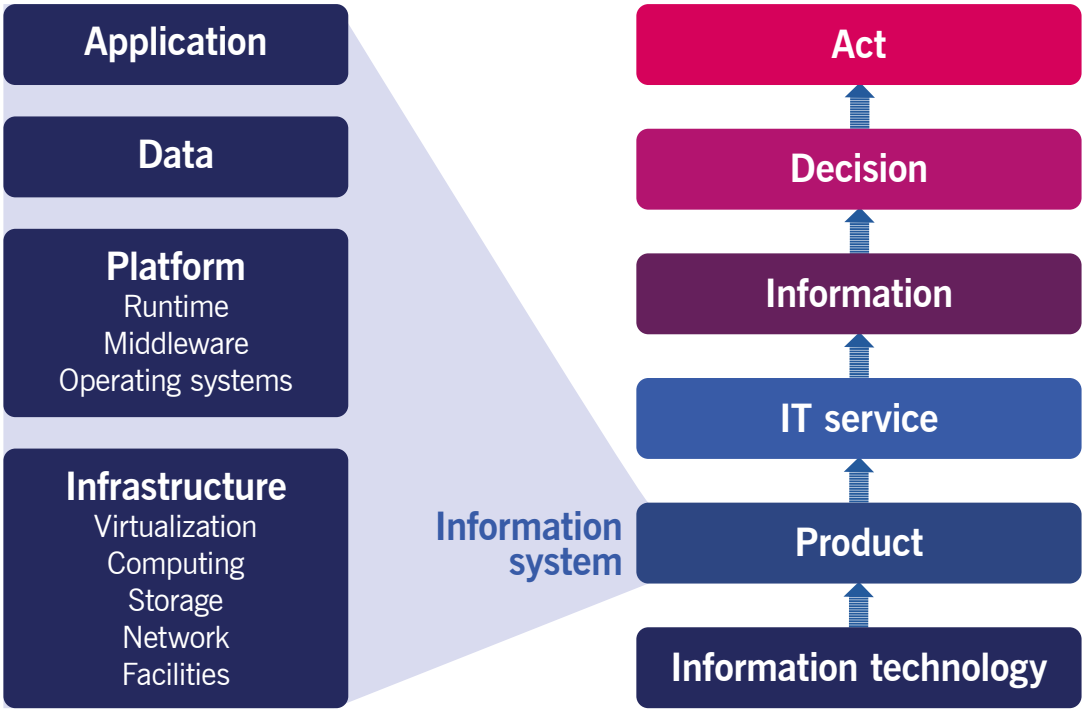

Figure 2.3 The IT value stack

Despite IT being such a core concept to organizations worldwide, the term is often misinterpreted. 'IT' can be used to refer to any of the following:

- the organizational IT function (the IT department); this publication refers to it as the 'IT function'
- IT infrastructure, including generic workplace productivity applications (such as word processing), but not the applications that support specific business functions

- an organization's internal information systems
- technical components used to create 'digital products'
- data processing technology (used for storing, retrieving, transmitting, and manipulating data)
- digital technology used to process data in order to digitize and automate business.

This publication uses the term 'IT' to refer to the application of digital technology to process data in order to digitize and automate business.

2.2.2 Operational technology

Definition: Operational technology

The application of digital technology for detecting or causing changes in physical devices through monitoring and/or control.

Operational technology (OT) differs from IT in that it uses digitized data as an internal means to a physical goal, rather than to make information available to users.

'OT' refers to physical devices (for instance, valves and pumps in machinery) in which digitized data is used to take physical action. OT devices can be as small as the engine control module (ECM) of a car or as large as the distributed control network for a national electricity grid.

The collective term 'industrial control systems' (ICSs) is used to refer to OT systems such as supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), remote terminal units (RTUs), and programmable logic controllers (PLCs), along with dedicated networks and organization units. The OT sphere also includes embedded systems (such as smart instrumentation) and a large subset of scientific data acquisition, control, and computing devices.

OT devices are supported by the Internet of Things (IoT), allowing them to connect both to each other and to information systems.

The ITIL story: Digital technology

Henri: *Axle Car Hire is leading the way in integrating our mechanical, or operational, technology with our information technology. Our vehicles are an extension of our digital platform; they are our entry point into the Internet of Things. The information gathered by the sensors and GPS installations in our cars is stored and shared so we can optimize and automate our service.*

2.3 Digital organizations

Digital organizations are enabled by digital technology. Digital technology is a significant underpinning enabler for these organizations' internal processes, and is often part of their products and services. As such, digital technology is a strategic part of a digital organization's business model, and is applied to its primary (rather than supporting) activities. Because of this, prioritizing digital technology ('digital first') is often part of such an organization's culture (in other words, the way things are done in that organization).

Definition: Digital organization

An organization that is enabled by digital technology to do business significantly differently, or to do significantly different business.

The exact definition of what 'being digital' means will typically vary between organizations, but is often reflected in the customer experience, the product or service, the business model, the operating model, and the employee experience that the organization provides. In a digital organization, these will mostly be enabled by digital technology.

Most organizations make some use of digital technology, and so, in a sense, are digital. In practice, however, an organization is regarded as 'digital' if it makes differentiating, and therefore strategic, use of digital technology. For instance, a traditional taxi company relies on digital technology for its operations; but for a taxi company that relies on an app for bookings, digital technology is strategic, and is a key part of its business model.

Although digitization across an organization usually manifests itself in its products and services, digitization of only internal practices may also be sufficient to qualify it as 'digital', as long as this results in significant benefit. Digital organizations are not wholly and consistently digitized, and often some parts will be more digitized than others. In these cases, this diversity will represent a challenge that must be carefully managed to make sure the different areas can work together effectively.

The parts of organizations that are significantly digitized place specific demands on the people responsible for digital enablement. These demands depend on the degree of digital enablement, but are generally higher than the demands for information systems for supporting activities. For digital products and customer experiences, innovative digital investments are needed to create or maintain competitive advantage, and have to be realized quickly. The resulting digital products, services, and customer channels have to be operationally resilient, and their users must use them well to achieve the desired return on investment.

The digitization of an organization has significant implications for its operating model (in other words, the resources it needs and how they interact). A major consideration for organizational operating models is the centralization or decentralization of the IT function, and how each of these options will affect the organization's effectiveness and efficiency. It is important that a digital organization's operating model is based on the co-creation of value by both the service provider and the service consumer to make sure that value from digital investments is properly realized.

For better and for worse, the societal, political, and economic impact of IT is unprecedented. There is, therefore, an increasingly strong moral obligation for digitally enabled organizations, in which IT empowers the business rather than just supports it, to consider how they apply digital technology beyond their direct economic interests.

The ITIL story: Digital organizations

Henri: *Our business is digitally enabled, but not always digitally integrated. For example, some branches lag behind in digitizing the bookings they take in person or over the phone.*

Radhika: *We have seen the emergence of fully automated car hire services, where every touchpoint and service interaction happens on a user-friendly, customizable app, and where the customer can even locate and unlock the car without a single human interaction.*

Solmaz: *Some customers prefer this, particularly when they are in countries where they do not speak the local language. We need to keep up with this demand so that we can give our customers the best possible experience.*

2.4 Digital transformation

Transformation is about doing things differently, or doing different things. It is also about reframing work to think about things differently, or think about different things.

'Digital transformation' is often used to indicate major investment in digitizing, robotizing, and other forms of automation that enable organizations to do business significantly differently, or do significantly different business. This technological change often requires organizational change in how the organization uses the digital solutions.

Definition: Digital transformation

The use of digital technology to enable a significant improvement in the realization of an organization's objectives that could not feasibly have been achieved by non-digital means.

The term 'digital transformation' is not specific to a particular type of transformation, and can be used to refer to any transformation that is digitally enabled. The transformed entity is often a combination of the organization's customer experience, products or services, business model, operating model (for example, the value stream), and employee experience.

Digitalization vs digitization

Digital transformation is sometimes referred to as 'digitalization'. However, the use of this term is not recommended because of the potential confusion with digitization, which is the technical process of changing something from analogue form to digital form.

The term 'transformation', used correctly, means major change. Despite this, transformation does not necessarily imply a single, large change. Based on the approach an organization selects, transformation can be achieved just as successfully with a few big changes, or many smaller ones. In many cases, a series of smaller changes can even be the more successful approach.

2.4.1 IT transformation

In organizations where business and IT are regarded as separate organizational functions, 'IT transformation' is often used to denote major change that improves how IT services are provided. IT transformation is focused on how IT services and information systems are developed, run, and supported. This can include decentralizing the IT function and integrating it into digital lines of business.

Before they undergo a digital transformation, organizations are managed separately from their IT service providers, whether internal or external. An IT service provider is focused on the management of IT resources to create and deliver IT products and services, whereas a service consumer is focused on the management of its products, services, and resources, including those delivered or supported by the IT service provider. Acting as a service consumer, this organization may influence the management of the service provider. This is shown as Model 1 in Figure 2.4.

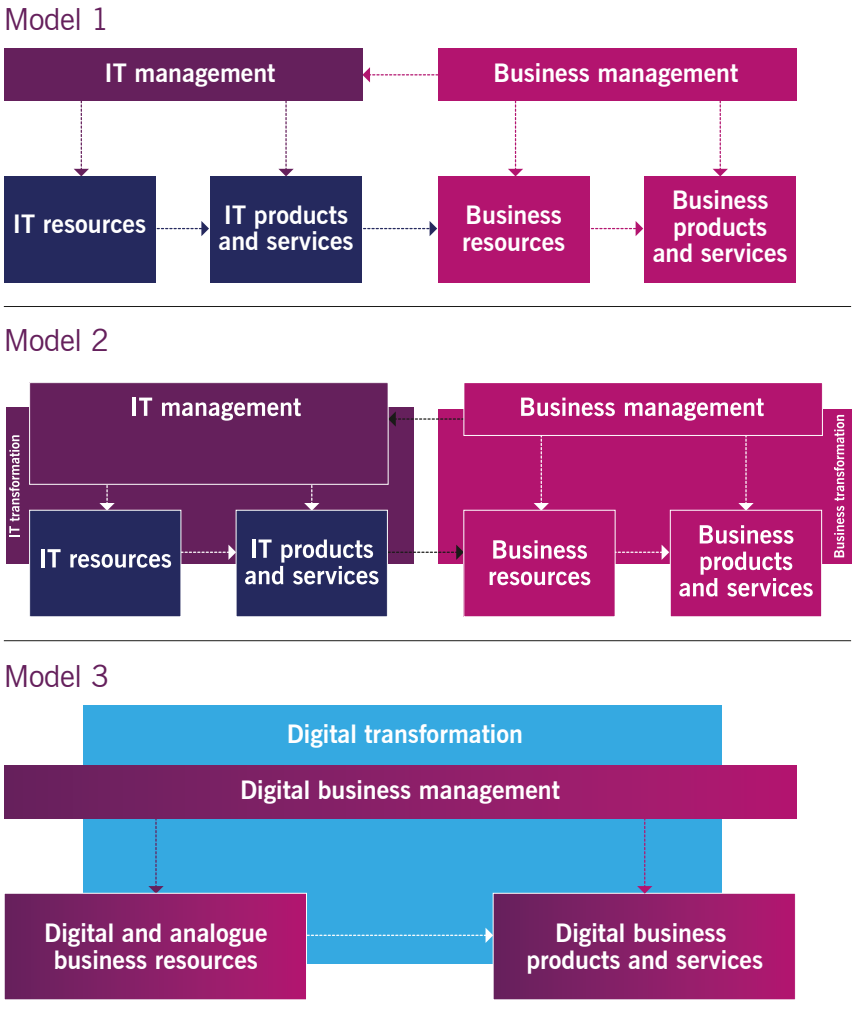

Figure 2.4 Digital transformation and IT transformation

The IT service provider and service consumer can both transform their management, resources, products, and services. These transformations can be interrelated, but they do not significantly change the way these organizations work together, or the role of IT in the service consumer organization. This is shown as Model 2 in Figure 2.4.

When the organizations undergo a digital transformation, the role of digital technology in the business of the service consumer significantly changes. This includes some or all of the following:

- digitization of the organization's products and services
- digitization of the organization's management practices
- digitization of a significant part of the organization's resources
- integration of IT management into business management; development of a partnership with the IT service provider or the merging of management practices.

This digital transformation is shown as Model 3 in Figure 2.4.

Where business and IT are not regarded as separate organizational functions, as is the case in most digital organizations, 'IT transformation' is not an appropriate term to use. 'Digital transformation' would be used instead.

Many of the approaches and techniques described in this publication are software-oriented, and may seem less relevant to IT operations practitioners. However, as physical platforms shift to code-driven technologies such as cloud, virtualization, containerization, and serverless infrastructures, it is essential for these practitioners to become competent in applying software engineering techniques. Increasingly, practitioners use scripts, code, and version control systems such as GitHub to make and manage changes, rather than manually implement them. It is therefore crucial for practitioners in HVIT environments to adopt a software engineering approach.

Digital products and services are based on software, which is another reason for IT operations practitioners to invest in more knowledge of software engineering. Software is such a primary business concern that many banks, for example, regard themselves as software companies with a banking licence. This increased importance has led to a shift in outsourcing policy where application development and management is often executed as an integral part of the product/service line or line of business that uses the digital technology. This is illustrated in Figure 2.5:

- In a decentralized IT function, each business unit has an integrated IT function and manages its IT services.
- An IT department acting as a centralized service provider serves multiple business units, and manages most of its IT services itself.
- An IT department acting as a centralized service integrator serves multiple business units, and combines its own IT services with those acquired from external service providers.
- When IT is integrated with business units, they are focused on the management of digital business services.

Practitioners who work in such decentralized IT functions that are an integral part of a line of business will have knowledge and competencies across more domains than those who work in specialized units. These domains include software engineering and the line of business itself. These practitioners no longer work with the concept of providing services to other departments: instead, they work with their co-workers to provide services for external customers.

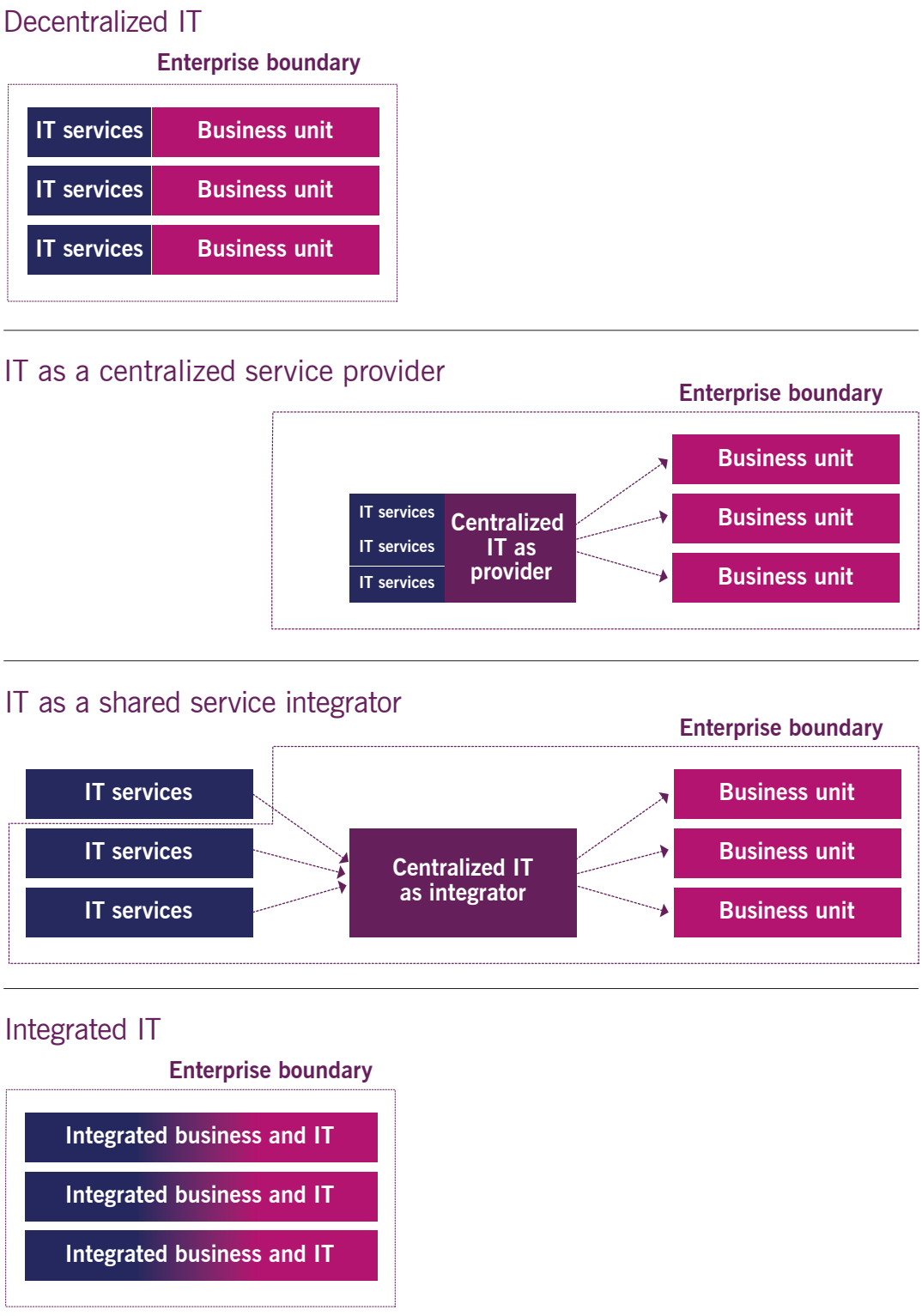

Figure 2.5 Examples of sourcing options for an IT function

The ITIL story: Digital transformation

Henri: *Our goal at Axle Car Hire is to introduce new services based on the evolving requirements of customers, and to continue to integrate technology to improve our services. When we adapt to environmental concerns, optimize the customer journey, increase service customization, and change our services, we make sure we are mirroring current consumer trends.*

Marco: *We recently transitioned our services from an on-premises infrastructure to a hybrid cloud solution. This was a major overhaul for our IT team. We had to update our scripts, codes, and version control systems. Now we can offer better cross-department functionality to our internal customers, and our business teams are not working in silos.*

Solmaz: *Our transition to the hybrid cloud solution was not a single, large change. For best change control purposes, we made sure we could make a series of smaller, targeted changes that could be thoroughly tested before deployment, and which could be continually reviewed and revised.*

2.5 High-velocity IT objectives and key characteristics

2.5.1 High-velocity IT objectives

Technology is strategic to digital organizations' business models, and as such, higher demands are placed on the lifecycle of their digital products. These demands can be represented by five high-level objectives that translate the vision and strategy of an organization into more operational objectives and indicators. These objectives are:

- valuable investments
- fast development
- resilient operations
- co-created value
- assured conformance.

It is important to remember that these objectives should not be managed in isolation; they influence each other and interact with each other, and need to be managed collectively. They are outlined in Table 2.1.

Table 2.1 HVIT objectives

Objective	Description	Closely related service value chain activities
Valuable investments	Strategically innovative and effective application of IT	Engage, plan, improve
Fast development	Quick realization and delivery of IT services and IT-related products	Engage, design and transition, obtain/build, improve
Resilient operations	Highly resilient IT services and IT-related products	Engage, deliver and support, improve
Co-created value	Effective interaction between service providers and service consumers	Engage, deliver and support, improve
Assured conformance	Adherence to governance, risk, and compliance (GRC) requirements	All value chain activities

As Table 2.1 shows, each objective is closely related to one or more ITIL service value chain activities, except for assured conformance, which relates to the governance component of the ITIL service value system, and applies to all activities. The relation of the objectives to the value chain activities is as follows:

- The valuable investments objective is mainly achieved by the decision-making that occurs as part of the plan value chain activity.
- The fast development objective is mainly achieved by the application development and infrastructure engineering that takes place as part of the design and transition and obtain/build value chain activities.
- The resilient operations objective is mainly achieved by running and maintaining the system as part of the deliver and support value chain activity.
- The co-created value objective is mainly achieved by supporting the system as part of the deliver and support value chain activity (together with effective use by the service consumer).
- The assured conformance objective is achieved by attention to compliance with corporate and regulatory directives as part of all of the value chain activities, not only during deliver and support.

Just as the assured conformance objective is supported by all value chain activities, the engage and improve value chain activities contribute to all HVIT objectives.

In some instances, there will be conflicts between the different objectives. For example, fast development could negatively affect resilient operations where insufficient time is given to ensuring that services and products are robust. For this reason, it is important to ensure that the objectives are properly balanced.

The valuable investments objective determines the potential value of an investment. Investments can be made in a variety of areas, but ultimately returns and benefits should be assessed against the other four objectives. As the organization progresses, it will incur additional costs, and might experience value leakage where the solutions and benefits are found to be sub-optimal. The actual realized benefit is the difference between the potential value of the investment, and the costs and value leakage. Return on investment can then be expressed as the realized benefit compared to the organizational investment. This is shown in Figure 2.6.

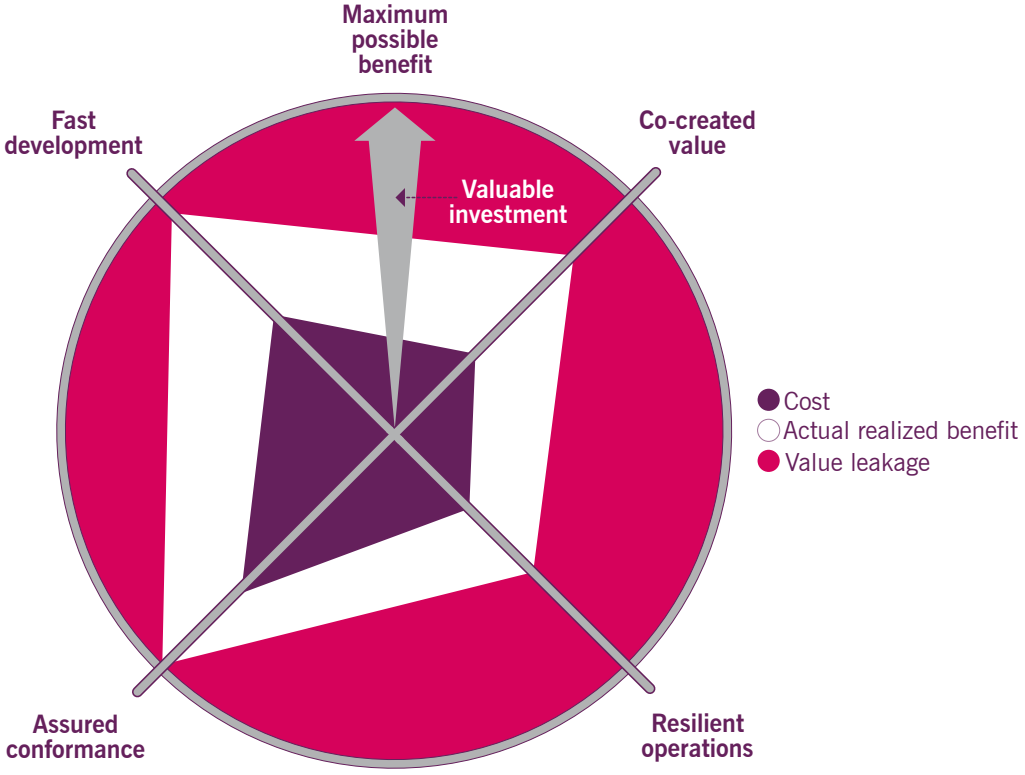

Figure 2.6 Objectives from an economic perspective

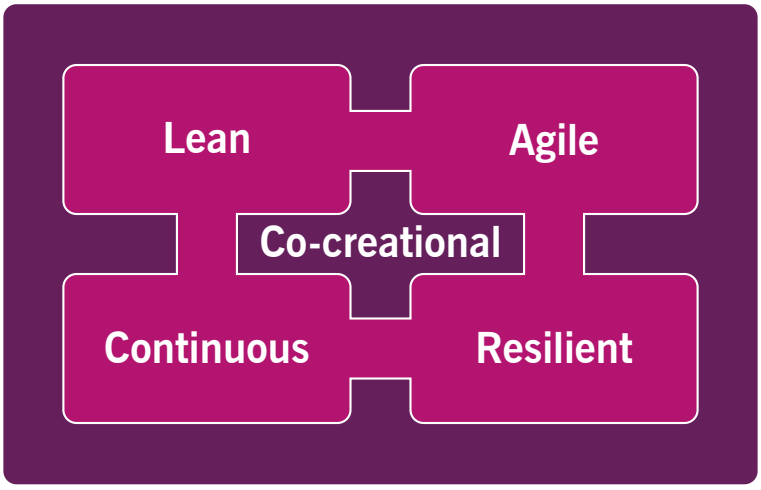

Figure 2.7 Key characteristics of HVIT

2.5.2 Key characteristics of high-velocity IT

There are many approaches that can be taken to reach and maintain HVIT. Four characteristics are dominant in common HVIT approaches. These are:

- Lean
- Agile
- resilient
- continuous.

When combined and used together properly by organizations, these characteristics enable the co-creation of value, as shown in Figure 2.7.

The benefits of each characteristic are outlined in Table 2.2.

Table 2.2 Key characteristics of high-velocity IT

Characteristic	Benefits
Lean	Helps to improve throughput and reduce waste. HVIT environments benefit from approaches with Lean characteristics due to the pressure on time to market and time to customer.
Agile	Adds close and iterative collaboration with users. Approaches with Agile characteristics are important for HVIT environments because digital products and services have to be developed in response to changeable market demands.
Resilient	Maintains workable availability and performance. The systems that support HVIT environments are complex and therefore error-prone. Approaches with resilient characteristics minimize the effect of incidents by degrading systems gradually and restoring service quickly.
Continuous	Ensures fast and reliable deployment. Approaches with continuous characteristics extend the Lean focus on throughput by standardizing and automating processes for integrating, building, testing, and shipping code, enabling digital products and services to be available when required.

When used together, these characteristics lead to the co-creation of value, extending the focus of an approach to effective service consumption. Value is only realized when the user actually uses the digital products and services. An approach that uses all four key characteristics of HVIT will help the service provider to ensure that the service consumer achieves the desired outcomes.

These characteristics are technical in nature, focusing more on the tangible parts of information systems (the product), but many of the principles can also be applied to IT services.

In isolation, these characteristics are not unique to HVIT. Together, however, they help to fulfil the higher demands that digitally enabled organizations place on IT. They contribute to:

- planning the right investments in digital products and services
- the quick and reliable development and deployment of these products and services
- keeping them operational
- ensuring that service consumers realize value by using them effectively.

Similar to the five HVIT objectives, each HVIT characteristic relates to several or all ITIL service value chain activities. This is outlined in Table 2.3.

Table 2.3 Summary of the influence of HVIT characteristics on ITIL service value chain activities

	Plan	Improve	Engage	Design and transition	Obtain/build	Deliver and support
Lean		✔		✔	✔	
Agile	✔		✔	✔	✔	
Resilient	✔			✔		✔
Continuous		✔	✔	✔	✔	✔

2.5.2.1 Lean

Approaches with Lean characteristics focus on breaking down large pieces of work into smaller batches. A short lead time is the best way to ensure quality, customer satisfaction, and employee happiness, and a good way to achieve short lead times is by using small batch sizes of work. It is therefore beneficial to break down larger pieces of work into smaller ones. This will often present a different challenge to work that is organized into larger batches that are passed between functions with formal handover procedures.

Small batch sizes help to reduce the disruptive effect of changes on product systems. The smaller the change, the lower the risk of disruption. Reducing the size of changes also means that they can be executed more frequently. A higher frequency of change improves an organization's ability to change, which also lowers the risk of disruption to operational systems. This in turn helps to reduce the organizational tension between the HVIT objectives of fast development and resilient operations.

Another Lean technique to improve throughput is to reduce work in progress. Figure 2.8 shows an example of this technique, based on a value stream with a series of 'workstations'. These consist of individuals, teams, or departments that are tasked with executing work based on input from another workstation, and then handing over their output to the next workstation.

Instead of utilizing each workstation in a value stream to maximum capacity and 'pushing' work onto the next workstation, work should be 'pulled' when a particular workstation requires input. Although the efficiency (utilization) of workstations may appear low, this is beneficial for the flow of work through the value stream.

Figure 2.8 A value stream with three workstations

Kanban boards can be used to support this way of working by providing a visualization of the backlog of work, the work in progress, and the completed work.

The theory of constraints offers the following guidance on how to improve throughput:

- Identify the weakest workstation in the value stream.
- Lighten the load as much as possible.
- Organize work around the weakest link. The least productive workstation determines how fast each other workstation should work in order to achieve maximum throughput. This is important, as having each workstation (or functional team) in the value stream operate efficiently at maximum productivity will often result in a backlog of work for the next workstation.

2.5.2.2 Agile

Building on the Lean principle that small batch sizes of work are beneficial to throughput, approaches with Agile characteristics focus on delivering small iterations of products or services, in frequent increments, so that the approach can be adjusted in response to changes in the environment. In these approaches, feedback, in the form of information, is gathered as quickly as possible and decisions are delayed as long as possible.

Agile techniques are focused on ongoing conversations and interactions between software developers, business people, and other stakeholders who improve the customer experience. This description refers to software development, where the Agile way of working evolved, but can also be applied to other domains of work.

Software is developed by small, relatively independent, self-organizing and cross-functional teams in which a user representative (often called the product owner) plays a major role. The development teams are trusted to execute this work, which means that the management role can shift from control to facilitation, creating a more productive environment (this is often called 'servant leadership'). This is often challengingly different from organizing work across specialized functional teams such as design, development, deployment, and operations.

Small teams with cross-trained 'T-shaped' members (members with broad general knowledge and deep specialization in one area; more information on this can be found in *ITIL® 4: Create, Deliver and Support*) can lead to a reduced number of handovers between people. This helps these teams to be highly effective and improves the flow of work. However, when different kinds of tasks are combined and executed by a single person, there is a risk that a task will be executed based on principles that are appropriate for one kind of task, but not for the one that is to be performed. For example, technical expertise is important for coding, whereas empathy is needed when responding to user requests. A service agent who multitasks between analysing event logs and responding to user calls runs the risk of exhibiting the wrong behaviour in certain situations. This can be mitigated by physically reinforcing a change of persona by means of a ritual or artefacts; for example, by moving to another room to take user calls, or by wearing a uniform or badge to indicate a particular function.

The product owner manages a backlog of work, which is prioritized according to its value. This value can be determined by estimating the cost of delay of each work item.

Definition: Cost of delay

The benefits that are expected to be lost when the launch or update of a service offering is delayed.

When managing work, it can be beneficial to have a common understanding across multiple stakeholders (such as development, support, and compliance) of when work is considered completed. This is referred to as the 'definition of done', and comprises the discussed and agreed criteria that specify the required utility and warranty of a proposed product or service. In Agile software development, 'done' often means having software increments that are potentially deployable. DevOps extends this definition from just deployable software to three categories: deployed, released, and available for use. From a co-creational service perspective, a better way to define work as 'done' is when users have achieved the desired outcomes from their investments (see section 4.3.3 for further detail on the definition of 'done').

DevOps methods build on Agile software development and service management techniques by emphasizing close collaboration between the roles of software development and technical operations. Using high degrees of automation to free up skilled professionals' time so that they can focus on value-adding activities, DevOps is able to shine a light on aspects such as operability, reliability, and maintainability of software products that can assist in managing services.

2.5.2.3 Resilient

Approaches with resilient characteristics are focused on maintaining workable availability and performance, and minimizing the effect of incidents. Two examples of approaches with resilient characteristics are site reliability engineering (SRE) and DevOps.

SRE applies a software development mindset to IT operations, and helps to bridge the gap between development and operations. SRE teams are created alongside existing teams for IT operations. These SRE teams split their time between executing IT operations and coaching IT operations teams, and developing software that helps to increase the resilience and performance of IT systems.

DevOps and, more explicitly, DevSecOps, promote the integration of security into the daily work of application development and IT operations, rather than having it as a separate 'policing' function. Here, the security officer's role shifts from specifying requirements and monitoring performance, to enabling practitioners to address security concerns. This enables things to be done faster and more effectively, but often involves a challenging leap of faith in trusting practitioners with this task.

DevOps also promotes the proactive monitoring of IT systems in production. This proactive monitoring is highly correlated with a quicker mean time to restore service (MTRS). There are many available tools that provide information about operational systems. The signal/noise ratio of this information should be considered carefully, as people only have a limited capacity to absorb information. It is not only ineffective but also an unreasonable burden to expect people to distinguish between what information is important and what is not.

Other approaches to resilience include antifragility, architecting for resilience in software and infrastructure, microservices, containerization, feature switches, soak testing, and disaster recovery.

2.5.2.4 Continuous

Approaches with continuous characteristics, such as continuous integration, delivery, and deployment (CI/CD), are based on the belief that small and frequent batches of work are not only more valuable because functionality can be used earlier, but are safer because the change is smaller and feedback is obtained sooner.

Continuous integration, continuous delivery, and continuous deployment (CI/CD) are descriptive terms for a collection of practices primarily associated with software engineering, which are central to the philosophy of Lean and Agile software development. The adoption of these practices has grown rapidly, and it is important to understand the defining characteristics of CI/CD in the wider context of evolving system development practices

when implementing services that are underpinned by software development. These are described in further detail in section 4.2.5.

Key to continuous integration, delivery, and deployment are a healthy working relationship between all parties involved, and extensive automation:

- **Build automation (the CI phase)** Encompasses version control and the merging of multiple developers' changes into one shared code branch.
- **Test automation** Automatically tests and validates each change in production-like environments.
- **Automatic provisioning** Installs and configures hardware and software to activate a customer's purchased services.
- **Deployment automation** Automates the process of moving code from pre-production environments to the production environment.
- **Post-deployment testing** Validates the functional and non-functional properties, in particular performance/load testing, which is difficult to test before deployment.

2.5.2.5 Combining HVIT characteristics to co-create value

Organizations that are Lean, Agile, resilient and continuous are better equipped for value co-creation in the form of services that can be easily adapted for ever-changing environments and customer needs.

Service science defines service as the application of resources (including competencies, skills, and knowledge) to make changes that have value for another organization. ITIL defines a service as a means of enabling value co-creation by facilitating outcomes that customers want to achieve, without the customer having to manage specific costs and risks. Regardless of which definition is applied, in any service there are at least two interacting entities (called service systems in service science). Service providers and service consumers form a simple example of a pair of service systems, but there are others, such as regulatory bodies. A core concept in service science is service-dominant logic.

Definition: Service-dominant logic

A mental model of an (economic) exchange in which organizations co-create value by applying their competencies and other resources for the benefit of each other.

Service-dominant logic regards service as the process of doing something for and with another party. Value creation is a collaborative process. In service-dominant logic, value is always co-created. Service-dominant logic contrasts with goods-dominant logic, in which the provider delivers value to the customer by transferring ownership of goods.

When service-dominant logic is applied to service delivery, the provider pays more attention to the customer's specific situation, and involves the customer in the service delivery process. This is a more effective way of helping to get the customer's job done. The service interaction is consumer-focused, and the consumer creates value when they integrate and apply the resources of the service provider (and of other service systems) to achieve the exchange. Each consumer determines the value or quality of the service experience based on personal needs at a specific time and in a particular context.

The ITIL story: Key characteristics of high-velocity IT

Solmaz: *At Axle, we pride ourselves on being able to adapt quickly to changes in demand and opportunity. The services delivered by our technology create value not only for us, but for all our stakeholders: our partners, suppliers, and customers.*

Henri: *We ensure that our technology investments align with our goals and comply with governance, regulatory, and compliance requirements.*

Marco: *Our technology choices reflect the four characteristics of high-velocity IT: Lean, Agile, resilient, and continuous. Given the extensive data we collect on our customers, we ensure that our data and technology are resilient to cyber-attack, and are stable under pressure. We work in small batches, tailoring each change to customer requirements, and we employ continuous integration, delivery, and deployment. We also monitor our processes to reduce wasted effort whenever we can.*

2.6 Adopting the ITIL service value system to enable high-velocity IT

HVIT aligns high-value IT-related work with high-velocity business, from innovation to value realization. This requires fast flow, fast feedback, and fast improvement, and not only results in things getting done faster but also improves the quality of IT-related products and services. This has significant implications for organizational IT operating models. Digitally enabled organizations define and structure their IT-related activities and resources differently from organizations in which IT is of less strategic significance. The higher demands that digitally enabled organizations place on technology are reflected in the way they operate. For instance, a digitally enabled organization may have relatively independent product/service-based teams rather than a functional organizational structure, and a higher appetite for rapid iterative experimentation and failure.

This section illustrates how ITIL guidance can be used to provide building blocks for an IT operating model that defines HVIT work and its organization.

An operating model is the 'back end' of an organization's business model, and delivers the value propositions defined in that business model. The operating model is a representation of the building blocks of an organization and the relationships between them. It can be used as a description of the current state of the organization, or as a future state design, referred to as a 'target operating model'.

Definitions

- **Operating model** A conceptual and/or visual representation of how an organization co-creates value with its customers and other stakeholders, as well as how the organization runs itself.
- **High-velocity IT operating model** An IT operating model where digital technology plays a major role in the co-creation of value.

A high-velocity approach is typically required for an operating model when there is co-creation of value with customers, when operating models are becoming significantly more digital, or in response to uncertain, ambiguous, and rapidly changing circumstances. There is often as much value, if not more, in the process of creating the operating model as in the end product.

An HVIT operating model is one in which digital technology plays a major role in the co-creation of value. All operating models have some digital elements, but a digital operating model is one where digital technology makes possible a model that is otherwise not feasible or practical without it.

Operating models for digitally enabled organizations are characterized by their focus on:

- dedicated value streams for each of their products and services
- co-creational culture that fosters high performance and continual improvement
- permanent product/service-based teams over temporary project teams
- automation of IT processes, including infrastructure as code.

In ITIL, value streams are at the core of the operating model because they are the sets of steps required to deliver products and services. HVIT can be supported by many elements of the ITIL service value system (see Figure 2.9).

The following sections outline how HVIT relates to the concepts of:

- digital products and services
- digital product lifecycles
- the ITIL service value chain
- value streams
- ITIL management practices
- the four dimensions of service management
- external factors
- governance and management.

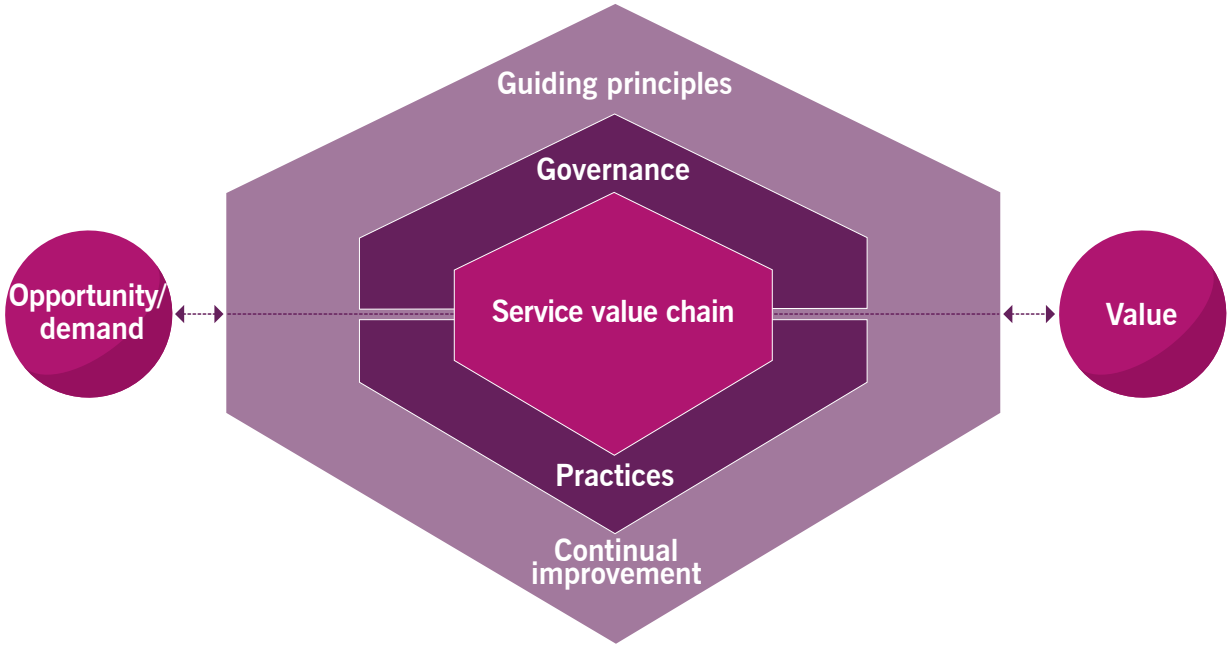

Figure 2.9 The ITIL service value system

HVIT organizations can also benefit from the application of the ITIL guiding principles. These principles, and how they can be used, are discussed in more detail in Chapter 3.

2.6.1 Digital products and services

ITIL defines a product as a configuration of an organization's resources designed to offer value for a consumer. The product is provided to the consumer by means of services that enable the co-creation of value by both parties. These services manifest themselves to the consumer as a combination of acquiring goods, using the provider's (tangible) resources, and interacting with the provider. The value is co-created during service provision and consumption, when the consumer applies its own resources to use the product.

Some products can be defined as digital.

Definition: Digital product

A product is digital when digital technology plays a significant role in its goods, resources, or associated service interactions.

In the sense of the definition of digital product, a significant role usually means that without the technology, the product either does not exist or lacks critical features or characteristics.

Services are based on one or more products, and may offer digital and analogue user experiences. Examples of services based on digital products include online banking, ride sharing, e-commerce, e-tickets for events, photo and video storage and sharing, e-learning, airline flight booking and management, music streaming, smartphone apps, and digital cameras. As digital algorithms become more sophisticated, however, the analogue experience will typically only be to deal with exceptional circumstances that exceed the scope of the algorithm.

In order to co-create value, providers and consumers engage in a service interaction. A service interaction is what takes place when a consumer makes actual use of a provider's service offering, which is based on the provider's products and other resources.

A service interaction between a consumer and a provider is a combination of three kinds of interactions:

- **Service actions** Where (either in response to a request or proactively) the service provider applies its resources, for example by providing information about the use of an IT service. HVIT organizations use digital products to automate large volumes of service actions and interactions with consumers for greater efficiency and effectiveness.
- **Access to resources** Where the service consumer utilizes the service provider's resources, for example by logging in to a website. HVIT organizations provide access to their resources, or access the consumer's resources using channels provided by digital products.
- **Transfer of goods** Where the service consumer acquires ownership of the service provider's resources, for example by purchasing a laptop. HVIT organizations typically do not offer physical goods as part of their digital products and services.

The sight that the provider and consumer have of each other's resources is referred to as the 'band of visibility'. Some resources are visible to the other party, and some are hidden or 'invisible'. During a service interaction, the consumer interacts with some of the provider's resources, and is affected by other 'invisible' resources used by the

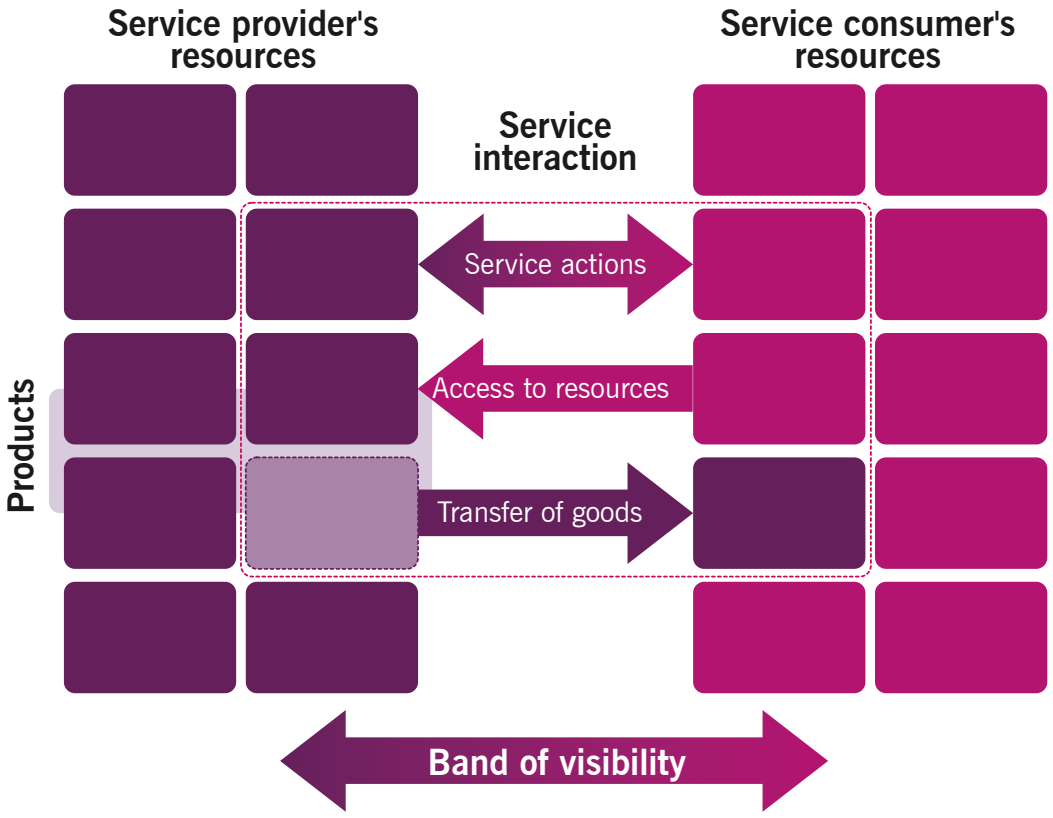

Figure 2.10 Service interaction and the band of visibility

provider that play an indirect role in the service interaction. Similarly, the service provider interacts with some of the service consumer's resources, and is indirectly affected by the service consumer's own 'invisible' resources.

The width of the band of visibility is crucial to the service experience and the effectiveness of the service interaction. If it is too broad, either party may feel confused or distracted by events that have no relevance or bearing to the interaction. If it is too narrow, either party may feel frustrated by a lack of information and influence. A delicate balance must be struck, and the correct way to handle this varies across different providers and consumers. To handle it effectively, each party should be able to perceive the other's needs and adjust their approach appropriately. HVIT organizations often use a complex network of their own products and services alongside those provided by partners and suppliers. In such an ecosystem, providers typically have greater visibility of consumer resources than vice versa.

An example of a service interaction and the band of visibility is shown in Figure 2.10.

More information on the band of visibility can be found in *ITIL® 4: Drive Stakeholder Value*.

2.6.2 Digital product lifecycles

Service consumers and service providers have different perspectives on digital products. They each have their own product lifecycles, which overlap during the period of engagement between the consumer and provider. For the service provider, the lifecycle of a product lasts for as long as there are potential customers for that product. For the service consumer, the lifecycle continues for as long as the product is used, and strictly speaking is a product use lifecycle.

From the perspective of a service consumer, the lifecycle of a digital product starts with the exploration of the market for possible solutions for a particular demand. These consumers can be characterized in terms of

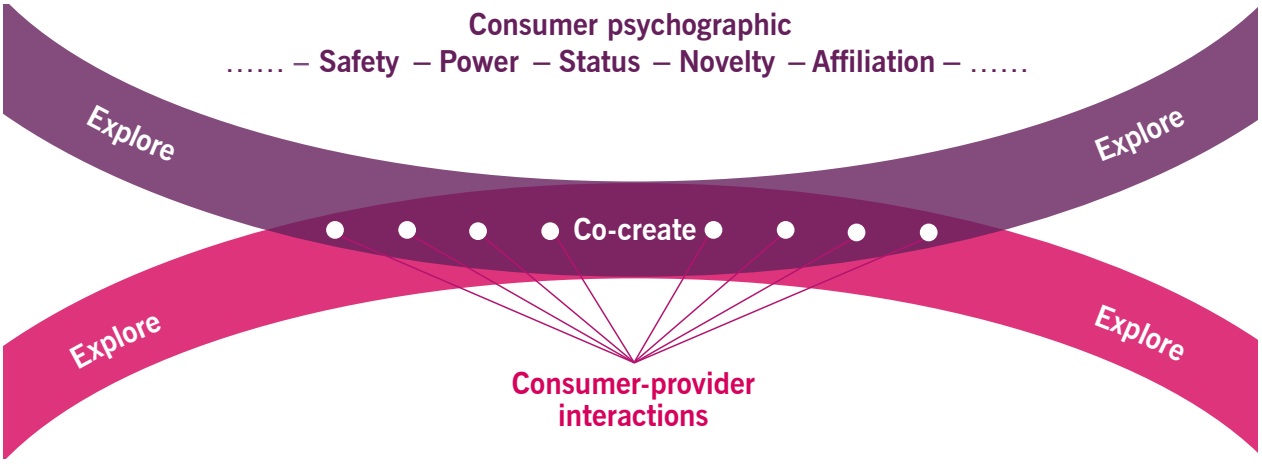

Figure 2.11 Digital product lifecycle from two perspectives: consumer and provider

psychographics mapping their shared personality traits, beliefs, values, attitudes, interests, lifestyles, and other factors. The service provider often considers these when designing and offering a product or service.

From the perspective of a service provider, the lifecycle of a product begins with the exploration of market opportunities for investment in new products, when they are seeking out potential customers for existing products.

At a certain moment, the service consumer and provider will discover one another and engage, sometimes leading to a transaction.

Before a service can be provided and consumed, there are onboarding activities, where preparations are made on both sides. Then the provider and consumer begin interacting, using the service to co-create value, until either party announces the end of the engagement. This is followed by offboarding activities and disengagement. There are many valid reasons for the termination of a service relationship, including:

- the consumer no longer has a demand for the product
- the consumer is dissatisfied with the provider
- the provider cannot fulfil changed demand
- the provider retires an unprofitable product.

Figure 2.11 illustrates how service consumers and providers each have their own digital product lifecycles. The service consumer engages the service provider, possibly as a replacement for a previous service provider. After the period of engagement, the contract is terminated and the service consumer either replaces the service provider with another one, or no longer needs a service provider at all.

At a more detailed level, the customer journey comprises seven interactions:

- **Explore** Understand markets and stakeholders.
- **Engage** Foster relationships.
- **Offer** Shape demand and service offerings.
- **Agree** Align expectations and agree service.
- **Onboard** Get on board or leave the journey.
- **Co-create** Provide and consume.
- **Realize** Harvest value and improve.

A model of the customer journey is shown in Figure 2.12. More information on the customer journey can be found in *ITIL® 4: Drive Stakeholder Value*.

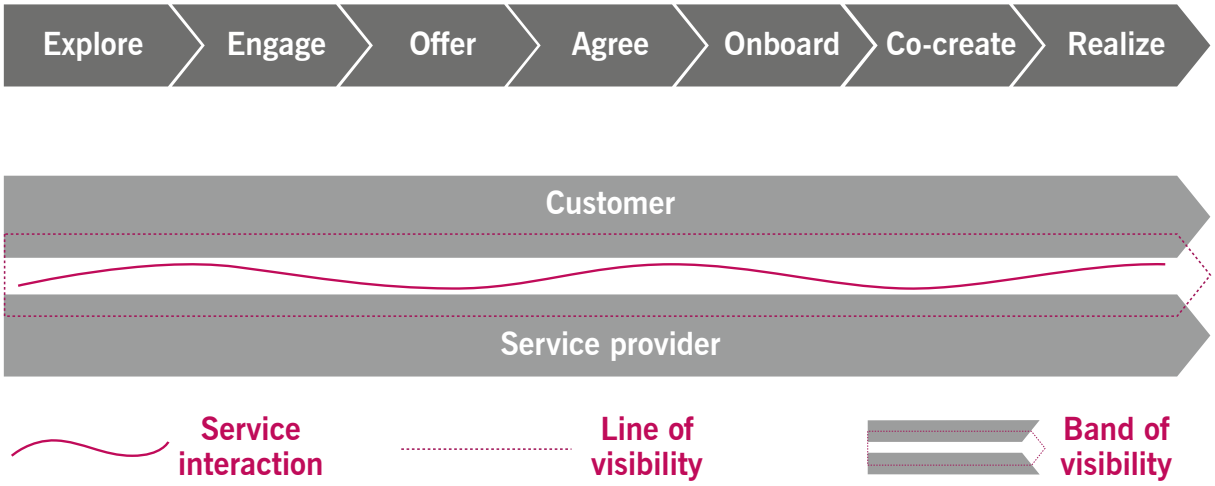

Figure 2.12 Customer journey model

When a service provider is replaced, there is a period of transition for both parties, before the new provider's service can be provided and consumed. The service consumer often requires assistance and assurance in transferring the service from the current service provider to the new one.

When the consumer still has demand for a solution, they are faced with the task of offboarding the unsatisfactory or retired product, in parallel with the onboarding of its replacement. The risk of discontinuity is a serious concern. To reduce this concern, providers sometimes offer to take care of the transition from the current provider to the successor. Figure 2.13 shows the consumer's perspective of digital product lifecycles.

Table 2.4 outlines the lifecycle stages of a digital product from the perspective of the service provider and service consumer.

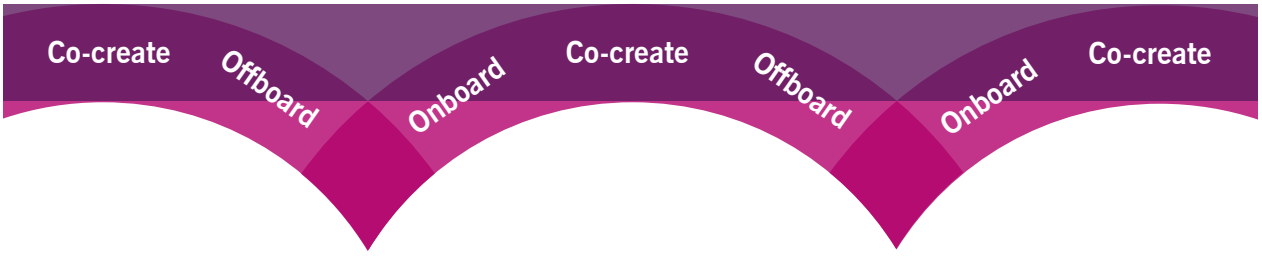

Figure 2.13 Digital product lifecycles from a consumer's perspective

Table 2.4 Stages of the digital product lifecycle

Lifecycle stage	Service provider	Service consumer
Exploration	The service provider researches and develops the product and service offering.	The service consumer becomes aware of the existence of a product and assesses it as interesting and then desirable, after which an agreement is made.
Onboarding	An instance of the product is installed, and the user organization is onboarded, sometimes with a transition from a replaced product.	
Co-creating value	The service provider delivers and supports the product and experiences increasing, stable, or decreasing return on investment, leading to an investment decision to buy (use and improve the product), hold (use but do not improve it), or sell (use and reduce, replace, or retire it).	The service consumer uses the product and experiences increasing, stable, or decreasing value, eventually leading to a decision to replace or retire the product.
Offboarding	The instance of the product is uninstalled and the user organization is offboarded, sometimes with a transition to a replacement product.	
Retired	The service provider no longer offers or supports the product.	The service consumer no longer uses this product, but other consumers may use it.

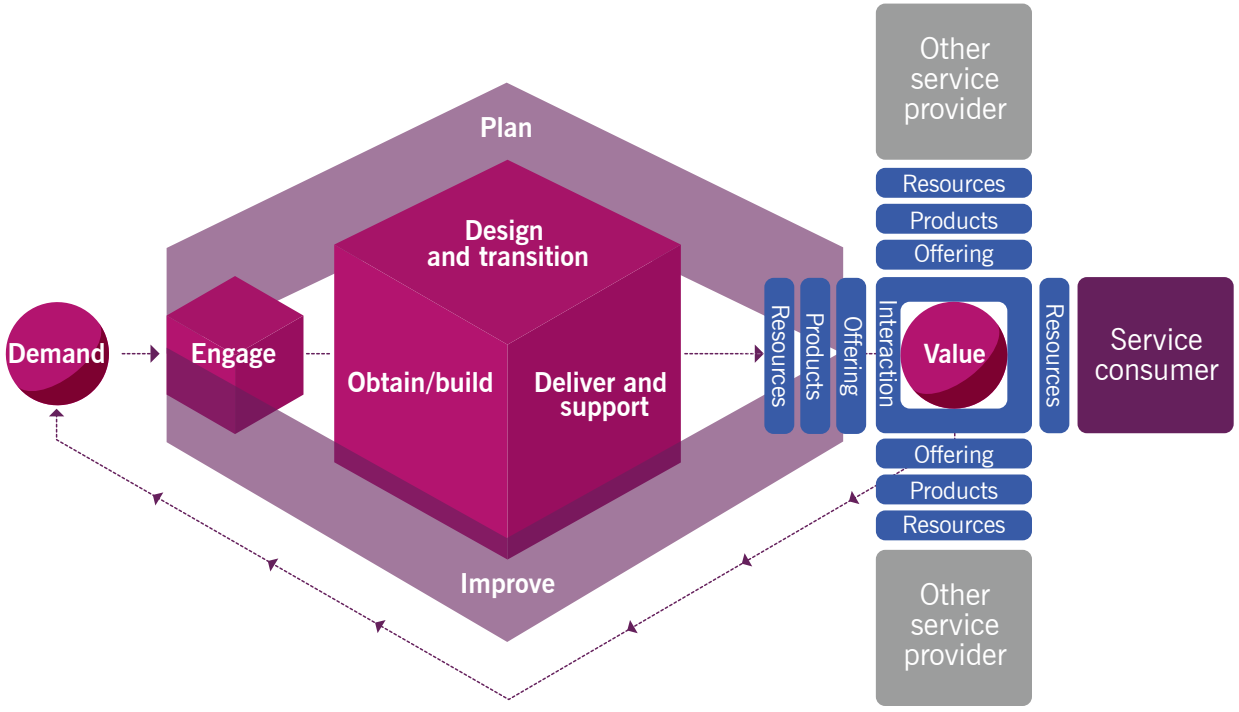

Figure 2.14 The ITIL service value chain

2.6.3 The ITIL service value chain

The activities that are required to deliver products and services are modelled by the organization's service value chain and practices. The service value chain describes the organization's archetypal activities (see Figure 2.14).

The ITIL service value chain is useful for describing, at a fairly high level of abstraction, the types of activities that a service provider executes. It helps people to focus on the goal of each value chain activity, and the inputs and outputs, without getting lost in the details of the lower-level activities in a value stream.

The value chain activities are closely interrelated, and can be arranged in any order to explain and discuss a range of different situations.

2.6.3.1 Value chain activities and DevOps

In HVIT environments, the concept of 'continuous' is often used, and is a key characteristic of many HVIT approaches. 'Continuous' refers to fast, iterative cycles of activities, enabled by short feedback loops. CI/CD is a well-known example of this, where new software increments flow through part of a value stream without delay, enabled by a high degree of automation.

The DevOps community often illustrates this using a figure of eight with a continuous loop of connected application development and IT operations activities, as shown in Figure 2.15. The Dev and Ops activities are enabled by infrastructure and platforms for coding, testing, deployment, production, and so on.

This loop can be used alongside the ITIL service value chain activities to give a combined DevOps and ITIL perspective. In ITIL terminology, DevOps focuses on developing, deploying, and running concrete service components rather than intangible services. The dominant service components in DevOps are the applications, data, and platforms that, together, form a product for the consumer.

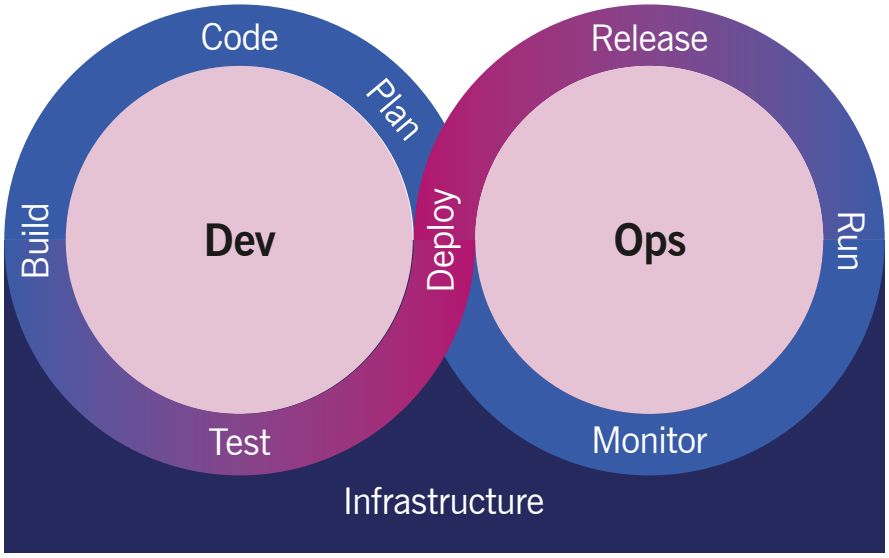

Figure 2.15 DevOps activities in a continuous loop

The focus of the service value chain is on products and services rather than individual service components. It describes what is needed for the interaction between the service provider's products and other resources, including the service consumer's resources.

Figure 2.16 shows how the DevOps loop and the service value chain activities can be combined. Some service value chain activities (such as design and transition) have been split into two 'sub-activities' that are easier to map to parts of the DevOps model.

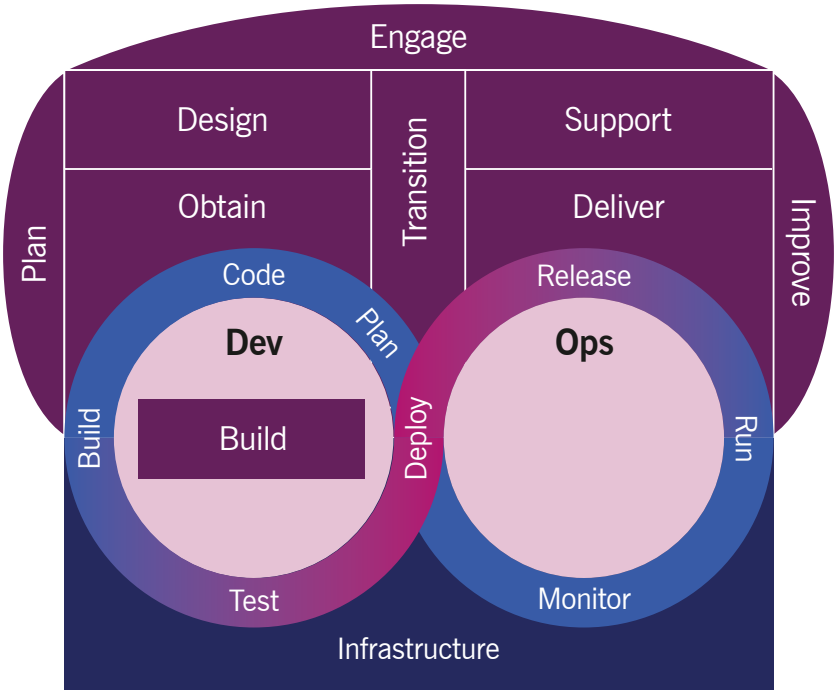

Figure 2.16 DevOps and the service value chain

The service value chain activities link to the DevOps model in Figure 2.16 in the following ways:

- The design value chain activity runs in parallel with Dev, focusing more on the service than the software product.
- The obtain value chain activity is the interface with external Dev, integrating the developed software product with the other resources that the service comprises.
- The build value chain activity corresponds to internal Dev.
- The transition value chain activity runs in parallel with, and corresponds to, the deployment from Dev to Ops.
- The deliver and support value chain activity corresponds to Ops, and is often more comprehensive than Ops.
- The engage value chain activity runs in parallel with all of the underlying activities.
- The plan value chain activity corresponds to the planning part of Dev.
- The improve value chain activity corresponds to one of the tenets of DevOps: continual experimentation, learning, and improvement across all DevOps activities (often referred to as the 'Third Way' of DevOps).

This is an example of how HVIT can be applied to the service value chain, to help bridge the gap between professional disciplines by enabling the discussion of work from different perspectives. To improve collaboration between disciplines, it is important first for each discipline to understand the other's perspective and to use illustrations that the other discipline is familiar with. Once this has been established, the discussion can move from understanding to being understood.

2.6.3.2 Service consumer

Another way that HVIT can be applied to the service value chain is in illustrating how closely service providers and service consumers interact. A service consumer regards the high-level interaction with a service provider in terms of acquisition, provision, and use.

The consumer translates their demand into requirements, and engages a provider to provide services that they then use to create value. This can be represented as a service value chain from a consumer's perspective, as shown in Figure 2.17. This illustration can be expanded to incorporate the service provider's perspective in terms of service value chain activities.

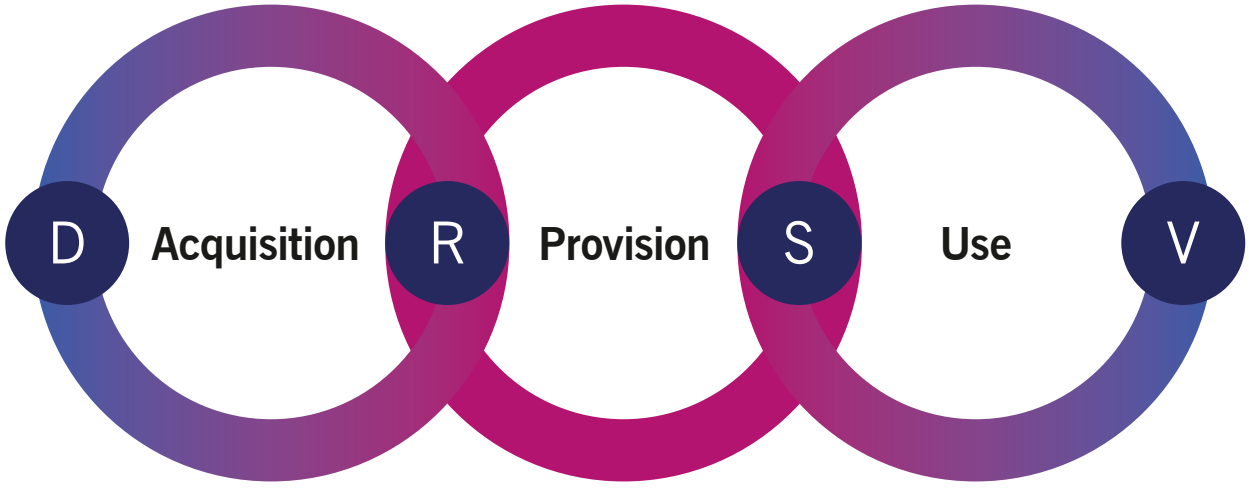

Figure 2.17 The service consumer's perspective

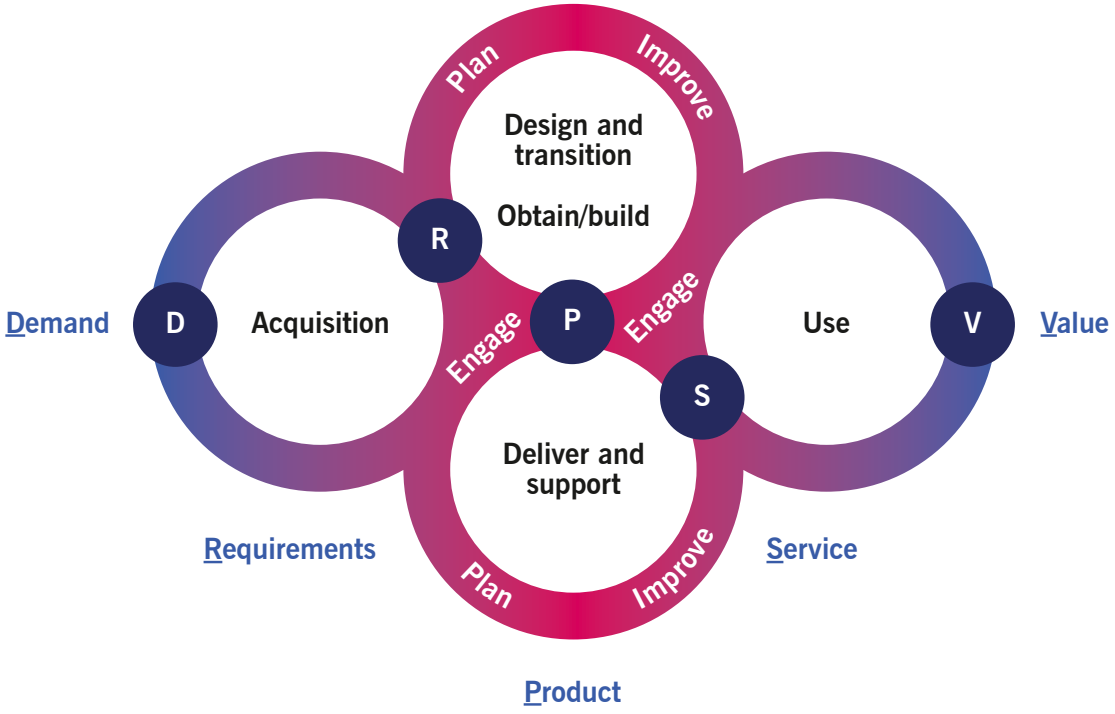

Figure 2.18 Interacting service value chains

Figure 2.18 shows the six value chain activities of the service provider, with initial design and development in the top half, and deliver and support (in parallel with subsequent enhancement) in the bottom part. The primary interfaces between the consumer and provider are marked by the consumer's requirements (R) and the provider's service (S). Design and development and deliver and support are connected by the product (P) that is deployed.

Figure 2.19 shows how a value stream can flow across the value chain activities. As with the linking of the value chain activities and DevOps, this combination can be used to promote better ways of working (including how to obtain faster feedback).

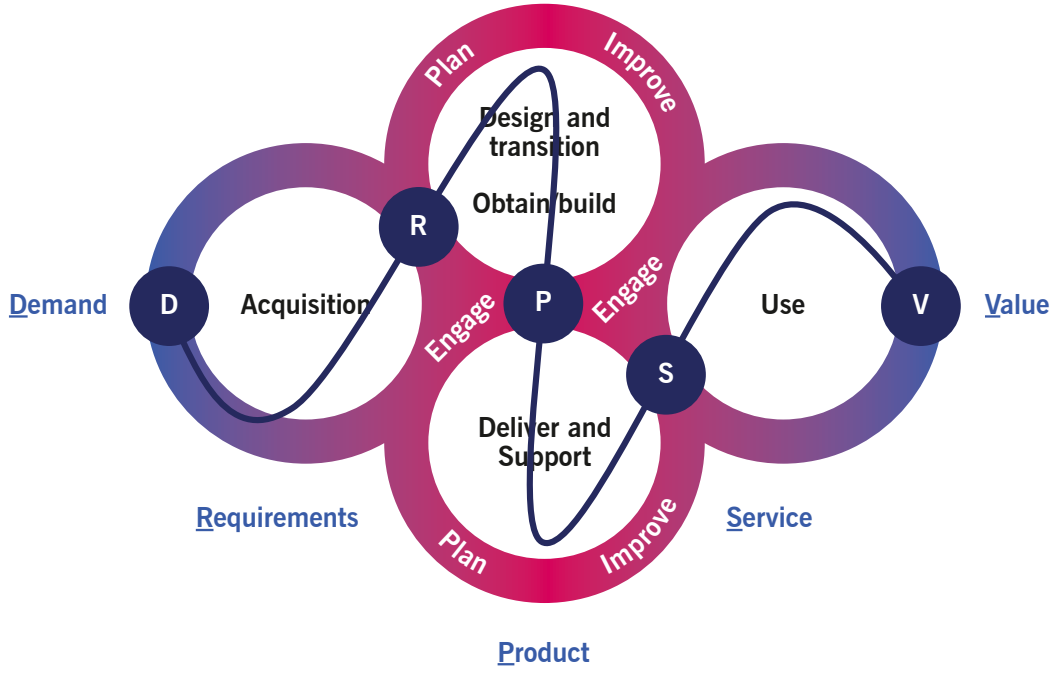

Figure 2.19 Example value stream referring to service value chain activities

2.6.4 Value streams

A value stream is a series of steps an organization undertakes to create products and services and deliver them to consumers. They are sets of activities that are performed according to guidelines (principles, approaches, techniques, etc.) and within constraints (regulations, policies, budgets, etc.).

It is important to recognize that the service value chain and associated practices are models on which actual value streams are based. Value streams can be observed where people and other resources act to co-create value. Service value chains and associated practices cannot be observed, as they are abstract representations that are used to model the value streams.

This is depicted in Figure 2.20, in a value stream that is triggered by demand for value, which is co-created using products and services. The value stream consists of steps in which resources are combined resulting in products designed to enable value creation, not only for the service consumer but also for other stakeholders. Activities performed by the resources are guided by various methods adopted by the organization (descriptions of value streams and processes/procedures/work instructions in the practices, descriptions of the activities in the service value chain, and the guiding principles).

The service value chain describes the activities that are required to effectively manage products and services, whereas a value stream comprises an actual series of steps to create products and services and deliver them to consumers. As such, value streams can be considered to be where things actually happen: where the ITIL practices are used and value is co-created. There is no set structure for value streams, and they are unique to each organization. HVIT organizations often are product/service-oriented and have multiple value streams that reflect the diversity of their products and services. Their operating models therefore comprise multiple value streams.

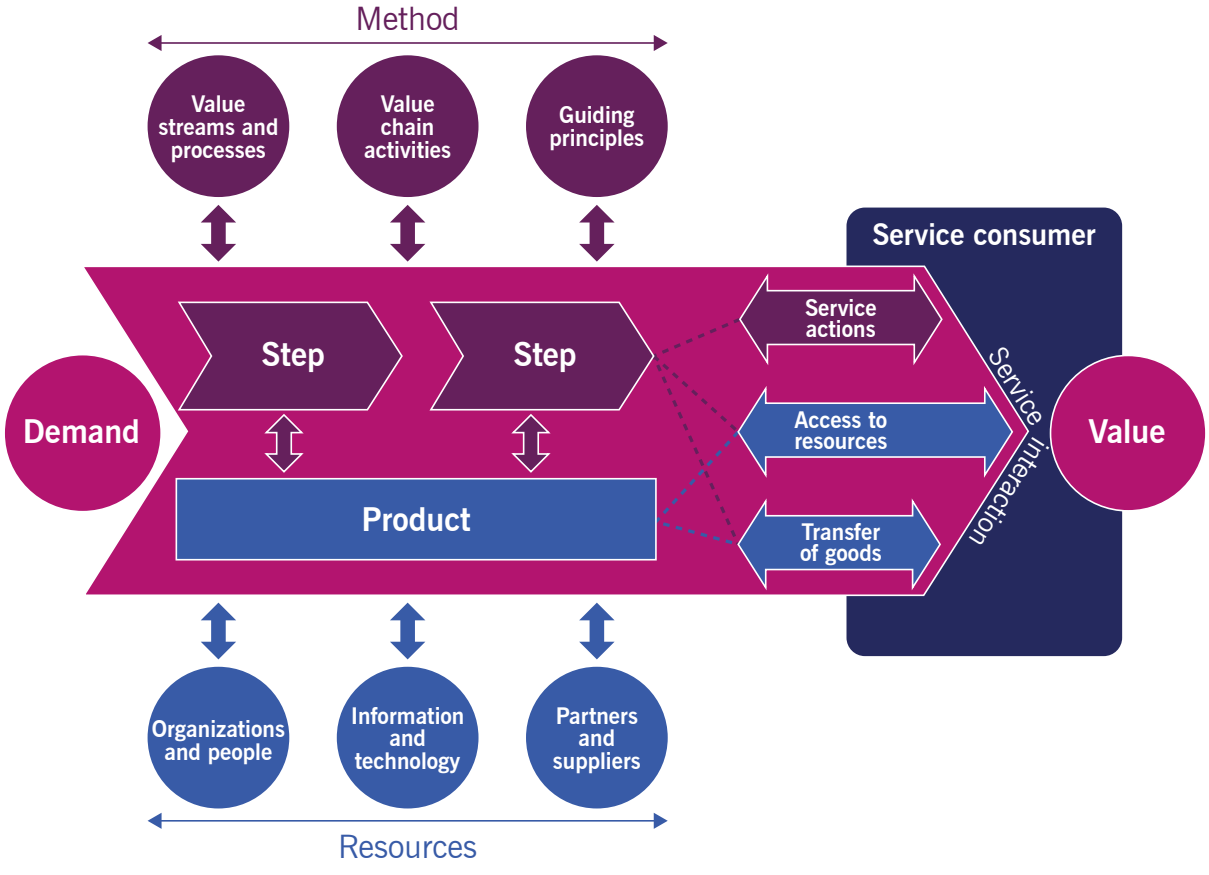

Figure 2.20 The value stream in context

The word 'stream' indicates the importance of throughput: getting work done quickly and on time. Many factors can slow things down in a value stream, including handovers and overloading, making queue management very important. Throughput will benefit if each task is done by one person or team, although this is rarely the case. It can also be increased by limiting the incoming flow of work to the capacity of the 'workstation'. From an end-to-end perspective, this means that the workload should be balanced across the whole value stream.

It is also important to get feedback as quickly as possible, not only to make any improvements that may be needed, but, crucially in a volatile HVIT environment, also to evaluate whether the way of working or way of thinking needs to be improved. In these dynamic environments, the improvement of daily work is as important as actually doing daily work.

In HVIT environments, systems are often complex and therefore unpredictable. This makes it less likely that detailed processes, procedures, and work instructions will be useful, as they often will not be followed. It is also not useful or feasible to predict or dictate the sequence of steps in a value stream, and the activities within those steps, other than at a high level of abstraction. Instead, the sequence of activities and steps will often emerge during, and as a result of, the 'micro-interactions' that take place during execution. This means practitioners must be able to observe the expected and unexpected changes that their actions, and those of others, make, and adjust their next actions accordingly. This way of working is exploratory rather than confirmatory, shifting from prediction and control to insight and understanding; from large, occasional changes to small, frequent ones; from detailed planning in advance to constantly experimenting and learning; and from failsafe to safe-to-fail.

Three key aspects of value streams are governance, execution, and improvement, as shown in Figure 2.21. Governance applies to both execution and improvement, and in turn improvement applies to both execution and governance. In execution, operations are provided with the necessary resources, and operations and resourcing are managed. The value stream is positioned in execution, and is therefore governed and improved.

The value stream that is governed, managed, resourced, and continually improved, together with the resources that guide and enable it, can be regarded as the core of the operating model: it is how an organization co-creates value with its customers and other stakeholders.

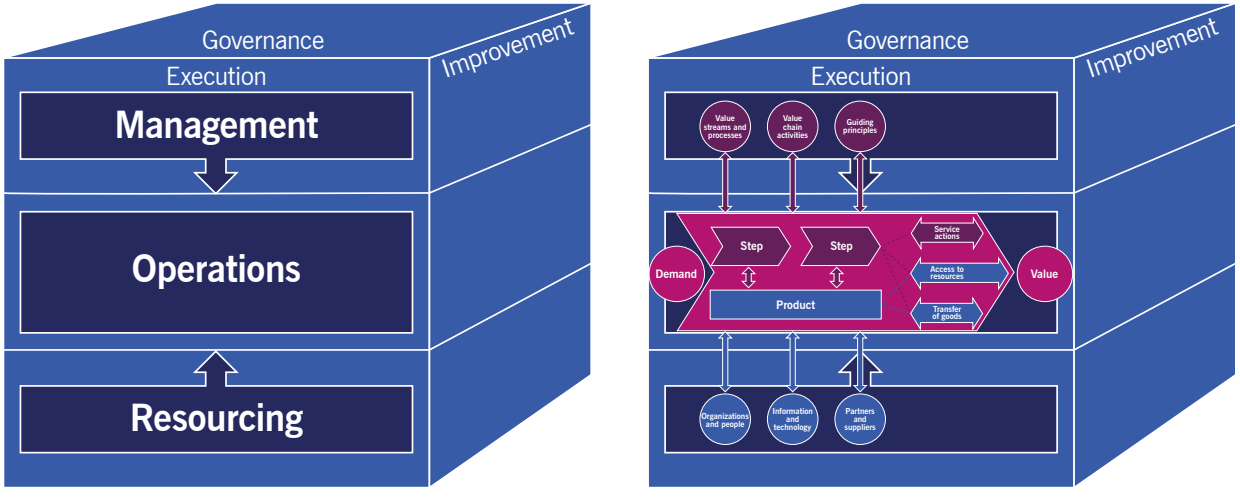

Figure 2.21 Value stream positioned with respect to governance, execution, and improvement

2.6.5 ITIL management practices

The ITIL management practices describe resources and activities required to fulfil specific purposes, and generally accepted ways to manage those resources and activities.

Organizations provide services, and often goods, that help their customers. This entails the execution of a set of activities, either with or without the direct involvement of the customer. In order to perform these activities, both the organization and the customer have to have certain abilities. These abilities may be exclusive to a particular activity or may be required by multiple activities, such as maintaining an effective relationship with one another during each stage of the engagement. Abilities involve the interaction of a combination of people and various kinds of resources.

ITIL terminology

The ITIL practices represent an organization's ability to fulfil a specific purpose. This involves resources of all four dimensions of service management. A practice may be supported by various organizational solutions including dedicated teams and tools. Although these teams and tools may have the same titles as the practices they are part of, they should not be confused with them.

There are 34 ITIL management practices, grouped into three categories:

- **General management** practices, such as organizational change management, are found in many other business domains, but also apply to service management.
- **Service management** practices are at the core of service management, and are described in detail so that the reader not only understands the practice but also knows how to develop and use the required ability.
- **Technical management** practices are closely related to service management practices because they focus on service components: in other words, the technical resources that are part of a service. These technical resources are applications, data, platforms, and infrastructure.

A practice in itself is just a collection of resources that enables an organization to do something. Value only emerges from a practice when it is applied within the context of a value stream. The ITIL practices can be regarded as building blocks for an organization's value streams. Each practice describes how it could be used in

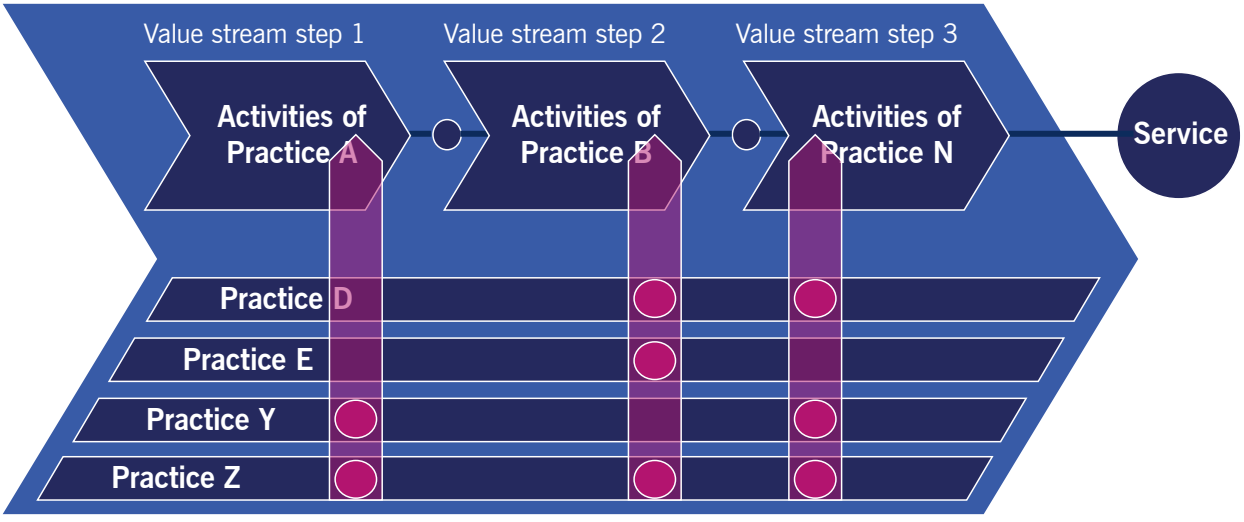

Figure 2.22 Multiple management practices involved in a value stream, depicted in a variation of a Porter value chain

a value stream, but each value stream will apply the practices in its own specific way. Although standardization has its benefits, particularly for efficiency, it is often more effective to tailor the practices to the specific needs of each value stream.

Each step of a value stream consists of activities defined and described in management practices; other practices may contribute to the value stream with information, tools, or methods. This is shown in Figure 2.22.

2.6.5.1 Key practices for HVIT

The practices that are most relevant to HVIT and digital products are illustrated in Table 2.5.

Table 2.5 Practices and their relevance to the five objectives

	Valuable investments	Fast development	Resilient operations	Co-created value	Assured conformance
Architecture management		✔	✔		
Availability management			✔	✔	✔
Business analysis	✔	✔		✔	
Capacity and performance management			✔	✔	✔
Change enablement		✔	✔	✔	
Continual improvement	✔	✔	✔	✔	✔
Deployment management	✔	✔	✔		
Incident management			✔	✔	
Information security management			✔		✔
Infrastructure and platform management		✔	✔		
IT asset management	✔				✔
Knowledge management	✔			✔	
Measurement and reporting	✔			✔	✔
Monitoring and event management			✔		
Organizational change management		✔		✔	
Portfolio management	✔			✔	
Problem management			✔		
Project management		✔			
Relationship management	✔			✔	
Release management		✔	✔	✔	
Risk management	✔		✔		✔
Service catalogue management				✔	
Service configuration management		✔	✔		✔
Service continuity management			✔	✔	✔
Service design	✔	✔		✔	
Service desk				✔	
Service financial management	✔			✔	✔
Service level management	✔			✔	
Service request management			✔	✔	
Service validation and testing		✔	✔		✔
Software development and management		✔	✔		
Strategy management	✔			✔	
Supplier management		✔	✔	✔	
Workforce and talent management	✔	✔	✔	✔	✔

2.6.6 The four dimensions of service management

Services and service management require active and passive resources, which could be anything an organization has acquired or developed to undertake relevant activities. An organization's employees, partners, and suppliers are active (operant) resources, or actors. They interact with other actors, and with passive (operand) resources and products.

Examples of organizational resources include:

- organizations (including finances and physical resources such as buildings) and people
- information and technology
- partners and suppliers.

These are three of the four dimensions of service management that are a key component of the ITIL framework. The four dimensions are influenced by the PESTLE variety of external factors: political, economic, social, technological, legal, and environmental, as shown in Figure 2.23.

The four dimensions of service management identify the kinds of resources that are used in a value stream. Three of the four dimensions (organizations and people, information and technology, and partners and suppliers) are concrete resources that are used operationally during the execution of the value stream steps. The value streams and processes dimension represents abstract resources that are used as input for the design of value streams. This is shown in Figure 2.20.

Examples of the application of each dimension to HVIT are described in the following sections.

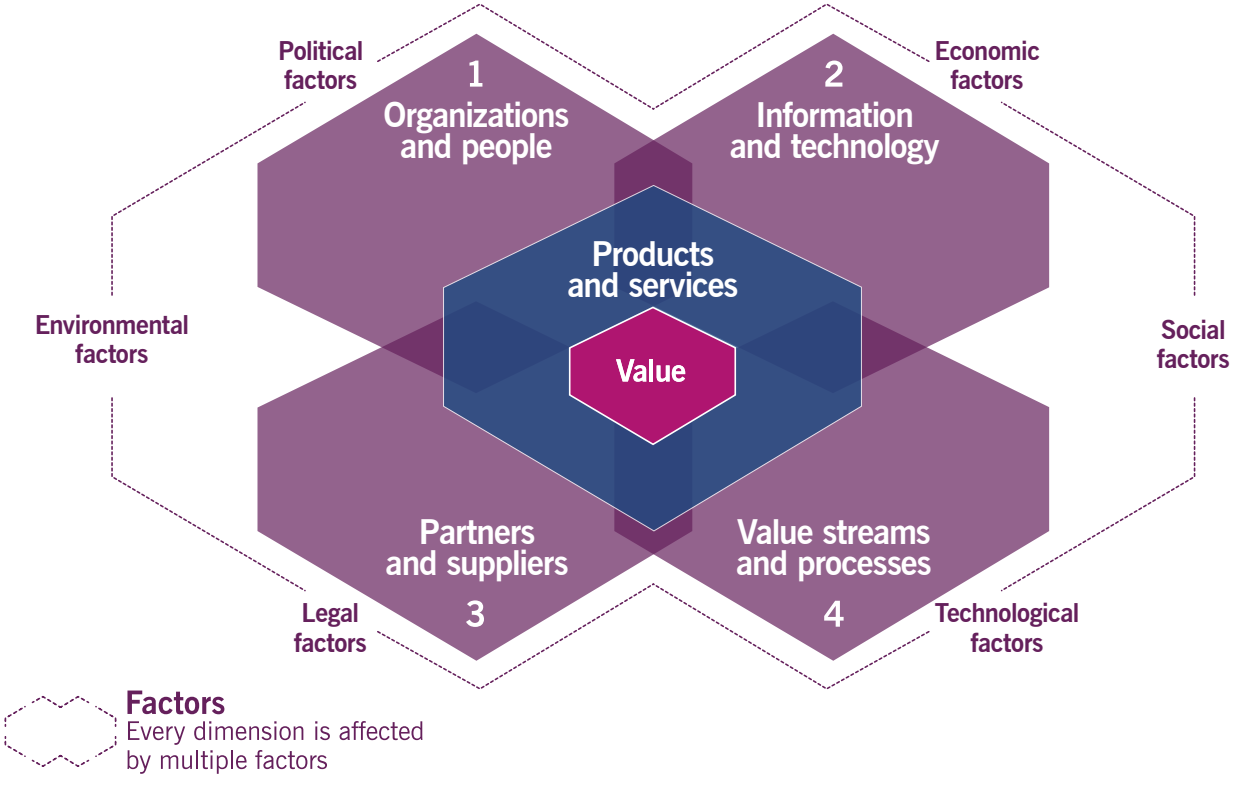

Figure 2.23 The four dimensions of service management including the six PESTLE factors

2.6.6.1 Organizations and people

In HVIT environments, where IT is an integral part of an organization's products and services, it is likely that the IT function will be an integral part of the lines of business that are responsible for the various products and services. There will often be multifunctional product/service-based teams, with both business-oriented and IT-oriented team members. Although there may be a centralized IT service centre for non-differentiating digital technology such as email and Wi-Fi, the differentiating digital technology will often be managed as part of the primary activities. There may also be platform-oriented teams that support various decentralized product/service-based teams. One of the organizational challenges in such environments is to manage a hybrid environment where some technology is dedicated and some is shared, and where small, dynamic, product-oriented systems have to interact with large, less flexible backend systems.

Within product/service-oriented teams, there are no service level agreements between business and IT, as they are part of the same team. The same applies to agreements between IT groups within the team. There may still be metrics and indicators, but these are directly aligned with the primary objectives of the line of business, not as part of internal transactions with an IT function.

IT practitioners in HVIT environments usually work at the same physical location as their non-IT co-workers, often in the context of self-contained product/service-based teams. This is not only good for communication, but also for building a better and common understanding of the business context in which digital technology is used.

2.6.6.2 Information and technology

This section relates to the information and technology that are used as 'production resources' to produce the information and technology that comprise digital products and services; in other words, the tooling that is used to produce the product, not the product itself.

In HVIT environments that derive significant value from digital technology, part of this value is related to the information that the digital systems provide, and part is related to the digital technology that provides the information quickly, resiliently, securely, and efficiently. Higher demands are therefore placed on both the information and technology that digitally enable organizations, and the information and technology (IT tooling) that support the IT processes. An example of HVIT tooling is an automated deployment pipeline that ships new versions of applications faster and more reliably to production.

Information is often a poorly utilized resource in the IT function. Practitioners should be increasingly aware of the data they have at their disposal, and make better use of it. It is common in HVIT environments for information to be created and used in real time and at times spontaneously. For example, monitoring tools provide real-time access to performance information when it is needed (such as when an alert indicates an exception condition or an incident has been detected). Collaboration tools integrated with bots (ChatOps) can then be used to exchange and acquire additional information as needed to take action.

It is critical to understand the flow of information and how work gets done, so that people have the information they need, when and where they need it, as part of their 'work as usual' practices. This critical need is partially addressed by business analysis, but the domain of business information management[1] expands the scope to include the actual use of information. Business information management not only ensures that the right information needs are identified, but also that the information is used effectively in the context of business processes. This applies equally to the information that supports the processes within the IT and service management domain. Freely flowing information is a characteristic of a high-trust organization. This is consistent with the 'generative culture',[2] characterized by high degrees of cooperation, sharing of risks, appetite for innovation, and a desire to learn from failure, that is often found in HVIT organizations.

The duplication of data (with the risk of different sources of truth) can be avoided by integrating systems. This also concerns the use of different tools; for example, developers using one tool and operations using another, with no interfaces and consolidation. Automatically adding a common and visible identifier (such as a Sprint ID, backlog item ID or ticket number) is a way to combine them. For example, adding a user story ID to a change record while a change progresses through a deployment pipeline is a great way of adopting the ITIL guiding principles of 'collaborate and promote visibility' and 'optimize and automate'.

The expected extreme growth of artificial intelligence (AI) and machine learning will only place higher demands on well-managed information and knowledge.

2.6.6.3 Partners and suppliers

HVIT environments typically make extensive use of cloud-based infrastructures, platforms, and other services. These utility services are often provided by organizations on their own terms and conditions. Public cloud-based services are usually high quality and affordable, but the individual consumer has little or no influence on the provider. It is therefore crucial to analyse the dependencies and take appropriate action in terms of contracts and SLAs, including risk-sharing agreements, secondary providers and contingency plans, and workarounds.

When outsourcing work in an HVIT environment, it is important to consider whether the external service provider works in a similar way to their customer, as it is difficult to integrate parties with fundamentally different ways of working into the same value stream. Functional outsourcing, in which discrete functions, such as testing, are outsourced, is often less effective than outsourcing a whole value stream. Because of this constraint, IT functions often contract external staff rather than outsource work.

2.6.6.4 Value streams and processes

HVIT environments recognize that it may be better to create a unique value stream for each digital product or service. This may be less efficient than a standardized and centralized single value stream that serves multiple products and services, but the benefits in terms of effectiveness will often outweigh the costs, so it is advisable to consider this alternative. If this option is considered, the organization should also think about ways to share tools and best practices across value streams where appropriate. When designing unique value streams, it is important to remember that some kind of standardization may be required for scalability, so the trade-off needs to be carefully considered.

A process is a predetermined sequence of interrelated activities that transforms inputs into outputs. Processes can be used to detail the steps in a value stream, and procedures and work instructions can provide further levels of detailed guidance.

Because processes are predetermined, they are applicable to situations that are predictable. When a situation is unpredictable, the application of a process is unlikely to result in the desired outputs and outcomes. In these circumstances, a case-based approach is more effective, as it gives the practitioner freedom to apply their professional judgement as to which activities are appropriate. HVIT environments often deal with complex systems that are unpredictable, so the use of processes should be reserved for those situations where it is feasible to predetermine the appropriate sequence of activities. Practitioners will often think in terms of various patterns of activities that might work, and will experiment to select the right patterns for the task at hand.

2.6.7 External factors

An organization's choices of markets, products and services, and resources and activities are influenced by multiple external factors, which can be political, economic, social, technological, legal, and environmental (PESTLE). These all influence the four dimensions of service management.

HVIT environments are also characterized by relatively high degrees of volatility, uncertainty, complexity, and ambiguity, which are referred to by the acronym 'VUCA'. VUCA presents a serious managerial challenge, and should be considered in the planning and management of organizations. The degree to which an organization experiences the elements of VUCA is reflected in whether it pursues more short-term or long-term plans, and how many large-scale transformation programmes it undergoes.

An organization that experiences VUCA to a greater degree will typically pursue more evolutionary approaches, based on an assessment of the disposition of the actors in the external and internal environment, and experimentation with the factors that influence those actors. In these organizations, the focus is more on managing the present, rather than making and following a roadmap to a predetermined future state.

Managers constrain behaviour by imposing controls driven by external and internal policies, rules, sanctions, and so on. At the same time, workers determine how they contribute to the organization within the boundaries of these controls. In HVIT organizations, practitioners play an active role in organizational performance and improvement. They are also in a position to challenge boundaries where they are less rigid. Practitioners have the opportunity to exhibit leadership by taking the initiative.

2.6.8 Governance and management

It can be easy to confuse the terms 'governance' and 'management' and how they are applied within organizations. Governance, in particular, can be applied to the highest level of non-executive governing bodies, but also to lower levels where it becomes increasingly difficult to distinguish between governance and management. For the sake of clarity, this publication refers to different organizational entities for governance and management. A governing body is at a higher level of authority than the organizational entity that is governed, whereas a manager is part of that organizational entity.

Governance is the means by which an organization is directed and controlled. The governing body evaluates the organization's situation, sets the direction for managers, and monitors the organization's performance. Management is anchored in governance. Managers deal with planning, building, organizing, and improving the organizational entity.

Responsibility for digital technology is an integral part of the lines of digital business in digitally enabled organizations, and is not a separate unit such as a centralized IT service centre (although these may exist for non-differentiating IT services, such as email). The governed and managed organizational entity is therefore responsible for both digital technology and its use in the context of digital products and services. This is beneficial for the alignment of business and IT within an organization, as only business and IT activities and resources need to be aligned, rather than separate organizational entities with different targets. Digital technology practitioners will therefore report to the same management as their less technical co-workers.

Practitioners operate within a governance and management framework. They understand the applicable constraints and know how to act within that framework, and their insight and judgement influence how they act. The more insight they have, and the better their judgement skills, the more the practitioner will take the initiative to bend rules when the associated benefits and risks are justifiable. This can be very beneficial, as HVIT environments are often highly unpredictable, meaning that predetermined instructions are neither feasible nor desirable.

An HVIT practitioner therefore has to exercise judgement on the job. To do this effectively, they must understand the reasoning behind certain constraints that are in place. A major role of the manager in an HVIT environment is therefore to provide context and to enable the practitioner to take charge.

2.7 Summary

Chapter 2 provided an overview of the key concepts related to a digital organization, including digital technology, digital transformation, and high-velocity IT. Understanding these concepts and aligning the taxonomy they form across organizations and ecosystems are critical for the success of digital transformation initiatives, as they usually involve many people from different organizations and backgrounds.

To succeed in a digital transformation, organizations need to be (or become) Agile, Lean, resilient, and able to deliver continuously. All these attributes enable value co-creation in a high-velocity way: faster, with a clear direction of development.

Finally, the chapter explored the ITIL service value system (SVS) and its components from the high-velocity perspective, and described how the components of the SVS can work together in a high-velocity environment to ensure effective, sustainable, and resilient value co-creation.

To summarize, Chapter 2 explained:

- the key concepts of digital transformation and high-velocity IT
- what needs to be achieved to enable high velocity and digital transformation
- how ITIL 4 models and concepts can help in the success of digital transformation.

Chapter 3 will address the cultural aspects of digital transformation: key behaviour patterns for a high-velocity organization and key approaches to the related cultural evolution.

CHAPTER 3

HIGH-VELOCITY IT CULTURE

3 High-velocity IT culture

This chapter explores the kind of culture that supports and enables HVIT work. It outlines five key behaviour patterns that reflect the organizational needs and the aspirations of practitioners to work in a rewarding environment. It also explores a number of models and concepts that support these behaviour patterns and inform organizational culture.

3.1 Key behaviour patterns

There are five key behaviour patterns that reflect organizational needs and the aspirations of practitioners to work in a rewarding environment. These behaviour patterns are intended to appeal to humans' desire to contribute to something worthwhile, to learn and improve, and to be recognized for their intent and effort.

The five key behaviour patterns shown in Figure 3.1 are:

- accept ambiguity and uncertainty
- trust and be trusted
- continually raise the bar
- help get customers' jobs done
- commit to continual learning.

Although most of these behaviour patterns are not exclusive to HVIT, the combination of and adherence to all five is beneficial to the kind of people who understand the demands of a more digitally enabled enterprise.

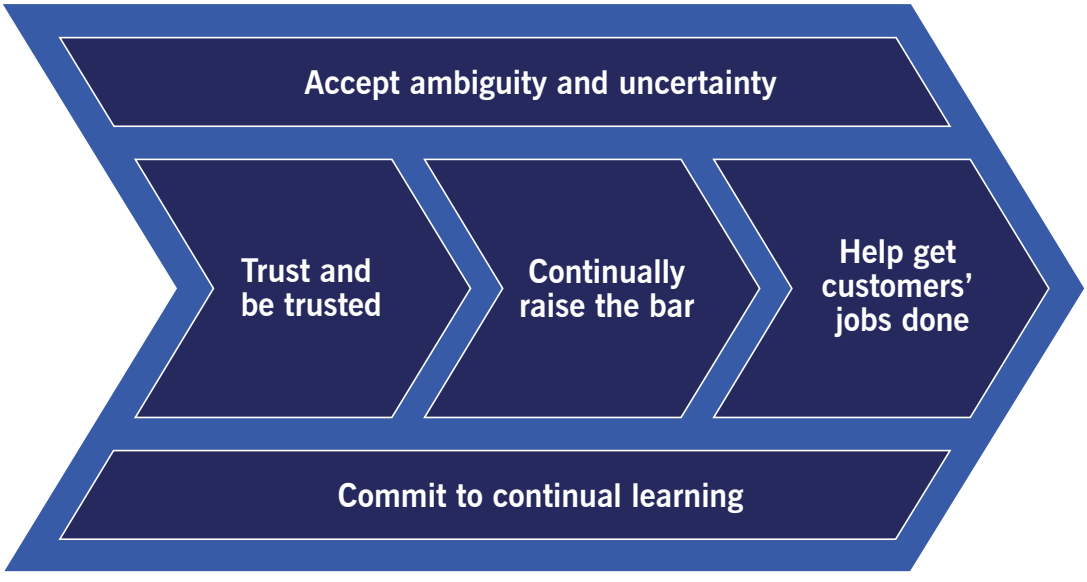

Figure 3.1 Key behaviour patterns

3.1.1 Accept ambiguity and uncertainty

This behaviour pattern reflects the volatile and ambiguous nature of typical digitally enabled organizations. In these environments, where failsafe systems are often an illusion, it is important to favour experimentation and not be afraid to fail safely and mindfully. Practitioners should not be scared of the unknown. They should be able to accept that things are not perfect, and have techniques to deal with that.

3.1.2 Trust and be trusted

This behaviour pattern is about respecting people's professional skills and trusting them to take the right decisions, along with being considerate to others, acknowledging diversity, and being inclusive. People should be given feedback on their work that is honest, but considerate. This behaviour pattern is also related to reducing stress and burnout by paying attention to unfair treatment, toxic relationships, lack of recognition, lack of control, conflicting values, and insufficient resources. Psychological safety is also important to fostering better performance. Juniors should be encouraged to share their knowledge rather than being afraid to speak, and seniors should feel comfortable speaking out if they do not know something, or need to ask for help.

3.1.3 Continually raise the bar

This behaviour pattern focuses on never being satisfied with the status quo, with the viewpoint that even if things are satisfactory today, they will not be tomorrow. Customers and users will always be happier when the initiative is taken to make improvements, and there is always an improvement to be made, however small. Practitioners are encouraged to show leadership by spotting improvement opportunities and contributing to following up on them.

3.1.4 Help get customers' jobs done

This behaviour pattern represents the essence of every organization. All organizations have a range of stakeholders, but customers are typically the most important. This behaviour pattern focuses on the act of helping somebody solve their problems and become who they seek to become. It recognizes the psychographics of the customer: that is, how the customer wants to feel before, during, and after using digital and physical products and services. This involves making a bold, honest assertion outlining that the products and services an organization offers will provide a certain outcome for people who believe or want certain things, and are not designed for those who do not believe or want those things.

3.1.5 Commit to continual learning

This behaviour pattern underpins the others. Ignorance is the root cause of many organizational problems, typically when somebody does not have the right information when they have to act, or even when they are initially creating a system. It can also be an issue when it is mistakenly believed that situations are knowable and can be predicted. This links back to the behaviour pattern of accepting ambiguity and uncertainty. It is important that practitioners commit themselves to continually learning and improving their knowledge and the level of information they have. Data-driven experiments can be used to challenge and improve hypotheses. 'Trust and be trusted' is an important precondition for experimentation. Short feedback loops are another key to continual learning.

The ITIL story: High-velocity IT culture

Radhika: *We engaged with customers to better understand their current and future travel requirements. We compared the results of this analysis with our existing service offering. One request customers kept making was for better digitization of our services.*

Henri: *The board of governors has agreed to invest in improvements to our app which will make it easier for our customers to book cars on a wider range of smartphones and devices. We are also looking at adding several other features to our app to improve the experience of our users.*

Su: *First, we will develop the improvements to the smartphone booking functionality. This will be followed by supplementary features, such as an affiliate programme, a membership programme, automatic vehicle upgrades for regular customers, and streamlined priority bookings. Each of these concepts will be tested before we begin development.*

Solmaz: *We want to pursue a high-velocity approach to developing the new features for our app. We can support this by making sure we have the right culture in place.*

3.2 Models and concepts of HVIT culture

There are a number of models and concepts that inform organizational culture. These are grouped into three categories:

- purpose
- people
- progress.

The models and concepts are outlined in the following sections. Table 3.1 outlines the key behaviour patterns to which each of the models and concepts, and each of the ITIL guiding principles, relates.

Table 3.1 Models and concepts and related key behaviour patterns

	Accept ambiguity and uncertainty	Trust and be trusted	Continually raise the bar	Help get customers' jobs done	Commit to continual learning
Purpose					
Ethics	✔	✔	✔	✔	✔
Design thinking				✔	
People					
Reconstructing for service agility			✔		
Safety culture		✔			
Stress prevention		✔			
Progress					
Working in complex environments	✔				
Lean culture		✔	✔		✔
ITIL continual improvement model	✔		✔		✔

	Accept ambiguity and uncertainty	Trust and be trusted	Continually raise the bar	Help get customers' jobs done	Commit to continual learning
ITIL guiding principles					
Focus on value				✔	
Start where you are	✔				
Progress iteratively with feedback	✔				✔
Collaborate and promote visibility		✔			
Think and work holistically	✔				
Keep it simple and practical			✔		
Optimize and automate			✔		

The ITIL story: Models and concepts of HVIT culture

Solmaz: *We can adopt a number of models and concepts to help us achieve a high-velocity way of working as we develop the new app functionality.*

3.2.1 Purpose

Purpose is the external goal that drives people's efforts: that is, the changes that products and services enable. Models and concepts in the purpose category include:

- **Ethics** A system of principles that defines what is good for individuals and society.
- **Design thinking** The set of cognitive and practical techniques by which design concepts are developed.

3.2.1.1 Ethics

The societal, political, and economic impact of IT is unprecedented, for better and for worse. Digital organizations, where IT drives the business rather than just supports it, therefore have an increasingly strong moral obligation to consider how they apply IT, beyond their direct economic interests.

Software engineering often becomes a means of social engineering, and this is not always intentional. Machine learning algorithms are susceptible to take a training data set and amplify the worst of human biases. Organizations have a responsibility to behave ethically in order to understand and correct such biases.

Definition: Ethics

A system of principles that defines what is good for individuals and society.

Why ethics is important

The creation of the internet and near-instant information transfer between previously unconnected people is already having a profound effect on human society. Its full impact is not yet known, or fully knowable. Just like scientists or engineers, IT practitioners should accept moral responsibility for what they create. This means that close and detailed attention must be paid to ethics and morality, in day-to-day work as well as in the big picture.

A compounding problem is that the world is becoming increasingly complex. The systems that people live and work in have many connections, some of which are known, some knowable and some unknowable, and any intervention will have unintended consequences. Sometimes small things have large consequences. In the modern hyperconnected age, things can get out of hand very quickly. A number of considerations can be taken into account when promoting ethical behaviour in the workplace.

Digital technology allows for wider and deeper information about people. An individual may therefore be presented with detailed information about others. This can change people's perception, and perhaps behavioural responses, in business and personal relationships. Professional competence includes the extent to which a person is able to apply their own morals in combination with the ethics espoused by their organization or society. Forcing ethics on an employee through a policy only is not effective, as ethics are deemed personal.

Education

Ethics is a much-neglected aspect of software engineering education. It is not enough to provide post-core training: ethics must be part of the education that precedes this. Measuring and monitoring attitudes related to ethical behaviour is as important as creating rules and managing compliance.

Practitioners can be educated in ethics and encouraged to participate in workshops that make people aware of ethical consequences. It is also critical that, within organizations, scenarios are explored and attitudes to ethics are measured. Recruitment policies and practices should focus on a wider understanding of ethics. The ability to see the wider picture, to think in abstractions, and to imagine consequences should be seen as a key skill required at all stages of the product lifecycle. At a micro-level, ethics could even be included in retrospectives and lessons-learned programmes. Ironically, Agile methods that focus on clearing backlogs often neglect this. The individual work units are often performed independently of the context of the wider system and people's ability and willingness to imagine the potential consequences of their actions. The ITIL guiding principle of thinking and working holistically is intended to prevent such situations.

The objective here is to create an ethical culture in the IT industry as much as a culture focused on quality and user-centricity.

Organizational design

The way organizations are designed can also be altered to increase trust and interaction between team members, following the principle of 'collaborate and promote visibility'. This requires a cognitively and culturally diverse workforce that is tightly connected, with high degrees of visibility within tight networks but not necessarily throughout the whole system. Organizational design should therefore be examined with a view to building long-term trust rather than short-term gain, and seeking to change social interactions in the network to increase interdependency by working together.

Not only does this help to solve problems, it also creates a decision support ecology built on trust. If you know your behaviour is observed by those whose respect you seek, this changes things for the better. It builds an ecology of trusted interaction in which ethical questions or dilemmas are less likely to surface.

Habits and the role of failure

Habits are more powerful than rules, and they create patterns of behaviour and responses that do not involve analysis and decision-making. Education and organizational design need to create circumstances in which the habits of virtue emerge naturally. In these circumstances, social interaction excludes behaviour that is unethical. Abstract knowledge of what is right is combined with the day-to-day habits of behaviour, which means that virtue comes naturally to what we do.

Habits are often best formed by physical interaction and by tolerated failure. Abstraction and the use of metaphor in learning allow for greater adaptability and contextual application under conditions of uncertainty. More is learned through failure than through success, but only in a no-blame environment.

Ethics and artificial intelligence

Much work is being done in the area of not only emulating human intelligence in technology but also incorporating emotional intelligence into self-learning technology, also known as AI.

AI is a key concern of ethics in the digital era. The term 'techno-ethics', originally coined in the 1970s, has recently re-emerged when talking about ethics and AI. Techno-ethics is the responsible use of science, technology, and ethics in a society shaped by technology. As the fourth industrial revolution develops, organizations need to expand their ethical agenda to include aspects of 'bio-tech'.

ChatOps is a good example of this, which includes emotional intelligence as a focus to improve operations. ChatOps tries to ascertain the customer's emotional state, in order to respond appropriately and understand and manage their emotional state while dealing with their incident or request. Other examples can be found in voice assistants, image analysis software, search engines, speech and face recognition systems, and so on.

The use of AI or any area of new technology continues to evolve as more is learned about its characteristics through use. However, it is necessary to develop emotional intelligence in people before making any attempt to design and use it in technology.

Definition: Emotional intelligence

The ability to understand the way people feel and react, and to use this skill to make good judgements and to avoid or solve conflicts.

Techno-ethics topics for discussion

It is recommended to discuss ethics and their application with management and co-workers. Examples of topics that should be considered include:

- Who within an organization decides what is moral or which principles to follow?
- Which principles are expected to govern ethical behaviour? These could include integrity, honesty, respect, personal responsibility, compassion, and dependability.
- What do these principles mean?

Examples of ethical questions

A common situation with ethical concerns is the use of browser cookies that track a user's pattern of use and therefore enable the business to 'pop up' suggestions for other services or products that match the customer's browsing history. It is important to consider the ethical and social issues involved in this situation; and whether this should be regarded as common practice, needing no further thought, or could have long-term implications for personal privacy or even breach the law.

HR departments are now using a candidate screening method that registers the emotions of candidates while being interviewed and cross-references this with their systems, checking the internet for their social media posts to analyse behaviour responses. No psychologist is involved in the assessment of the specific data gathered; it instead relies on the analytics of the software, which may have included psychology in its design. Are you comfortable if this is how you will be judged for future roles (internal or external)? Is this a technology you would like to have in helping you choose your team?

A service desk analyst has received a request from someone to access a colleague's system so that they can continue with necessary work while that colleague is away. The normal response is 'no', because most organizations have a password policy that denies anyone but the owner access to their system. However, there is another response, as IT does not exist to prevent the business or individual from carrying out work. The simple answer is 'Yes, of course. All that is required is for the authority form to be filled in and signed.' Both parties in this situation are applying appropriate business privacy ethics; however, the person accessing their colleague's system is still under their own recognisance while in that system. Something seemingly innocent can lead to incredible damage if either person's ethics have been compromised. Trust is a primary facet of ethics.

Key message

When applying ethics, practitioners should aspire to the following behaviours:

- Think about how their actions affect others.
- Establish generic ethical principles.
- Accept that ethical principles simply help to clarify specific situations.
- Discuss dilemmas.
- Take responsibility for choosing the least worse course of action.

Figure 3.2 highlights the key behavioural patterns that are relevant to ethics.

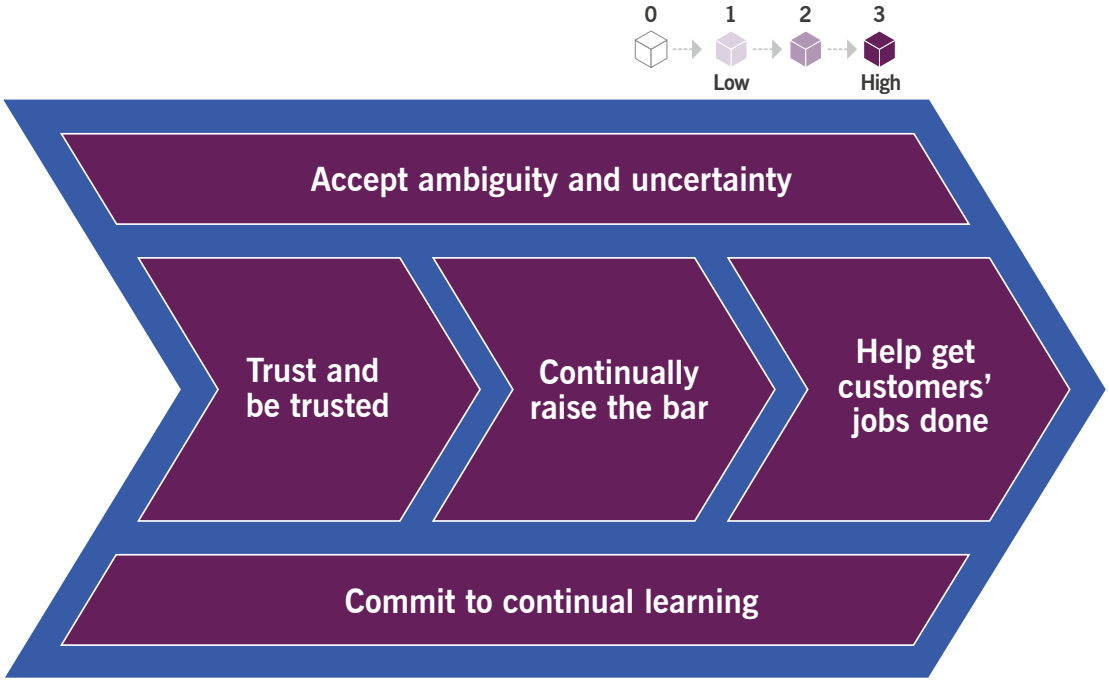

Figure 3.2 Heat map of the importance of the key behaviour patterns to ethics

The ITIL story: Ethics

Su: *Every decision we make has an ethical component, from the company we work for to the services we use. Axle Car Hire's vision includes an emphasis on environmental sustainability. We know the world's resources are finite, and we believe that, ethically, we are obliged to minimize our footprint when we use these limited resources. Furthermore, many of our customers choose our services because of our green initiatives.*

Henri: *Axle's commitment to ethical business behaviour is not limited to sustainability. It includes the way we treat our employees, select and deal with suppliers and partners, and contribute to society. An ethical stance can be good for business, as well as being the right thing to do. Customers will give their business to organizations that reflect their values.*

3.2.1.2 Design thinking

Design is about more than just goods. User experiences, processes, and systems are also designed by applying an approach that has come to be known as 'design thinking'. Design thinking is just how designers think. It is continually evolving, and has recently attracted more attention and interest as digitally enabled organizations have begun looking to design thinking to improve market perception of their digital products.

Designers create products with a particular intent and a specific purpose, to transform a situation that is not convenient or optimal into one that is. They think about ways to change these unfavourable conditions into ones that are more preferable, making them more fit for purpose and fit for use.

Definition: Design thinking

The set of cognitive and practical processes by which design concepts are developed.

For digitally enabled organizations, the quality of their digital products and customer experiences is immensely important. An understanding of design thinking gives practitioners, together with co-workers and customers, the ability to contribute to the creation of better digital products and customer experiences, and help to complete jobs for customers effectively. Design thinking is particularly useful when addressing 'wicked problems': in other words, problems that have unarticulated needs and conflicting hypotheses.

Customers are not the only ones who use products. They are also used by service providers, sales people, managers, developers, operators, and other stakeholders. A concern of design thinking is trying to solve the issue of balancing the interests of all of these stakeholders, as well as those of the organization itself. The consumer, in general, has the highest priority, as an organization is ultimately financed by its customers. Designers appreciate this and are therefore also concerned with the economic aspects of design

Key message

When applying design thinking, practitioners should aspire to the following behaviour:

- **Empathize with stakeholders** It is important to be able to understand the point of view and needs of stakeholders.
- **Speculate and experiment** Designers should be able to create hypotheses based on observation and reflection, and test them with prototypes.
- **Dare to decide** Designers should focus on what customers do and need, not necessarily on what they say they want. Designers should be empathic and able to feel what is most important, and make decisions accordingly.

Figure 3.3 highlights the key behavioural patterns that are relevant to design thinking.

The ITIL story: Design thinking

Marco: *When designing the new functionality for our app, we tried to consider everything that would have an impact on our customers. We took into account the customer experience inherent in smartphones, the processes which underpin app functionality, and the customer's interactions with our systems as they order, collect a car, and travel. Each component of the service combines with the others to engage the customer, and is part of the whole customer journey.*

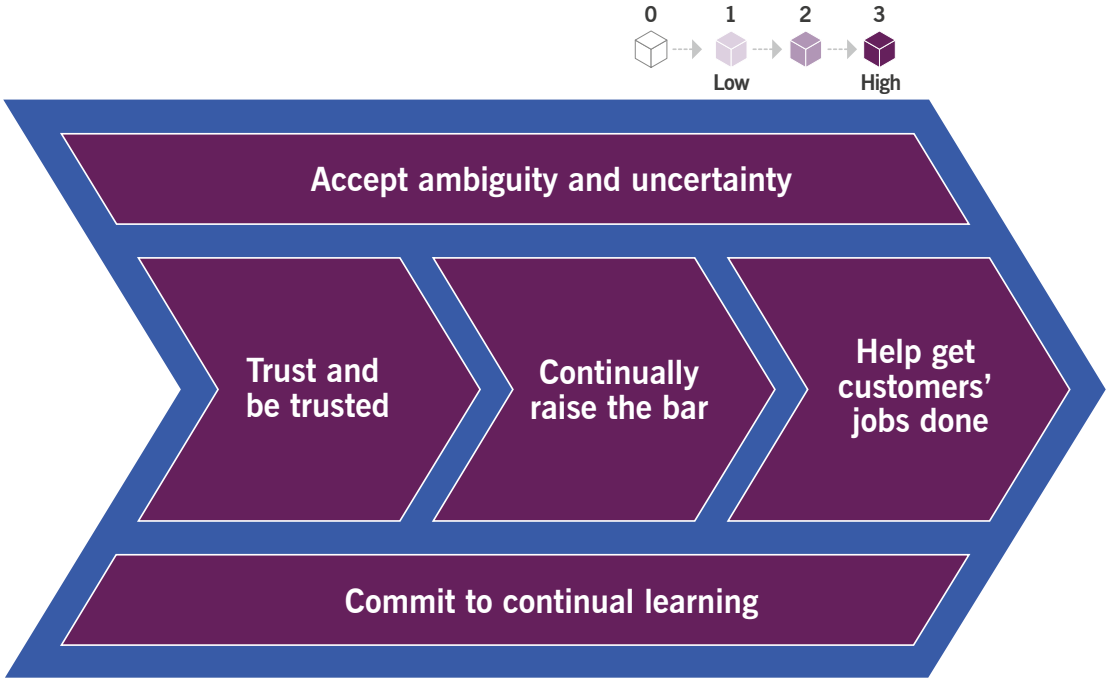

Figure 3.3 Heat map of the importance of the key behaviour patterns to design thinking

3.2.2 People

The models and concepts in the people category are about organizing work that produces or relies on knowledge, and fostering a healthy and productive workplace. Models and concepts in the people category include:

- **Reconstructing for service agility** An approach to organizing knowledge work and service provision that reflects its complex and social nature.
- **Safety culture** Fostering a climate in which people are comfortable being and expressing themselves.
- **Stress prevention** Preventing, monitoring, and remediating unhealthy tension in the workplace.

3.2.2.1 Reconstructing for service agility

HVIT ways of working have serious implications for an organization and the way it manages its services. Management is traditionally focused on specialization of work, prescriptive processes, and targets to drive and measure performance, based on the assumption that resources and work are predictable enough to be controlled at a low level of granularity.

Within knowledge work and service provision, however, there are many subtle human interactions that make it impossible to predict the outcomes of a service interaction. This makes it pointless to impose detailed targets on humans as if they were machines. People are intrinsically unpredictable, and will respond to the perceived intent of a co-worker or customer, who will then respond in return. Sometimes the intent is incorrectly perceived, leading to an unexpected response. Both parties must therefore constantly assess their responses and adjust accordingly.

Where humans are involved, service interactions are social interactions. A typical management approach does not reflect the social reality of a service workplace, and is therefore ineffective. A better approach[3] is to loosen control so that people have more freedom to use their professional judgement in intrinsically unpredictable circumstances. This is central to the idea of reconstructing the organization for service agility.

Definition: Reconstructing for service agility

An approach to organizing knowledge work and service provision that reflects its complex and social nature.

When following this approach, it is desirable to keep some degree of process in place, as people generally like to know more or less what to expect and what is expected of them. Some control should also be kept in the form of constraints related to organizational structure.

Within these guidelines and boundaries, however, it is important to treat people as human beings. Service outputs and outcomes can be ruined by the smallest details of human interaction, not only with customers but also with co-workers. When detailed processes and other constraints encourage people to act as predictably as algorithms, things will invariably go wrong, as people will naturally act more intuitively and unpredictably. People tend to exhibit intelligent disobedience, and this should be encouraged, rather than punished, so that the right thing is done for the customer.

The degree of control that should be maintained depends largely on the predictability, and therefore the controllability, of the work being performed. For instance, processes can be put in place for work that is standardized or algorithmic, and has known or knowable requirements, priorities, approaches, and resources. The provision of a standard laptop to a new employee is a good example of this type of work. Work passes between specialized workstations according to a predetermined route, and each workstation (approval, preparation, logistics) is oblivious to what other workstations have done or will do; but, by following standard procedures, the work will get done.

This type of work, where a person follows a pre-defined process driven by a set of established instructions along a consistent pathway, is described as algorithmic. Heuristic work, on the other hand, is centred on people devising ideas and approaches, and creating hypotheses and experiments until a solution is found. A good example of this is case management, which is used in many sectors such as emergency services, medicine, social work, law, and policing. Each situation or 'case' has to be dealt with differently by the case manager because the situations do not fit a predetermined process. Algorithmic work will always produce the 'correct' result that it is designed to achieve, whereas heuristic work will usually be successful, but may lead to a variety of results.

When pursuing a heuristic and less pre-defined approach to work, processes and other forms of control are sometimes used temporarily to provide support until there is enough competence and confidence to work heuristically with fewer constraints.

The genesis of the DevOps movement is a good example of a bottom-up transformation, driven by practitioner values. It started with the initiative of a few professionals who formed a group, which grew into a conference and eventually into a global movement, well known outside of the industry.

This is a good example of change that was initiated and driven by people's values and their desires to make things better, and have fun in the process. There was no top-down direction or control. In fact, much of this happened below the corporate radar, meaning that people had (and claimed) enough freedom to act without corporate constraints.

Personal values

People make their best contributions to work that is aligned with their personal values, and this should always be a prime consideration for organizations when recruiting new employees. This does not necessarily mean that

employees should have the same set of values as their organization. This is not productive, as coherent diversity and functional tension are beneficial to the evolution of an organization.

It is important that practitioners be aware of their team members' values. Differences in values can explain different behaviours, and being aware of these can trigger reflection and recalibration of individual or collective mental models.

It is both naïve and morally dubious to try to change other people's values. Values emerge over time as a result of interactions in a particular social context; they are often difficult to articulate, and have to be experienced to be understood.

Key message

When reconstructing for service agility, practitioners should aspire to the following behaviour:

- **Loosen control** Give people more freedom to use their professional judgement in intrinsically unpredictable circumstances.
- **Use process as temporary scaffolding to provide support** Processes and other forms of control can provide support until there is enough competence and confidence to work heuristically with fewer constraints.
- **Respect team members' values and allow them to work within these** People make their best contributions to work that is aligned with their personal values.

Figure 3.4 highlights the key behavioural patterns that are relevant to reconstructing for service agility.

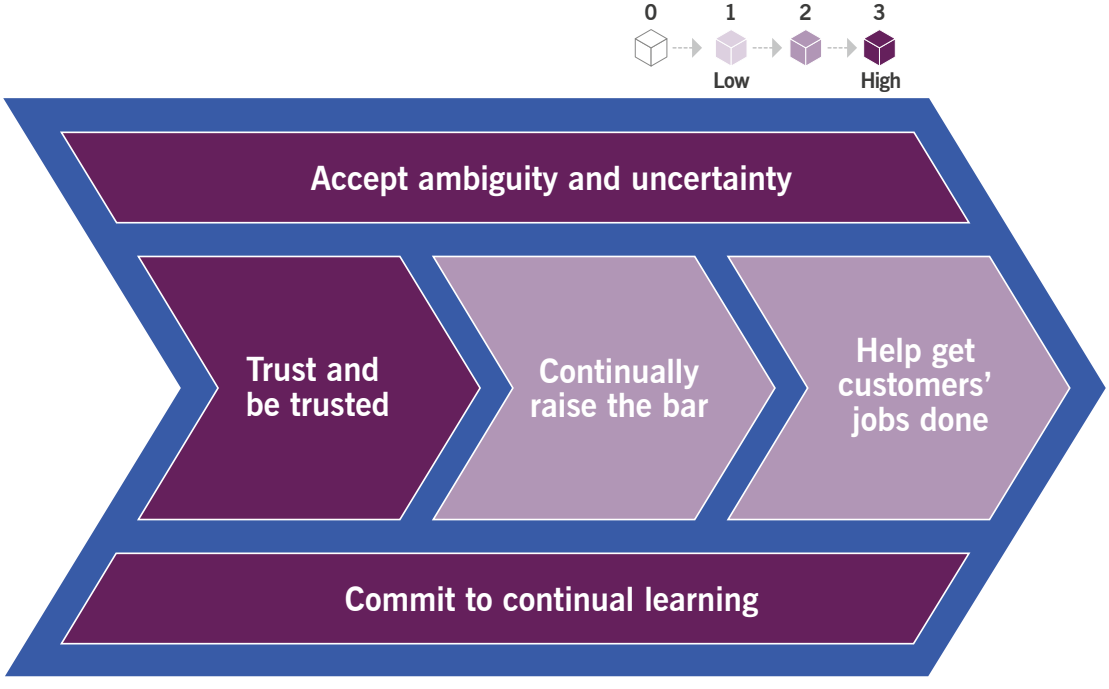

Figure 3.4 Heat map of the importance of the key behaviour patterns to reconstructing for service agility

The ITIL story: Reconstructing for service agility

Solmaz: *At Axle, we have a framework to guide our work, and the team working on the development of the new app functionality follows this, along with Axle's principles and policies. The framework has been designed to be flexible enough to allow the team to adapt to changing circumstances and the experimental nature of their work. This makes it easier for them to try new approaches and methods when developing the new app functionality, which means they have more freedom to experiment.*

3.2.2.2 Safety culture

Digitally enabled organizations are under particular pressure to continually improve their performance under changeable market conditions. Technological progress means that their products have increasingly significant economic, societal, and political impact, and therefore organizational failure can often lead to disastrous consequences. The challenge for these organizations is to foster an effective set of shared beliefs, perceptions, and values in relation to risks. This in turn will create a safety culture, and result in behaviour that is beneficial for all stakeholders, including the workforce.

Definition: Safety culture

A climate in which people are comfortable being (and expressing) themselves.[4]

In a safety culture, people feel trusted and valued. They are therefore more likely to point out risks than when they fear that this would damage their reputation and position. A good safety culture can be promoted by the commitment of senior management to safety, realistic practices for handling hazards, continual organizational learning, and care and concern for hazards shared across the workforce. A 'hazard', in this case, should be understood in a wider sense than just physical dangers; it includes environmental factors, technology failures, human errors, and process flows.

The complex systems that are often present in HVIT environments are inherently hazardous. These systems always contain multiple flaws and therefore latent issues. Continual changes to the system and its environment mean that the flaws also continually change. Most of these flaws are too small to cause significant issues, either because of redundancy and other forms of resilience in the system, or due to human intervention. When significant issues happen, it is usually due to an unpredictable combination of several flaws, and therefore not a single root cause. Dealing with such issues requires knowledgeable and skilful staff, and favourable working conditions. These conditions are most critical when things get stressful, because at these times, people revert to survival mode based on deeply embedded assumptions about organizational culture. It is therefore crucial that things like not blaming people and treating failures as improvement opportunities are more than espoused corporate values. In HVIT environments, it is crucial that people feel able to share their opinions and experiment with improvement without the fear of judgement or embarrassment.

The inherently hazardous nature of complex systems

Richard Cook[5] provided insight into the inherently hazardous nature of complex systems in his study of how such systems fail. His research focused on medical devices, which are sometimes critical to patients' lives. Complexity science has evolved since his article was published, and so has its terminology, but many of the following statements have been proven and further developed in the context of various industries, including IT:

- Complex systems are intrinsically hazardous systems.
- Complex systems are heavily and successfully defended against failure.
- Catastrophe requires multiple failures – single-point failures are not enough.
- Complex systems contain changing mixtures of failures latent within them.
- Complex systems run in degraded mode.
- Catastrophe is always just around the corner.
- Post-accident attribution to a 'root cause' is fundamentally wrong.
- Hindsight biases post-accident assessments of human performance.
- Human operators have dual roles: as producers and as defenders against failure.
- All practitioner actions are gambles.
- Actions at the sharp end resolve all ambiguity.
- Human practitioners are the adaptable element of complex systems.
- Human expertise in complex systems is constantly changing.
- Change introduces new forms of failure.
- Views of 'cause' limit the effectiveness of defences against future events.
- Safety is a characteristic of systems and not of their components.
- People continually create safety.
- Failure-free operations require experience with failure.

Key message

When applying safety culture, practitioners should aspire to the following behaviour:

- **Act on safety requirements** Don't just talk about why safety is important: do something about it.
- **Exhibit vulnerability** Don't be afraid to speak up about any doubts, and ask for help when needed.
- **Foster feedback and act on it.**
- **Be kind and compassionate** Build human relationships.
- **Be realistic about failure** Acknowledge that failure will happen and that people are not to blame, but the system.

Figure 3.5 highlights the key behavioural patterns that are relevant to safety culture.

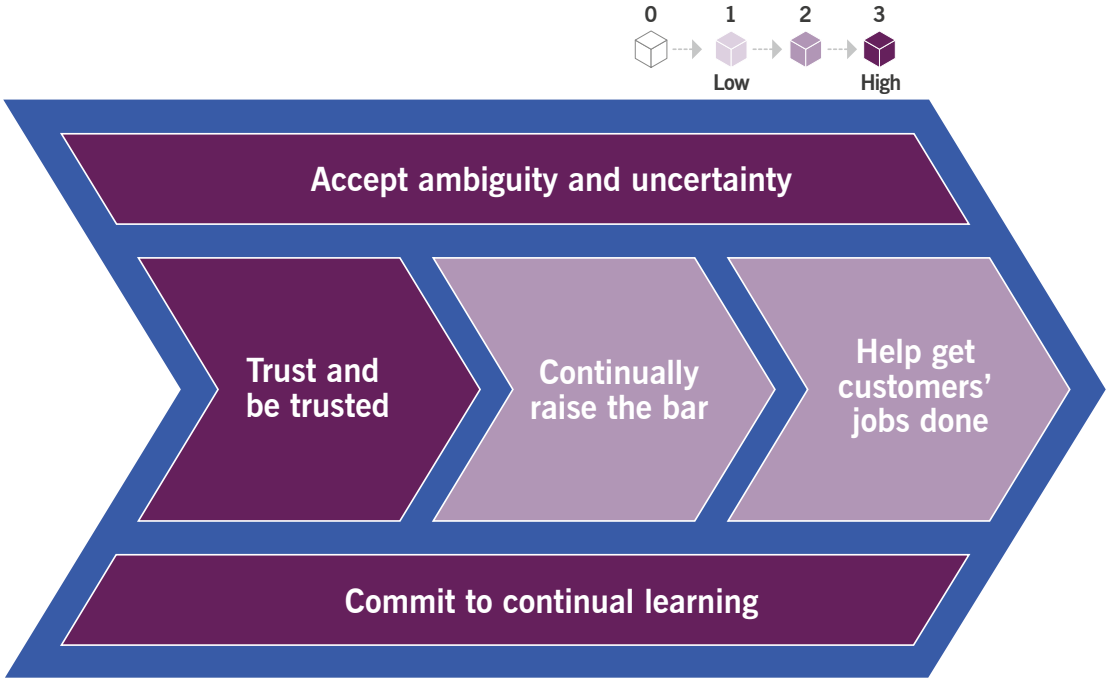

Figure 3.5 Heat map of the importance of the key behaviour patterns to safety culture

The ITIL story: Safety culture

Su: *The development of the new app functionality may require experimentation and sometimes an acceptance of failure or an unexpected change. If the members of our team feel safe and valued, they will be in a better position to produce their best work.*

Solmaz: *Creating a culture and environment where each team member feels comfortable when demonstrating their work, expressing doubt, or asking for feedback means we can rapidly move forward with designing and releasing further updates for the app, and reduces the chance of issues going unchallenged or good ideas being ignored.*

3.2.2.3 Stress prevention

Definition: Stress prevention

The prevention, monitoring, and remediation of unhealthy tension in the workplace.

Many people suffer from stress, burnout, and mental health issues. Although physical health issues have been recognized and talked about at work for a long time, discussion of mental health issues has only seriously begun more recently. It is now, however, becoming acceptable as well as desirable to address such issues.

Mental health concerns are not exclusive to HVIT or IT in general. They are, however, critical for both high organizational performance and the well-being of employees.

Some jobs are more intrinsically stressful than others. Development work, for instance, can be stressful because of the pressure to hit deadlines, whereas in operations work, stress can come from the constant pressure to respond to a never-ending stream of calls, or waiting for a pager to buzz.

No workplace is always free of stress. Things happen unexpectedly, and the right balance has to be struck between the interests of all stakeholders, including employees. It is therefore beneficial to help employees develop basic life skills to negotiate and survive temporary work pressure, inadequate tools, misunderstandings with managers and other co-workers, and other stressful situations.

People are key both to the effective delivery of IT and service management and to successful transformations. Often, however, the response to this is to treat people as a problem that needs to be solved, rather than addressing the concerns of those impacted. As part of good IT corporate governance, it is vital to understand the human impact that work has on all involved. This has to be managed effectively to generate an enthused and healthy workforce.

A healthy work–life balance should be seen as a basic requirement of an organization. This is particularly relevant now, as the hyperconnected nature of work tends to blend work and life, and the introduction of new technologies makes it far easier for people to be contacted and pick up work outside of normal working hours. The absence of such a healthy work–life balance cannot, in the long term, be compensated for in other ways.

As delivery models adopt high-velocity approaches, organizations need to change the way that they create and maintain products and services. In some cases, this may require new technologies; but in all cases, there will be a significant shift in culture, performance measurement, and other human elements; hence it becomes even more important to consider the human impact. This applies both during the transformation of the model and once the model has become the norm. These two stages present very different but equally serious challenges that many organizations are yet to address.

Stress during change

When fundamental ways of working are altered, what was previously seen as good behaviour can be undermined, and turned into behaviour that is less desirable. Managers may respond to this challenge by trying to get people to change those behaviours. Although this is important, they should also understand the need to help people through the process of change, and that the way individuals process this level of change internally may differ significantly from one person to another.

This is especially true of those who thrive on working in highly structured ways, or whose roles do not require frequent human contact, or who have latent mental health issues.[6] The organization's duty of care to them is not removed by the shift to new operating models; nor is the value they can bring to the organization diminished.

Stress during business as usual

When a high-velocity approach has been adopted, and becomes the regular, embedded way of working in an organization, new forms of stress can be generated that are situational but long term. The pressure to constantly deliver can be exhausting and demoralizing. The threat of AI replacing jobs is ever-present, and the delivery cycle can mean there are few chances to collectively acknowledge and celebrate success. Where high velocity is also dependent on on-call support at all times, and the need to work unconventional hours, the impact on domestic life can also be significant.

Approaches

Failing to properly manage these issues can lead to significant consequences for both individuals and the organization, and so it is important that they are properly addressed. There are several approaches that can be taken to preventing stress in the workplace.

Lean and Agile approaches encourage limiting work in progress and pulling work in rather than having it pushed from other areas. This not only improves throughput, but also reduces overburden and potential stress.

CI/CD techniques allow smaller, more frequent, and more reliable deployments. This reduces the potential negative impact on operational systems, enabling deployment to be executed during business hours rather than in evenings and weekends. This, in turn, helps to reduce overburden and potential stress from having to work outside of regular working hours.

HR departments can play a crucial role in reducing workplace stress, and should be involved in developing strategic, tactical, and operational approaches towards this goal, including a detailed risk assessment. The assistance of mental health workers should be actively sought to develop these approaches. An educational programme within the workplace is essential to ensuring that people are aware of the signs of mental health issues in both themselves and others, and of how to correctly respond to them. This programme should extend to clearly understood actions that will protect individuals who suffer acute issues in the workplace.

Assistance should be provided to those with invisible disabilities, to ensure they can effectively interface with high-velocity teams and projects while retaining a sense of personal control over their environment. This might mean, for example, providing alternative options to a physical presence in Scrums, ensuring they have a liaison point who understands their preferred ways of working, and taking account of their needs as part of planning processes.

Shifts in performance management systems should be balanced, and retain the flexibility to reward behaviours that are specific to an individual's profile when required. When teams are meeting multiple high-pressure delivery requirements, they may see a decrease in stress from being able to self-organize and retain local management of prioritization and resourcing.

Whichever option is taken, it should be made clear to employees that these approaches are non-judgemental, and their prime concern must be the long-term welfare of individuals. Once the approaches have been put in place, it is vital that they are not subverted to meet other objectives.

Continual vigilance

Once one or more approaches to stress prevention have been put in place, it is vital to constantly scan for potential new issues and new guidance, particularly in a rapidly changing IT and workplace environment. HVIT work is a relatively new concept, which means that people will inevitably be facing new challenges and stresses. The increasing emphasis on stress prevention and mental health in the workplace will also give rise to new ways of managing these issues, which could lead to previously accepted advice becoming obsolete as a result of new evidence and clinical studies. It is therefore important to keep mental health and stress prevention approaches up to date.

Figure 3.6 highlights the key behavioural patterns that are relevant to stress prevention.

Key message

When looking to prevent stress in the workplace, practitioners should aspire to the following behaviour:

- Elicit advice from mental health professionals.
- Conduct a risk assessment of mental health issues.
- Develop strategic, operational, and non-judgemental approaches to stress and mental health.
- Ensure that the approaches adopted are only used for the welfare of individuals, not subverted for other objectives.
- Help those with invisible disabilities to interface with HVIT teams effectively.

The ITIL story: Stress prevention

Henri: *We are asking our team to make improvements to our app at a fast pace, which can be stressful.*

Marco: *The launch date for a new piece of functionality can be particularly stressful for a development team.*

Radhika: *We can help to reduce stress for our employees by making sure our HR department is pursuing the correct approaches to support good mental health and stress prevention.*

Henri: *We can also provide educational talks to our employees to make them aware of the dangers of excessive levels of stress, and what they should do if they are having stress-related problems.*

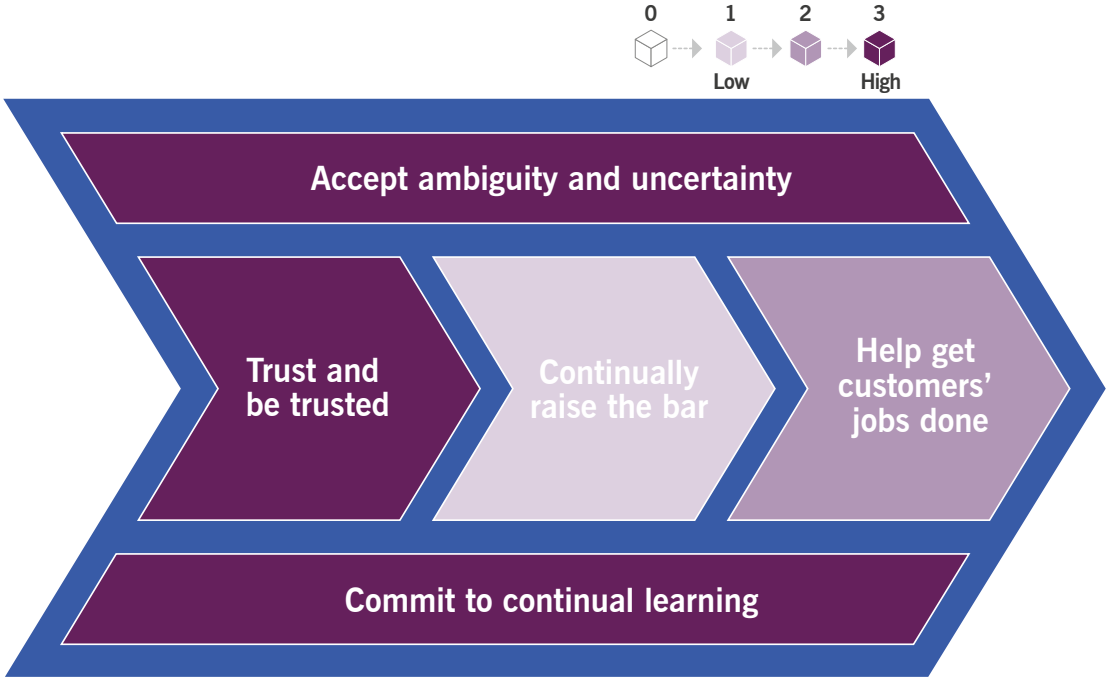

Figure 3.6 Heat map of the importance of the key behaviour patterns to stress prevention

3.2.3 Progress

The models and concepts in the progress category are about understanding the complex nature of work, and improving by learning. Models and concepts in this category include:

- **Working in complex environments** An approach to the management of work that applies different heuristics to decision-making based on complexity thinking.
- **Lean culture** A way of working that is foundational for continual experimenting, learning, and improvement.
- **The ITIL continual improvement model** A model that provides organizations with a structured approach to implementing improvements.

3.2.3.1 Working in complex environments

In order to work effectively, practitioners must understand the nature of the environment, or system, in which they work. A major characteristic of a system is its predictability, as this determines which way of working is effective. Systems thinking in general, and complexity thinking in particular, help make sense of the system and offer guidance as to effective approaches.

From a practical point of view, systems thinking is about seeing the results of our actions in a larger context. A system is regarded as a set of parts that, when combined, have qualities that are not present in any of the parts themselves. Systems thinking is a field that is still developing, and one of the more recent developments is the application of systems theory to complex systems and complex adaptive systems.

Definitions

- **Systems thinking** A holistic approach to analysis and decision-making that focuses on the relationship between a system's components and the way the system works, both as a whole and within the context of larger systems.
- **Complexity thinking** A systems thinking approach based on the recognition and understanding of the various levels of complexity inherent in the systems and the context in which they operate.
- **Complex adaptive systems** Systems that adapt in, and co-evolve with, a changing environment, resulting in:
  - behaviour that is not predicted by the behaviour of parts of the system
  - the inability to examine the system in isolation from the other systems in its environment.

The study of complex adaptive systems is highly interdisciplinary and blends insights from the natural and social sciences. 'Complex' means that the behaviour of the whole is not predicted by the behaviour of the parts, and things can happen organically and unpredictably. In other words, behaviour is emergent. 'Adaptive' means that the systems have the capacity to change and learn from experience.

Complex adaptive systems exhibit behaviour that cannot be predicted, but can often be explained retrospectively. Some systems are unpredictable because their boundaries only partially constrain the agents that act within the system, and the agents modify the boundaries.

Examples of complex adaptive systems include the stock market, social insect and ant colonies, the biosphere, the brain and the immune system, bacteria, cities, manufacturing businesses, and any human social group-based endeavour in a cultural and social system, such as political parties or communities.

A heuristic way of working is based on exploration and experimentation that favours speed over completeness, accuracy, or precision.[7] Heuristics usually work, but results may vary. Complexity-based heuristics are therefore explorative and experimental ways of working with systems that are, by definition, too unpredictable for algorithmic work.

Complex adaptive (organizational) systems are frequently found in HVIT environments. The inherent unpredictability of these systems presents a challenge to people who are used to working with predetermined processes, detailed analyses, and project plans that are based on knowledge or knowability of the work. In complex systems, behaviour emerges as a result of unforeseeable interactions between many agents and the boundaries that constrain and therefore affect them. Boundaries can also be affected by the agents, increasing the unpredictability.

In these circumstances, an experimental and non-linear approach is needed. Multiple experiments should be performed in parallel (so they will not influence each other), and what works and what doesn't should then be observed. Practitioners should be prepared to deal with negative side-effects of 'failed' experiments, and to continue and broaden successful experiments, bearing in mind that circumstances and results will change.

Most organizations will experience a variety of work contexts, some being more predictable than others. It is therefore important to be able to understand the nature of the particular work being executed, and to act accordingly. The Cynefin[8] sense-making framework (see Figure 3.7) offers a practical way of assessing complexity and determining appropriate courses of action. It distinguishes between five domains or contexts that characterize the relationship between cause and effect:

- **Obvious** Clear causality, where predetermined best practice should be applied.
- **Complicated** Unclear but knowable causality that can be determined by analysis or expertise, followed by good practice.
- **Complex** Unclear and unknowable causality requiring safe-to-fail experimentation (emergent practice).

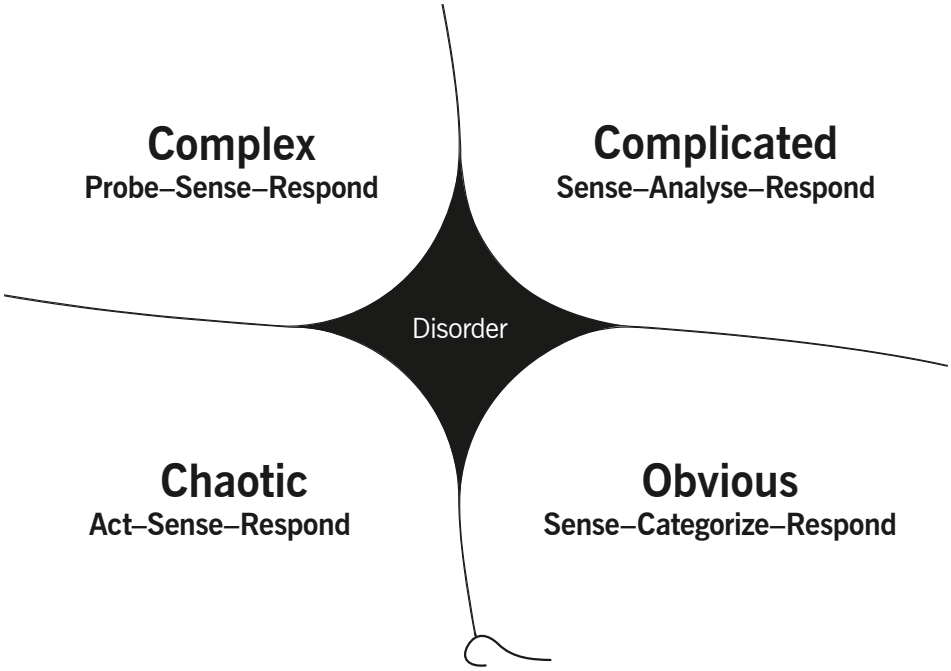

Figure 3.7 The Cynefin framework
After Snowden (2011); reproduced by permission of Cognitive Edge

- **Chaotic** A more extreme form of complexity that demands immediate action to transition the situation to complex (novel practice).
- **Disorder** The state of not knowing in which of the other domains you are, with a bias to assume that the domain corresponds to the context in which you are most experienced.

Cynefin is also described as a 'liminal' system, because rather than having completely solid boundaries between each domain, the 'edges' between order, complexity, and chaos tend to be thought of as a phase shift. Liminal boundaries are transitional: an issue may be in a state of tension between adjacent domains.

The recognition of complexity, and therefore unpredictability, as a legitimate state of affairs enables engineers to rethink their belief that rigid processes and plans are the organizationally correct way to respond to every situation.

Because a complex failure lacks a single, consistent path from a cause to an effect, multiple factors are likely to be causal, and it is likely that there will not be an exact precedent for them in previously addressed issues. Additionally, the evidence available might support conflicting theories about what is causing the issue. To troubleshoot in this circumstance, Cynefin defines a number of steps to take:

- Identify multiple hypotheses for what might be happening.
- In parallel, test each 'coherent' (that is, plausible) hypothesis using small, safe-to-fail experiments.
- Observe the impact of the experiments.
- Where positive outcomes are observed, attempt to amplify them. With negative outcomes, attempt to dampen their effect.

Unpredictability is not necessarily undesirable; this depends on the nature of the activities. In general, operations benefit from low variability, whereas research and development benefit from high variability. The lower the variability in the production and delivery processes, the more consistent the product quality, and the quicker and cheaper the production. Variability can be reduced by discovering and dealing with its underlying (root) causes.

There are also circumstances in which high variability is beneficial. In research and development where there is uncertainty about the potential of a product, this variability can be exploited by obtaining fast feedback, taking rational risks, and dynamically changing the product in response. A good example is the concept of the minimum viable product, in which a useful but not complete product is launched in order to assess market potential and to adjust the product accordingly. There is a parallel with stock options: the higher the variability, the higher the potential profit with a limited loss. The longer product development decisions can wait to be made, the more options there will be to increase the product's value.

In complex systems, failures are inevitable; preventing failures is impossible. There are, however, multiple ways of responding to failures. For example, antifragility is a qualification of complex adaptive systems that increase in capability, resilience, or robustness as a result of stress or failure. It is contrasted with fragility (failing), resilience (recovering from failure), and robustness (resisting failure).

This divergent way of thinking about systems is effective but comes at a price, as practitioners must continually assess whether they are taking the right approach. They should not be tempted by the superficial simplicity of 'best practice' that was reported and believed to have worked elsewhere, but make the effort to search for the best solution through diligent study.

Figure 3.8 highlights the key behavioural patterns that are relevant to working in complex environments.

Key message

When working in complex environments, practitioners should aspire to the following behaviour:

- Assess the causality, while being aware of natural bias towards the domain of dominant experience.
- Act according to the context; in Cynefin terms:
  - In the Obvious domain, apply constructs such as prescriptive processes and detailed waterfall-based plans.
  - In the Complicated domain, apply constructs such as case management[9] and timeboxing. When transitioning from Complex to Complicated, apply constructs such as Scrum.
  - In the Complex domain, apply 'pre-Scrum' techniques such as parallel/independent experimentation and disintermediation of the analyst to discover and resolve unarticulated or ambiguous user requirements.
  - In the Chaotic domain, take control and act quickly to stabilize the situation and transition it into the Complex domain.
- Beware of using 'recipes' based on cases that are presented as if they can be adopted without adaptation; understand the context and do not confuse correlation with causation (sometimes things do not repeat themselves in the same context); never repeat 'what you do' without understanding 'why you do it'; do not blindly copy industry leaders.
- Be careful when defining the future state and closing the gap; do not try to engineer human systems, but evolve them; develop the evolutionary potential of the present and move to the 'adjacent possible'.

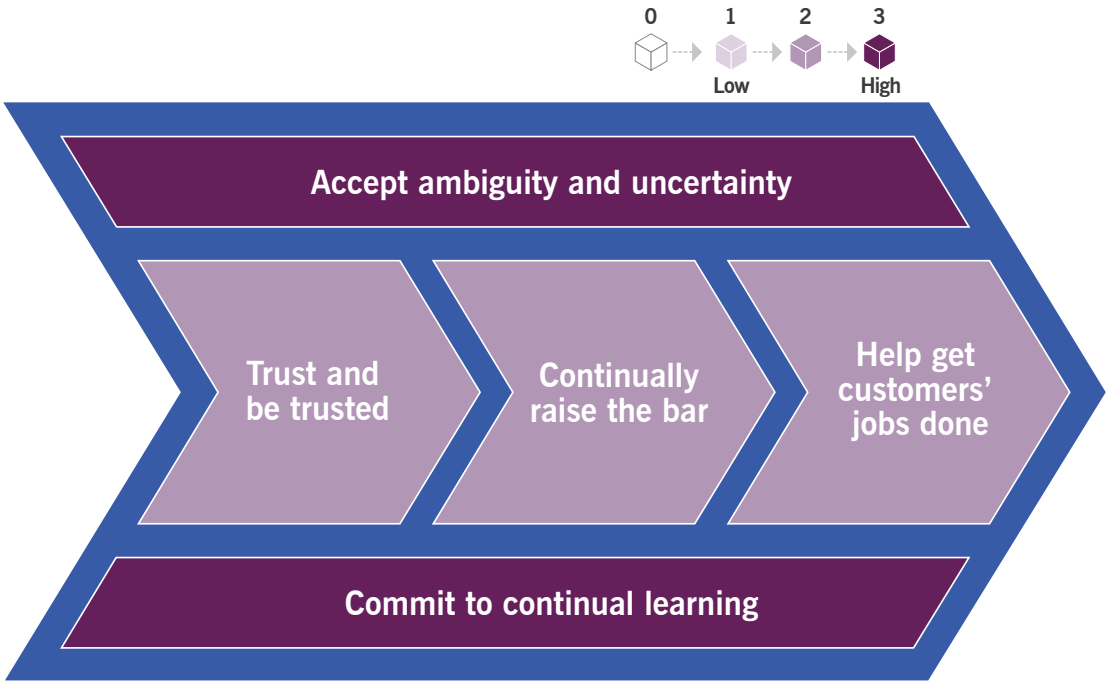

Figure 3.8 Heat map of the importance of the key behaviour patterns to working in complex environments

The ITIL story: Working in complex environments

Su: *As with any complex system, Axle Car Hire's business has many interacting components. These include internal and external teams and services, customers and their behaviours, and external factors such as traffic flow, road changes, and industrial actions that might impact our dependencies (for example, roadside assist or gasoline supplies).*

Marco: *As we develop the new functionality for our app, we must be mindful of how it might impact the wider areas of Axle and its business. For example, if we introduce automatic vehicle upgrades via the app, we need to be sure that our other systems and infrastructure are able to coordinate properly with this functionality.*

Su: *Should we need to make changes to the app to accommodate these other areas, we must be adaptive without losing control of our work.*

3.2.3.2 Lean culture

Definition: Lean culture

A work environment where trust, respect, curiosity, enquiry, playfulness, and intensity all co-exist to support learning and discovery.

Lean is a balance between striving for standardization and predictability to avoid errors, and at the same time fostering a culture of calculated risk-taking, curiosity, and enquiry, based on a foundation of trust and respect.

To be effective, resilient, and adaptable, an HVIT environment must be built on a solid foundation of Lean culture.

Leaders play a key role in setting the social norms and expectations of teams and their interaction with other teams. Lean culture boils down to the social patterns that motivate people and inspire a strong level of engagement. Lean leaders articulate this goal, and support people in solving problems as they seek to attain it.

Lean has produced many tools and methods (value stream mapping, structured problem solving, standard work, and Hoshin Kanri to name but a few). These tools have been refined over decades, but are rarely applied effectively at the level of achieving breakthrough results. This is typically due to a lack of the focused awareness and attention required to experience the thinking/learning/discovery process inherent in all Lean practice.

Table 3.2 outlines the elements of a Lean culture.

Table 3.2 Elements of Lean culture

Trust	The assured reliance on the character, ability, strength, or truth of someone or something, including a team, a work process, and, most importantly, process.
Respect	The act of giving particular attention, consideration, special regard, and esteem to another.
Curiosity	A relentless desire to know how and why things work, what makes things work better, and what 'better' looks like after things have been made better.
Enquiry	A systematic search for the facts about the nature of things: their origins, their causes, their interdependencies, their lifecycles, and their nature.
Playfulness	A fresh, fun way of viewing ideas and their relationship to other ideas while simultaneously maintaining serious focus.
Intensity	A deep focus on the topic at hand, and the persistence not to become distracted or lose the path.

What needs to be in place to create, nurture, and sustain a Lean culture?

Leaders must know what to do to foster a mindful Lean culture and why they are doing it. The key skill that leaders need to develop is present-moment awareness. This approach focuses on creating an environment where frontline workers who are given the safety, support, time, and space to try new things will learn from what works and from what doesn't work.

Leaders need to be deeply aware of what people are saying and doing, and respectfully connect with others. Employees have to believe they are safe when bringing up problems, offering ideas, asking for help, or simply saying they are unclear on their work. The behaviours of traditional managers do not communicate any of these beliefs, create psychological safety, or help create the kind of environment that invites or supports employee engagement or initiative.

Perhaps the most important thing leaders and managers can do is shape culture and align focus. To make this happen, they need to know what culture they are trying to build, and clearly articulate it within the context of the goals of their organization. Again, this requires a high degree of personal awareness and psychological safety.

Managers who are supported and incentivized to create a new way

Senior leadership must model, coach, and reinforce new ways of thinking, acting, and supporting people. For this to happen, they must be aware of, and put into practice, some new behaviours:

- Ask questions if they do not have the information they need.
- Listen to the person, not just the problem.
- Acknowledge that they heard, and what they heard.
- Ask questions focused on things they wonder about, not about what they are thinking.
- Ask what help is needed.

These behaviours can be espoused in conducting Gemba walks. Gemba walks are a major part of Lean management philosophy. Managers observe the actual work process, understand the work, ask questions, and learn. This leads to a better understanding of the whole value stream and its problems, rather than basing understanding on second-hand reporting and an idealized abstraction of the workplace.

These behaviours require a high degree of self-awareness and emotional intelligence. Mindfulness is the key skill that positions someone to develop the communication, coaching, and leadership skills needed to support and incentivize others. Thinking without awareness is known as *muda* (waste, non-value-added activity). This is why mindful presence or deep flow is so essential to a Lean culture.

Figure 3.9 highlights the key behavioural patterns that are relevant to Lean culture.

Key message

When applying Lean culture, practitioners should aspire to the following behaviour:

- Trust people and the 'system', but remain vigilant and give feedback when needed.
- Treat people decently.
- Strive to understand how things work and what could be improved.
- Gather facts systematically and challenge hypotheses.
- Develop new insights with a combination of creativity and analysis.
- Focus mindfully on the topic at hand.

The ITIL story: Lean culture

Marco: *We encourage our app development team to adopt a Lean culture, creating a work environment of trust, respect, curiosity, enquiry, playfulness, and intensity.*

Solmaz: *The app development work can often be experimental, and adopting such a culture enables our team to take calculated risks, and helps them to explore a variety of options for implementing new functionality.*

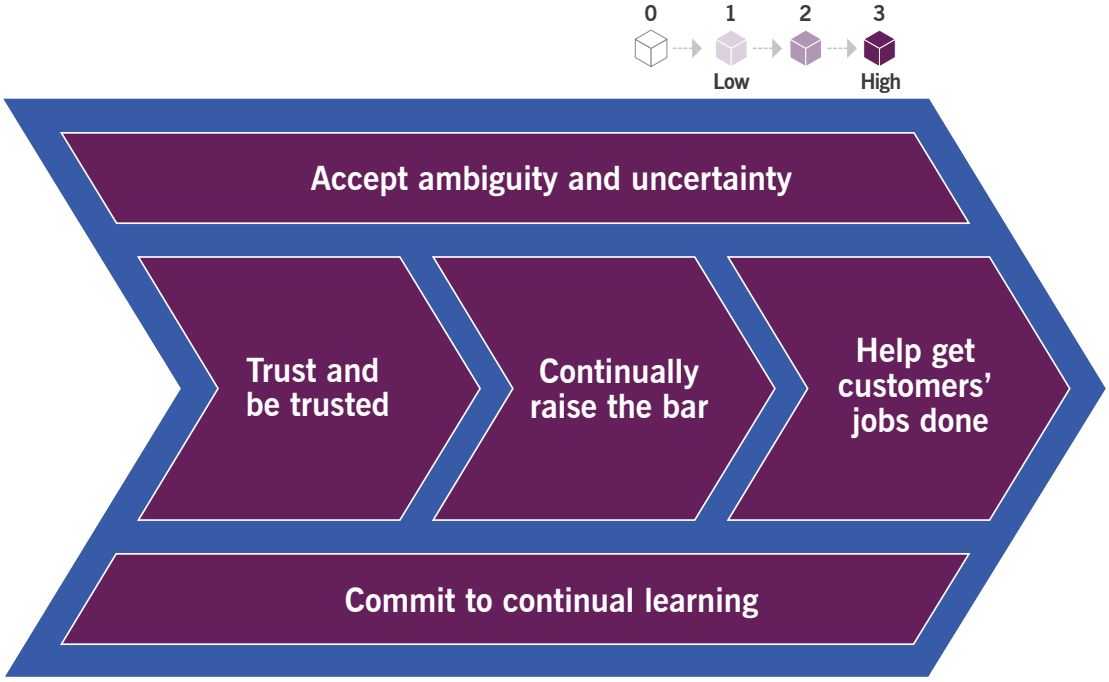

Figure 3.9 Heat map of the importance of the key behaviour patterns to Lean culture

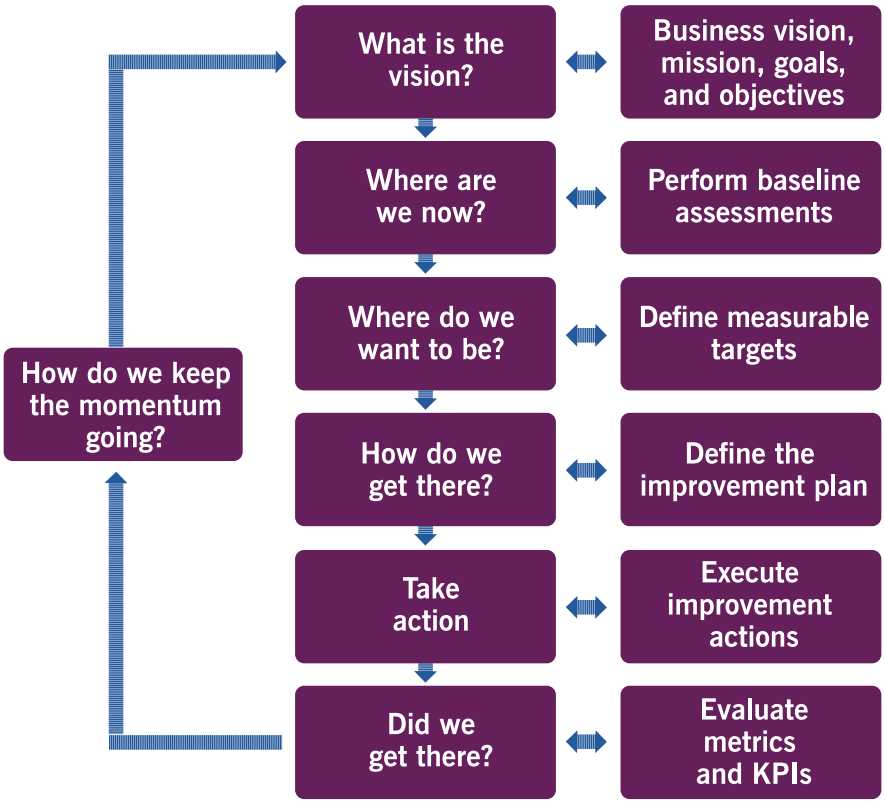

Figure 3.10 The ITIL continual improvement model

3.2.3.3 ITIL continual improvement model

The ITIL continual improvement model (see Figure 3.10) provides organizations with a structured approach to implementing improvements. The model applies to the whole service value system, and its use makes initiatives more likely to be successful. It supports an iterative approach to improvement, dividing work into manageable pieces with separate goals that can be achieved incrementally.

HVIT environments apply the continual improvement model with a focus on iteration, experimentation, and data-driven scientific thinking. The continual improvement model is a critical component of an organization's overall approach to improvement that should be applied to products and services, all management practices, and to the continual improvement practice itself (see Figure 3.11). Although the development and review of the

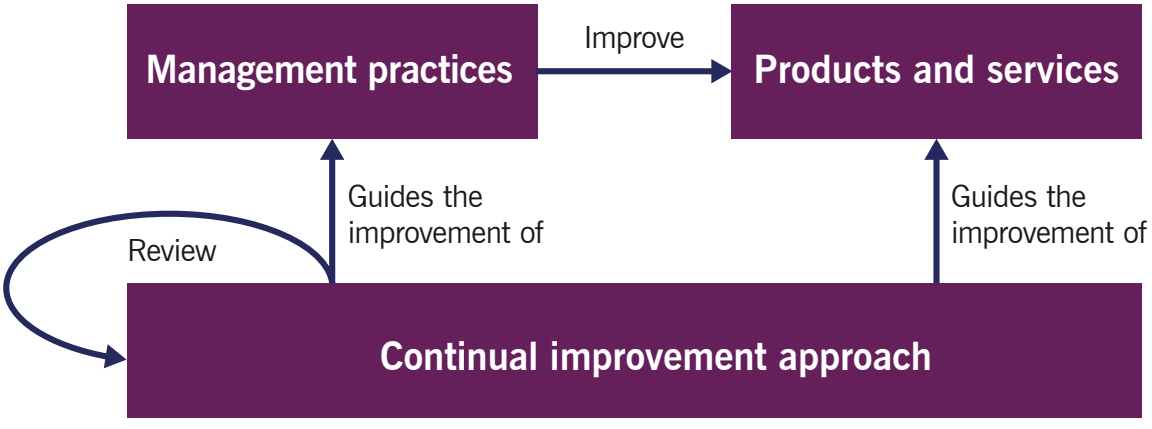

Figure 3.11 Improvement domains

continual improvement approach is the responsibility of an organization's management, it is recommended that practitioners at all levels engage in continual improvement.

More information about the continual improvement model and its application can be found in the continual improvement practice guide and in *ITIL® 4: Direct, Plan and Improve*.

Toyota Kata

Of all types of organization, those that are digitally enabled are under particular pressure to continually improve their performance under changeable market conditions. Circumstances are often unpredictable, making it difficult or even irresponsible to create and follow predetermined plans for large changes. Improvements are taken step by step, based on sound interpretation of available information. Changing circumstances require organizations to be flexible and creative in the way they define ways to improve. At the same time, improvement initiatives should be disciplined, data driven, and justified. This requires belief in scientific thinking, disciplined execution, practice to unlearn old habits and learn and sustain new ones, and confidence and competence to improvise when appropriate.

Disciplined scientific thinking with incremental improvements reduces our natural bias to jump to the wrong conclusions. Practice and coaching help us form and sustain new habits, after which we can start improvising. One approach to scientific experimentation is Toyota Kata.

Definition: Toyota Kata

A mental model and behaviour pattern for scientific thinking and routines for practice and coaching.

This four-step improvement approach is based on five questions:

- What are we trying to achieve?
- Where are we now?
- What obstacle is now in our way?
- What is our next step, and what do we expect?
- When can we see what we have learned from taking that step?

The steps of the improvement kata are:

1. Understand the direction: improvement should be aimed at specific goals, not just random.
2. Grasp the current condition: a direction is not useful unless we know where we are right now.
3. Establish the next target condition and identify obstacles: describe both the outcome we desire next and the expected condition of the process to generate that outcome.
4. Experiment towards the next target condition: come up with ideas to overcome an obstacle and run experiments with that idea. If possible, test only one hypothesis at a time.

These steps are outlined in Figure 3.12.

In the context of the ITIL continual improvement model, Toyota Kata helps to answer the questions:

- Step 1: What is the vision?
- Step 2: Where are we now?

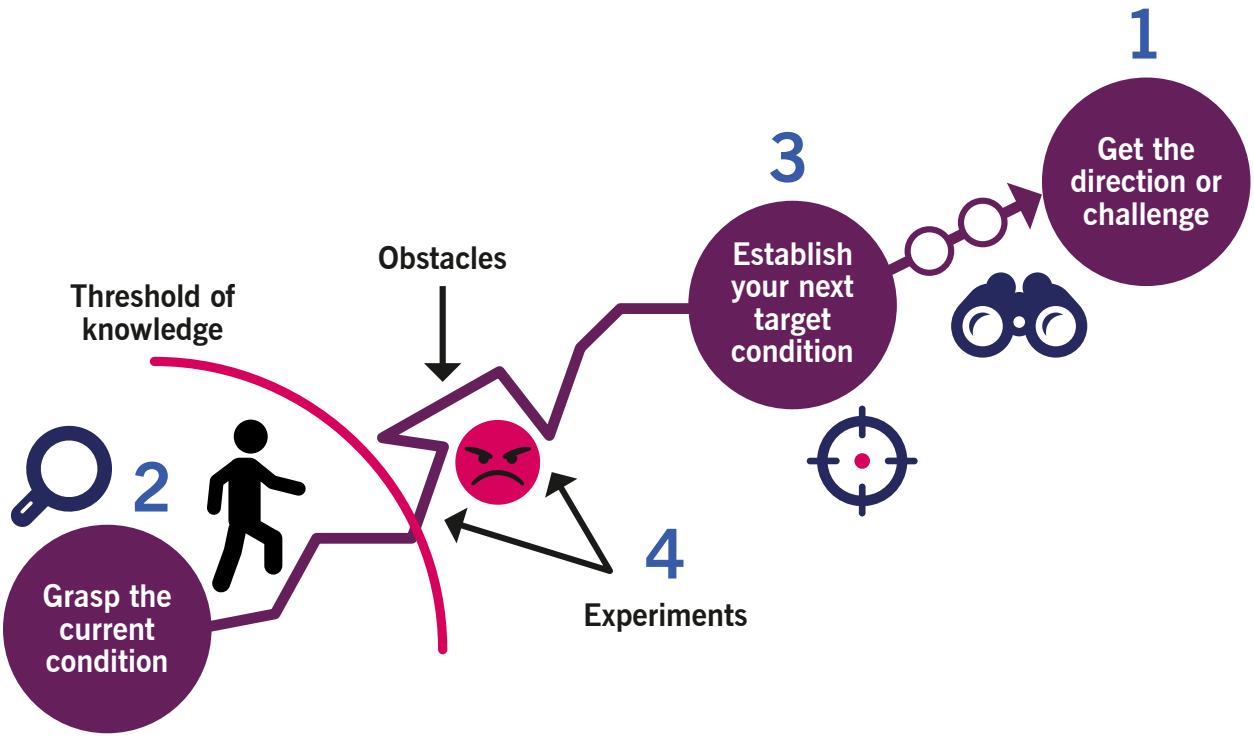

Figure 3.12 Toyota Kata[10]

- Step 3: Where do we want to be?
- Step 4: How do we get there?

This approach works effectively where there are many options available to answer the question of 'How do we get there?' or even 'Where do we want to be?', and experimentation is the best way to answer these.

Toyota Kata helps justify confidence to pursue seemingly unattainable goals in complex systems, and helps teams that make their own decisions and manoeuvre effectively.

The adoption of Toyota Kata demonstrates commitment to higher performance, and helps deal with uncertainty by taking a step-by-step approach. Its emphasis on observation, discovery, and improvement correlates strongly with improving by being inquisitive.

OODA loop

The OODA loop[11] is another improvement technique that can be used in the context of HVIT. OODA stands for 'observe, orient, decide, act'. The OODA loop originated in operations during military campaigns, and is now also often applied in other types of organizations. The approach shows how agility can overcome raw power. It is especially applicable to cybersecurity, and is considered to make processes seem more immediately reactive than the more familiar PDCA (plan–do–check–act) approach. It is therefore popular in industries where unexpected challenges arise.

The OODA loop was developed to explain how to direct energies to defeat a combatant. The loop is actually a set of interacting loops that are in continual operation during combat. All decisions are based on observations of the evolving situation in combination with filtering of the issue at hand. Decisions and actions are based on observations, which are processed to orient them for decision-making. Orientation is the most important part of the loop because it informs how to observe, decide, and act. Orientation is influenced by genetic heritage, cultural tradition, and previous experience.

Key message

When using continual improvement, practitioners should aspire to the following behaviour:

- Establish business vision, mission, and objectives.
- Perform baseline assessments.
- Define measurable targets.
- Define the improvement plan.
- Recognize complexity and experiment.
- Execute improvement actions.
- Evaluate metrics and key performance indicators (KPIs).

Figure 3.13 highlights the key behavioural patterns that are relevant to continual improvement.

The ITIL story: ITIL continual improvement model

Solmaz: *We follow the steps of the ITIL continual improvement model when implementing new functionality and improvements to our app. However, sometimes we do not know how to reach the next target state, or even what that state might be. In such situations, we will experiment to identify the best next step.*

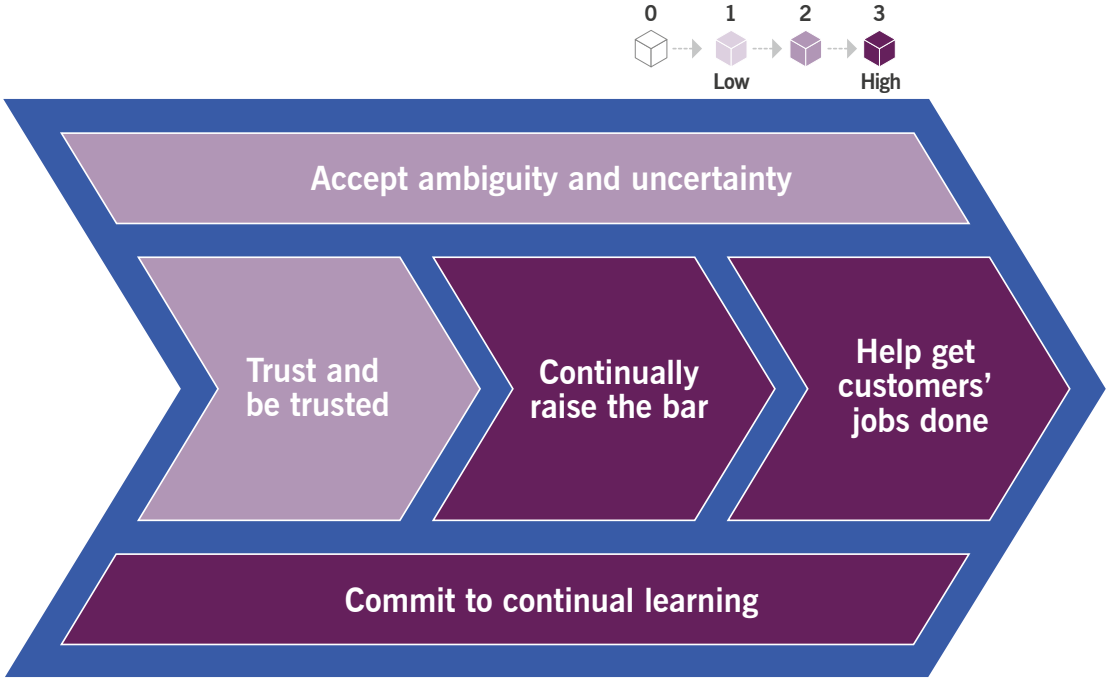

Figure 3.13 Heat map of the importance of the key behaviour patterns to continual improvement

3.3 ITIL guiding principles

The ITIL guiding principles embody the core messages of ITIL and of service management in general. They help with adopting and adapting ITIL guidance to the specific needs and circumstances of any organization, including HVIT organizations. The principles also encourage and support organizations in continual improvement at all levels.

Some of the ITIL guiding principles manifest themselves more in an HVIT environment than others. This section outlines how each principle can be applied to an HVIT way of working.

3.3.1 Focus on value

The focus on value principle states that everything an organization does needs to map, directly or indirectly, to value for the stakeholders. It encompasses many perspectives, including the experience of customers and users.

The focus on value principle is reflected in two of the HVIT objectives: valuable investments and co-created value. This principle is particularly applicable to HVIT because the digital products and services that HVIT organizations provide will have an effect on their consumers.

When working in an HVIT organization, it is important to keep in mind that value may need to be created for:

- colleagues across the organization (for instance, compliance officers, auditors, support agents, or development teams)
- external stakeholders who benefit from value, wealth, or information generated between the organization and its consumers (for instance, shareholders, tax authorities, regulatory authorities, suppliers, or technology vendors).

Each stakeholder group might have a different expectation of value, and the organization needs to find ways to balance often conflicting needs. In an HVIT environment, stakeholders and their needs are likely to change often. As a result, the focus on value principle has a very strong connection with the principle of think and work holistically.

3.3.2 Start where you are

The start where you are principle is focused on not starting from scratch and building something new without considering what is already available to be leveraged. This is particularly true for an organization that is looking to undergo a digital transformation. The organization may already have many of the resources needed for high-velocity work, although it might need to modify them to meet new objectives and ways of working. For example:

- Existing change enablement processes might need to be refocused to emphasize standard change models rather than approvals by a change advisory board.
- Project management may need to focus on governance of work against the business case, and coordination across multiple IT and non-IT teams.

There is a danger that, when pursuing an HVIT way of working, an organization may affect its existing relationships, wisdom, and intelligence. Although this may be necessary in some cases, organizations should be careful to review what they have before proceeding, as much of this may be kept for HVIT work. For example:

- Many project managers know and understand the time needed to release a product across a complex ecosystem.
- Many procurement managers have built relationships with key vendors, partners, and suppliers, who might be able to provide resources for high-velocity work.

HVIT is about reducing the total time from idea to launch for the delivery of a digital product. By considering options to re-use what is already in place rather than simply building from scratch, organizations can often deliver products to market faster.

3.3.3 Progress iteratively with feedback

This principle is focused on resisting the temptation to do everything at once. Even huge initiatives have to be accomplished iteratively. When work is organized into smaller, manageable sections that can be executed and completed in a timely manner, the focus on each smaller effort is sharper and easier to maintain. Using feedback before, after, and throughout each iteration will ensure that actions are focused and appropriate, even in changing and unpredictable circumstances.

Progressing iteratively with feedback is one of the core principles of the Toyota Kata way of improving, which offers guidance for step-by-step improvement based on data-driven, scientific thinking (see section 3.2.3.3).

An iterative approach is not limited to software development, and can be adopted by service management working in a high-velocity environment. For example:

- The introduction or modification of change enablement to focus on standardizing changes can be rolled out in an iterative way, using feedback from all relevant stakeholders to incrementally improve the practice.
- Continual improvement registers can be written as epics and user stories, and scheduled into short iterations that use feedback loops to determine whether the improvements have had the desired effects.

Service delivery review cycles with key partners and vendors can be shortened from quarterly to monthly, or even weekly, cycles, creating a fast delivery and feedback loop to enable a high-velocity way of working.

3.3.4 Collaborate and promote visibility

This principle states that working together across boundaries produces results that have better buy-in, greater relevance to objectives, and better likelihood of long-term success. Accomplishment requires information, understanding, and trust. Work and consequences should be made visible, hidden agendas should be avoided, and information should be shared to the greatest degree possible.

This principle is particularly relevant to HVIT, as high-velocity work relies on quick access to different resources, including information, and a highly collaborative way of work.

When working in an HVIT environment, it is important to:

- modulate the degree of independence, self-organization, collaboration, and visibility to minimize overhead work
- promote peer-to-peer visibility along a value stream, along with hierarchical visibility between senior management and leadership, and engineers and frontline staff.

Examples of this principle put into practice include:

- logging technical changes and incidents in a centralized tool for reporting and escalating to senior management
- using a publicly accessible white board or wall to document work in progress, but also maintaining information in an electronic format so that it can be easily communicated, analysed, and reported on
- senior leadership publishing and promoting organizational goals, objectives, and policies.

3.3.5 Think and work holistically

This principle states that no service or element that is used to provide a service stands alone. The outcomes achieved by the service provider and the service consumer will suffer unless the organization works on the whole, not just on the parts.

Results are delivered to internal and external customers through the effective and efficient management and the dynamic integration of information, technology, organization, people, practices, partners, and agreements, all coordinated to provide a defined value.

Much thinking in HVIT rightly focuses on meeting the needs of consumers. However, there are many other stakeholders that can be positively, or negatively, impacted by high-velocity work. It is therefore important to think about the implications that decisions made by high-velocity teams have for all stakeholders. For example:

- Reducing documentation efforts might have a positive impact on the software development and management practice, but can negatively affect other practices such as service desk, release management, knowledge management, incident management, or problem management.
- Moving to cloud computing changes the balance of capital expenditure and revenue/operating expenditure, which in turn has an impact on financial governance and the results the organization reports to shareholders and tax authorities.

Introducing new digital products and services can have knock-on effects on industries (famous examples include taxis, hospitality, and video rental), which in turn can create positive and negative outcomes for societies and local economies.

3.3.6 Keep it simple and practical

This principle states that if a process, service, action, or metric provides no value or produces no useful outcome, it should be eliminated. In a process or procedure, the minimum number of steps needed to accomplish the objective(s) should always be used. Outcome-based thinking should also always be used to produce practical solutions that deliver results.

This principle applies to HVIT work and the systems surrounding and enabling it, and to the consumption of products and services.

Keeping it simple does not mean keeping things simplistic, but refers to the philosophy of reducing overhead responsibilities, optimizing (or in some cases reducing) the degree of automation, and reducing friction between teams.

Examples of how this principle can be applied to HVIT work include:

- identifying the minimum information required to get work done, and not over-engineering monitoring and configuration management database (CMDB) tools to track everything that the organization can manage
- using a wall with sticky notes to track work, rather than investing in an electronic work management system
- reducing the number of fields that a consumer has to fill in to sign up for an account, but still collecting enough information to enable service and support work.

3.3.7 Optimize and automate

This principle states that resources of all types, particularly human resources, should be used to their best effect. Anything that is truly wasteful should be eliminated, and technology should be used to do whatever technology can do. Human intervention should be reserved for where it really contributes value.

The optimize and automate principle manifests itself not only in the continual improvement of an organization's digital products and services, but also in the improvement of its processes. Examples of applying this principle to HVIT work include:

- documenting and streamlining the processes that are in place to test software before investing in CI/CD tools
- iteratively designing forms for different ticket types (e.g. change requests, incident tickets, or request forms) before looking to IT and service management tools and workflow to speed up the processing of tickets.

It is a common mistake to optimize a single part of a system and to work for individual efficiency rather than overall effectiveness. This often creates queues that actually slow down the overall process, even when individual effectiveness is optimized.

3.4 Summary

This chapter explored the cultural aspects of digital transformation, and the key approaches through which the behaviour and attitudes of practitioners can evolve in relation to those aspects.

To survive and succeed in a high-velocity environment, organizations and people should adopt the following behaviour patterns:

- accept ambiguity and uncertainty
- trust and be trusted
- continually raise the bar
- help get customers' jobs done
- commit to continual learning.

To support these behaviours, organizations should evolve the way they think and operate in regard to purpose, people, and progress:

- how they define and fulfil their mission and objectives
- how they ensure a productive, safe, stress-free environment for their people
- how they enable high performance in constantly changing circumstances.

The chapter described and explained approaches and techniques to address all three aspects.

These behaviour patterns, and organizational culture in general, create an environment where many of the techniques and models described in Chapter 4 can be successfully applied.

CHAPTER 4

HIGH-VELOCITY TECHNIQUES

4 High-velocity IT techniques

This chapter describes a selection of techniques that characterize HVIT environments. Some are usually only found in these environments, whereas others are more general techniques that are crucial to HVIT work. The selection is not exhaustive; these techniques are examples that characterize ways of working that help highly digitally enabled organizations to achieve their demanding objectives.

The techniques in this chapter are grouped and described based on their relationship with one of the five HVIT objectives:

- valuable investments
- fast development
- resilient operations
- co-created value
- assured conformance.

Although the techniques listed here are grouped by these objectives, a technique will often support multiple objectives. Each technique can be used in the context of multiple ITIL management practices, and at the end of each of the following sections is a table that lists which practices the technique being discussed is relevant to, and outlines how that technique contributes to each practice. There is also a heat map for each of the techniques, which roughly indicates how much each ITIL service value chain activity is affected by that technique.

The ITIL story: High-velocity IT techniques

Su: *The update to our booking app that will improve its performance on smartphones and devices has been successfully deployed. We have plans to develop the app further, adding supplementary functionality, such as a membership programme, affiliate links, priority booking, and vehicle upgrades. We utilized, and will continue to utilize, many techniques that help us to optimize our work and focus on value.*

4.1 Techniques for valuable investments

The valuable investments objective involves identifying and justifying digital investments that would contribute significantly to business strategy. This exercise should result in a good understanding of a digital investment's potential value, anticipated costs, and return on investment, and the defined criteria for its utility. The warranty, or non-functional requirements, should also be determined, although it does not add to the potential value of the functionality. Warranty ensures that an investment's potential value is not adversely affected by outages, poor use, or other factors.

Making valuable investments is founded in market research and the development of new products. New digital products and services should be envisaged and evaluated in terms of profitability. The substantive quality of products and services and the timing with which they are launched are both crucial factors for gaining and maintaining a competitive advantage. The sooner a potential investment is envisaged and evaluated, the sooner benefits, such as competitive advantage, can be realized. Ethical principles should be used to make investment decisions that consider a broad range of stakeholders' interests.

It is also important to continually evaluate investments after they have been justified and approved, because more valuable options for investment may exist. The sooner information about alternative investments is made available, the sooner current investments can be re-evaluated.

Making valuable investments in commercial organizations often means achieving more revenue through increased sales and higher prices, reduced capital and operational expenditure, or reduced risk. In non-profit organizations, valuable investments are not focused on revenue, but relate instead to their organization-specific primary objective. Valuable investments can be measured by sales revenue and costs, but also by competencies that have been developed during the realization of the investment. The value of an investment can only be determined when the return is realized; this occurs after the investment, when value is co-created between the provider, the consumer, and other stakeholders.

In digital organizations, IT drives and enables the business. It is therefore of crucial importance that the ever-evolving potential of IT is continually evaluated for strategic advantage. The main concerns in this should be the intrinsic quality of investment initiatives and the speed with which they can be identified, evaluated, designed, developed, and deployed. There is always a period of unawareness before a potential investment, which will relate to either opportunity or demand, is identified. The sooner technological developments are identified and their business potential is assessed, the sooner an investment decision can be made.

Techniques that can be used to achieve valuable investments include:

- prioritization techniques
- minimum viable products and services
- product or service ownership
- A/B testing.

The ITIL story: Techniques for valuable investments

Marco: *Our business strategy is enabled by judicious investment in technology. Because we use Agile working techniques, we can guarantee the warranty of our technology: our code will always work. We monitor our investments to ensure that we spend wisely and that our app's functionality matches our customers' requirements.*

4.1.1 Prioritization techniques

Queues occur wherever the demand for work exceeds the capacity to complete it within the expected timeframe. In an ideal situation, an organization would have no variation in demand and would have the appropriate quality and quantity of resources needed to satisfy it. However, organizations often need to contend with having a fixed capacity but a varying demand for services. This imbalance creates queues or backlogs in which work items need to be prioritized.

Prioritization is an activity commonly associated with support and software development work (for example, prioritizing incident investigation or a sprint backlog), but its use is universal.

4.1.1.1 Cost of delay

A useful technique for prioritization is estimating the cost of delay of a new or improved service offering. This refers to the financial and non-financial benefits that would be lost if a service activity or task were delayed. An understanding of the cost of delay gives practitioners the ability to prioritize work based on data on value, rather than their intuition. This applies to the initial prioritization and the continual evaluation and reprioritization of ongoing work within changing circumstances. Almost always, the business criticality of digital products and services justifies the effort involved in estimating the cost of delay.

Cost of delay can be applied to decision-making at various levels, such as in large investments at a product or service level within a product or service portfolio, smaller investments at a feature level within products or services, or operational tasks.

This technique is particularly useful in HVIT environments because investments are generally more significant and market conditions rapidly change, meaning it is important to continually assess options for alternative investments.

4.1.1.2 Buy/hold/sell

A portfolio of products (or other assets) can be managed using the buy/hold/sell technique. This involves assessing each product and deciding which of three investment strategies best applies to it:

- **Buy** Invest in improving or extending the product.
- **Hold** Spend as little as possible to maintain the product, as long as the costs are affordable.
- **Sell** Invest in retiring, reducing, or replacing the product.

This technique illuminates the difference between the costs and benefits of developing, maintaining, or retiring a product, along with the associated risks and trade-offs. It helps decision-makers to be explicit about their choices and to accept the consequences of their decisions.

4.1.1.3 Other techniques

There are many other product prioritization techniques that product/service owners can consider, including stacked ranking, Kano, net present value (NPV), return on investment (ROI), and fit/feasibility/attractiveness.

Figure 4.1 shows the contribution of prioritization to the service value chain.

Table 4.1 outlines the practices for which prioritization is relevant.

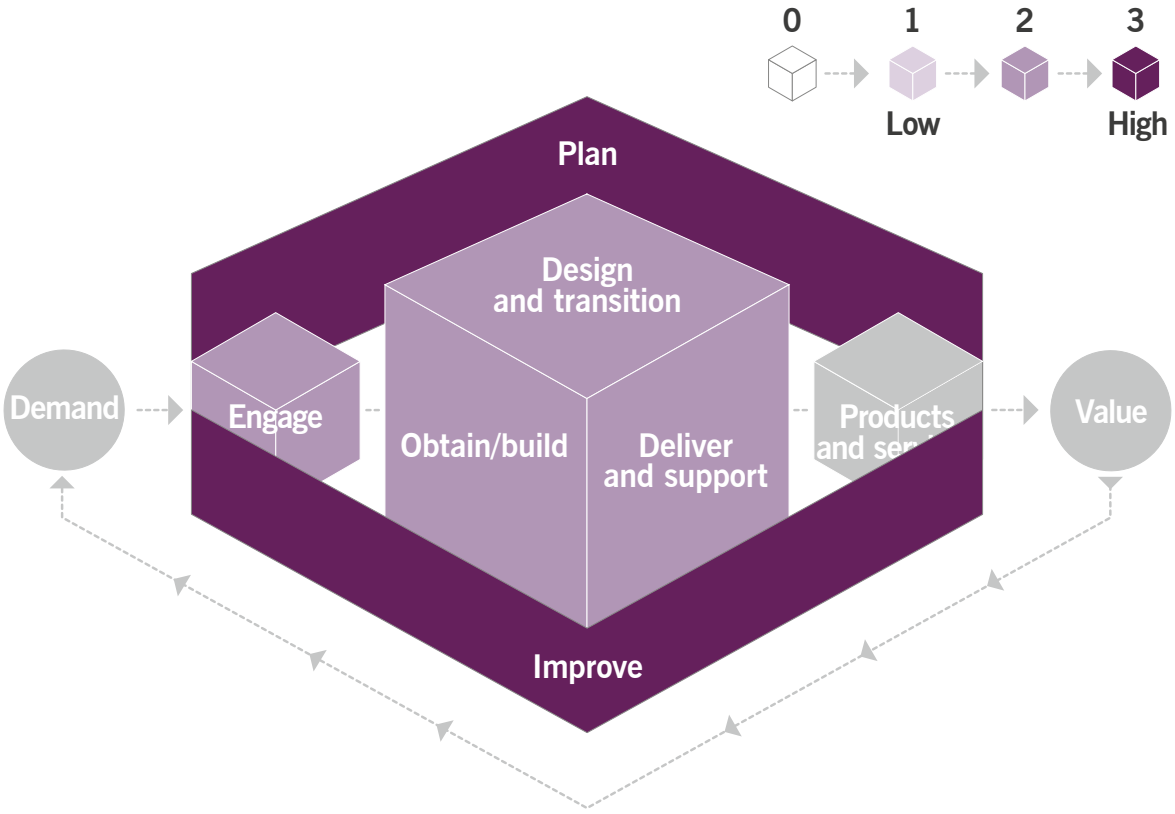

Figure 4.1 Heat map of the contribution of prioritization to the service value chain

Table 4.1 Practices for which prioritization is relevant

ITIL management practice	Activities/resources associated with prioritization	Impact
Portfolio management	Continually prioritizing service offerings based on value, incorporating the cost of delay.	H
Problem management	Calculating the financial cost of open problems and errors in order to prioritize and direct problem management efforts. Comparing the costs of workarounds with those of longer-term solutions.	H
Project management	Calculating the financial impact of performing or delaying project work.	H
Software development and management	Calculating the financial impact of delaying work on new software features or larger software-based service components. Deciding whether to obtain or build software components.	H
Change enablement	Calculating the cost and benefit of prioritizing and scheduling changes to services or service components.	M
Incident management	Calculating the cost of incidents and major incidents in order to prioritize work that has the highest economic impact.	M
Release management	Calculating the cost and benefit of prioritizing and scheduling releases of new or changed services.	M
Service financial management	Calculating time value profile data to provide information for prioritizing service offerings.	M
Service request management	Calculating and comparing the financial impact of fulfilling or delaying the fulfilment of requests in order to prioritize work with the highest benefit.	L

The ITIL story: Prioritization techniques

Su: *After the app updates were deployed, our priorities became split. We wanted to develop new features from our sprint backlog, but there were support requests that we needed to manage in order to ensure our customers were happy with our service.*

Radhika: *Just as we were engaging in a marketing push, we realized that the known errors in our app were generating bad publicity. Therefore, the cost of delaying the fixes became greater than the value of adding new features.*

Su: *We used the buy/hold/sell technique to plan investments in our product suite:*

- ***Buy*** *We invested in the app, improving the experience and extending its features.*
- ***Hold*** *The option to book on our website had to be maintained because customers were still using it. However, we chose to minimize the investment in it.*
- ***Sell*** *A small number of business customers still wanted to book via fax. We retired this functionality, working with customers to transition them to more modern modes of communication.*

We worked hard to guarantee a satisfactory return on investment.

4.1.2 Minimum viable products and services

A minimum viable product or service is one that has just enough features to enable its early assessment and the collection of feedback for future development. The 'minimum viable' approach is an efficient way of developing products and services, particularly when the market is volatile and unpredictable. It is consistent with complexity thinking, which recognizes that some things are unknowable and therefore that it is impossible to design a product or service with complete, predetermined requirements. When requirements are unknown, unarticulated, or ambiguous, experiments can establish what works and what doesn't. Therefore, a minimum viable approach enables valuable investments and contributes to fast development through an iterative working style.

In volatile markets, it can be difficult to judge which product or service offerings will be successful. This uncertainty can be addressed through a minimum viable approach. Instead of investing considerable resources and time into developing a full-scale product or service, the product or service provider should limit their efforts. They should aim for a product or service that is just developed enough to stimulate feedback and other data, and can then indicate whether and how development should continue. As soon as enough data is gathered, a decision can be made, which increases the chances of success. Furthermore, if a decision is made to stop development, the product or service provider can then allocate their resources to another investment, minimizing waste on the original idea.

A minimum viable product or service typically has three key characteristics:

- It has enough value that people are willing to use or buy it.
- It demonstrates the potential for enough benefit to retain those early adopters.
- It provides a feedback loop to guide future development.

Figure 4.2 shows a heat map of the contribution of a minimum viable approach to the service value chain.

Table 4.2 outlines the practices for which a minimum viable approach is relevant.

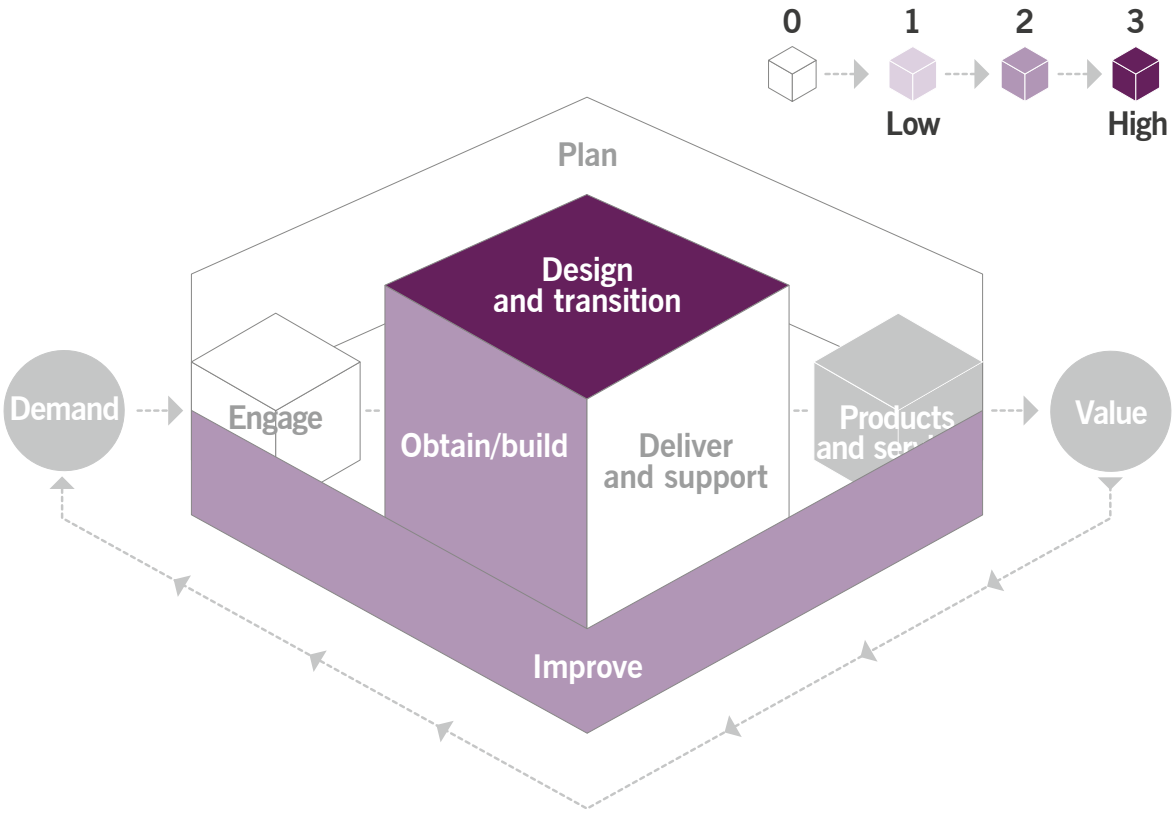

Figure 4.2 Heat map of the contribution of a minimum viable approach to the service value chain

Table 4.2 Practices for which a minimum viable approach is relevant

ITIL management practice	Activities/resources associated with a minimum viable approach	Impact
Architecture management	Using a minimum viable approach to describe the service, technical, information, or environmental architecture that will create the necessary constraints, boundaries, or enablers for other types of service or product work.	H
Business analysis	Using a minimum viable approach as a tool to extract core business value from a product or service.	H
Capacity and performance management	Using capacity and performance management as a basis for calculating the minimum resources (number of servers, number of service desk agents, etc.) required for a minimum viable product or service.	H
Monitoring and event management	Using monitoring and event management as the basis for designing and configuring tools for monitoring and telemetry, which are used to operate and learn from the minimum viable product or service.	H
Portfolio management	Using the concept of a minimal viable product as a dynamic decision-making tool to support good investments in the portfolio of features within a product or service.	H
Project management	Articulating the minimum output needed to satisfy the business case.	H
Service design	Using a minimum viable approach to design the necessary customer experience and user experience elements of a product or service.	H
Service validation and testing	Developing test cases to check that all service components support the minimum viable product or service.	H
Software development and management	Using a minimum viable approach as a decision-making tool to prioritize work on software features.	H
Infrastructure and platform management	Using a minimum viable approach as a decision-making tool to design, implement, and prioritize ongoing work on infrastructure components.	M
Service catalogue management	Using a minimum viable approach as a basis to describe all products, services, and service offerings, and ensuring this information is available to relevant audiences.	M
Service continuity management	Designing and building continuity plans to support a minimum viable product or service.	M
Supplier management	Using a minimum viable approach to articulate the outputs required when partners and suppliers provide products and services.	M

The ITIL story: Minimum viable products and services

Su: *As we develop new app functionality, we launch it as a minimum viable product so that we can gauge customer interest. This helps to ensure that we have not invested more resources than necessary in development, and enables us to understand market demand. The feedback on the minimum viable product determines future prioritization.*

Solmaz: *We acknowledge that we do not know what future customers would like. By working iteratively, we can test the product at every stage and, if we make a mistake, return to previous successful versions without sacrificing significant investment.*

4.1.3 Product or service ownership

Scrum suggests three roles: the product owner, the development team, and the Scrum master. The product owner is responsible for maximizing the value of the product that the development team produces. Product ownership entails establishing and prioritizing requirements, and communicating them to the development team. In the context of these software development teams, it is the product owner who liaises and negotiates with the various stakeholders, including consumers. HVIT environments are often product-oriented, so the concept of the product owner is highly relevant to HVIT. The concept of a product owner also applies to services and service owners.

The product owner role relies upon:

- **Skills and experience** Product owners need to be skilled in stakeholder management, requirements analysis, market research, customer segmentation, and more. Product owners from a business analysis or product management background often have these skills, and they may only need a short course to familiarize themselves with Agile ways of working. However, technical staff who become product owners may need more training.
- **Authority** At a minimum, a product owner must be able to identify and communicate short-term priorities without prolonged or unnecessary debate and discussion. Organizations should trust product owners to use their authority.
- **Legitimacy** The product owner also needs to have legitimacy. Direct customer contact and first-hand experience will bolster this legitimacy, and give them the knowledge they need to prioritize effectively.
- **Time** Product owners need the time to fulfil their role, including to think about their stories, filter and edit the backlog, visit stakeholders, analyse feedback, work with the team, evaluate post-delivery benefit realization, and reflect on the progress and current state of their projects.

Product or service ownership is a key part of all value chain activities, and contributes to valuable investments.

Figure 4.3 shows the contribution of product or service ownership to the service value chain.

Table 4.3 outlines the practices for which product or service ownership is relevant.

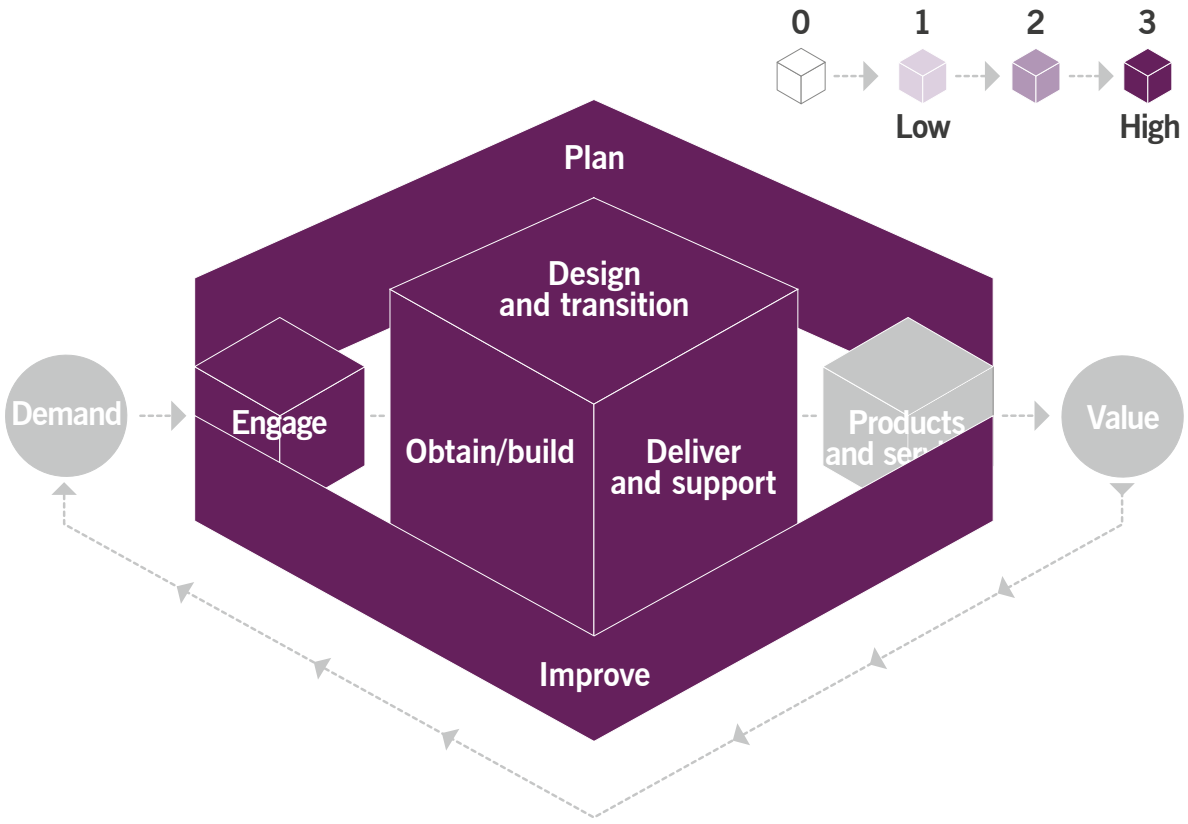

Figure 4.3 Heat map of the contribution of product or service ownership to the service value chain

Table 4.3 Practices for which product or service ownership is relevant

ITIL management practice	Activities/resources associated with product or service ownership	Impact
Infrastructure and platform management	Involvement of product and service owners in articulating, refining, and prioritizing the infrastructure and platform development backlog(s), and in deciding whether to acquire commercially available infrastructure components and services.	H
Portfolio management	Involvement of (software) product owners and product or service managers in evaluating and prioritizing the product or service investment proposals.	H
Relationship management	Structuring interactions with stakeholders. Involvement in establishing customers' priorities for new or changed products and services. Coordinating customers' requirements and feedback. Involvement in addressing complaints and mediating conflicting requirements.	H
Service catalogue management	Involvement of product or service owners and managers in publishing information on all products, services, and service offerings.	H
Software development and management	Involvement of product and service owners in articulating, refining, and prioritizing the software development backlog, and in deciding whether to acquire or upgrade commercially available software.	H
Project management	Involvement of product and service owners and managers in delivering project work and managing risks.	M
Risk management	Involvement of product and service owners in articulating and mitigating enterprise risks.	M
Supplier management	Involvement of product and service owners and managers in articulating needs, structuring interactions, and negotiating with partners and suppliers.	M

The ITIL story: Product or service ownership

Su: *I am the dedicated product owner for the booking app. I liaise and negotiate with the teams in development, marketing, fleet management, booking, and others. I prioritize requirements and regularly communicate the priorities to stakeholders.*

I have a technical background with experience in Agile development and training in business analysis, including time spent working with customers. I understand the level of changes I can authorize and when I need to escalate an issue. Axle ensures that I have the time to fulfil my role and that I understand what is required of me and how the product focuses on value.

4.1.4 A/B testing

It can difficult to predict whether a feature will be valuable to users. This problem could be solved by measuring user behaviour to gather solid data; however, when there are too many influencing factors, it is almost impossible to isolate the effect of the new feature. Therefore, a control group is required.

A/B testing is a time-limited experiment in which one group of users, the control group, is provided with an old version of a product or service. At the same time, another group of users, the treatment group, is provided with a new version of the product or service that includes the new feature. Assuming all other factors influencing both groups are equal, the measurements of the groups can be compared, thus gathering data for a value-based decision. This method is illustrated in Figure 4.4.

A/B testing contributes to the portfolio management practice. The essence of portfolio management is making the right investments within the constraints of funding and resources. Portfolio management is applicable at a variety of levels, including the 'portfolio' of features within a product or service. A/B testing helps when determining which version of a feature is most valuable. As such, it contributes to valuable investments.

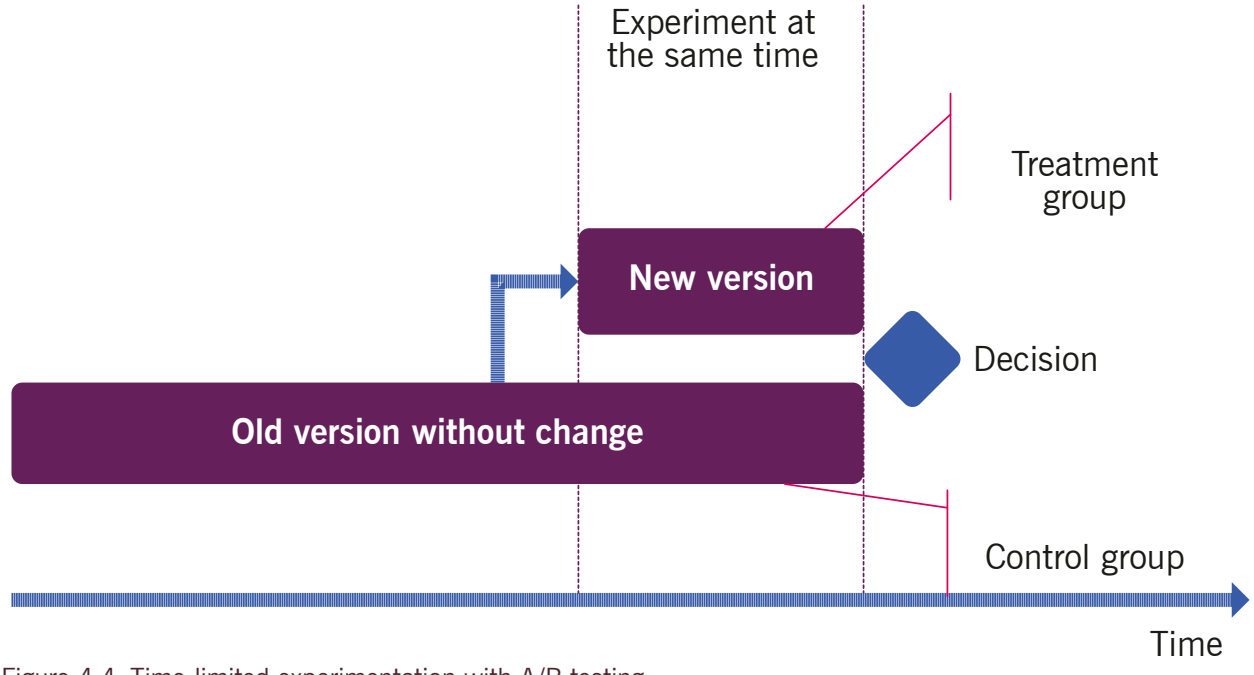

Figure 4.4 Time-limited experimentation with A/B testing

Example

An organization's marketing department wants to change a product description on the organization's website by adding short videos of the product. The department is sure this new feature will significantly increase conversion rate, which is currently 2.5 per cent (that is, 2.5 per cent of visitors add this product to their shopping carts). If the feature were implemented without A/B testing and the conversion rate increased, it would be impossible to say whether the increase was due to this particular change. Many factors could influence this metric at the same time, such as a new advertising campaign for the product.

The marketing department decides to conduct an A/B test. A treatment group of users is shown a product page with videos, and a control group of users is shown the previous version of the product page without videos. By comparing the conversion rate of the treatment group with that of the control group, it is easier to judge the effect of change.

Because HVIT environments are often product-oriented and operate under time pressure in unpredictable markets, A/B testing is particularly relevant to HVIT.

Figure 4.5 shows the contribution of A/B testing to the service value chain.

Table 4.4 outlines the practices for which A/B testing is relevant.

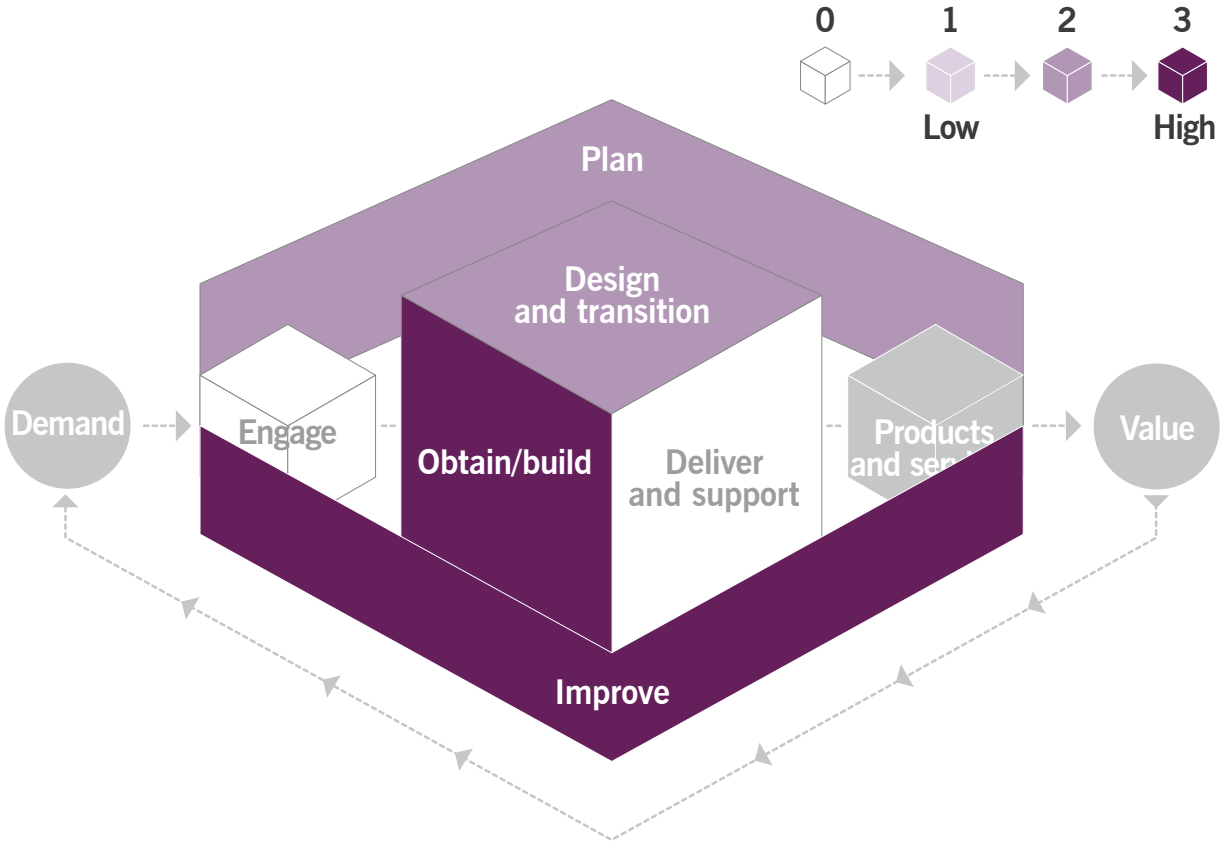

Figure 4.5 Heat map of the contribution of A/B testing to the service value chain

Table 4.4 Practices for which A/B testing is relevant

ITIL management practice	Activities/resources associated with A/B testing	Impact
Portfolio management	Deciding and prioritizing which services, products, and features to invest in using A/B testing data.	H
Risk management	Using A/B testing methods to determine the effectiveness of risk mitigation options before making further investments.	H
Service design	Using A/B testing methods to determine the effectiveness of customer experience and user experience prototypes before making further investments and design decisions.	H
Architecture management	Designing and refining technical, information, product, and service architecture using A/B testing methods.	M
Continual improvement	Using A/B testing methods to determine the effectiveness of various improvement options and initiatives before making further investments.	M
Knowledge management	Using A/B testing methods to determine the effectiveness of different knowledge management, presentation, and communication techniques and tools before making further investments.	M
Organizational change management	Using A/B testing methods to determine the effectiveness of organizational changes before making further investments.	M
Problem management	Using A/B testing methods to determine the effectiveness of workarounds and error control approaches before making further investments.	M
Service validation and testing	Defining and performing service, validation, and product-testing activities using A/B testing methods.	M

The ITIL story: A/B testing

Su: *We developed a new functionality for the app: on every fourth booking made through the app, we will offer customers a complimentary upgrade to a better car.*

Marco: *To gauge the value of this new functionality to our customers, we engaged in an A/B test: 50 per cent of our customers had the upgrade option, and 50 per cent did not. The results were conclusive: 70 per cent of customers upgraded their vehicles when offered the opportunity. Of those who upgraded once for free, 20 per cent chose to hire the higher-spec vehicle on their next booking.*

Su: *Based on these results we can be confident in releasing this new functionality.*

4.2 Techniques for fast development

The fast development objective involves realizing new and improved digital products and services frequently, quickly, and reliably. 'Development' refers to product development in general, although application development is often included in this.

In general, the sooner digital products are delivered, the sooner value can be realized. Sometimes, however, this is not the case, and the schedule should be amended accordingly; for example, an early delivery might not align with market demand. Separating a single product into a series of incremental deliveries enables faster overall delivery, and allows users to realize value earlier than if they wait for the whole product.

In addition to being fast and frequent, delivery must be reliable. However, sometimes it is better to deliver a product or service quickly, or restore service quickly, when there is capacity, than it is to wait to deliver a

marginally more reliable product or service. Mean time to recover service (MTRS) is often a better metric than mean time between failures (MTBF) in these instances.

Fast development has no intrinsic value; its value is related to the value of whatever is being developed. In commercial organizations, it enables a faster return on investment by improving the time to market and the time to customer. This is usually expressed in terms of more sales revenue, as the revenue stream starts earlier. It can also be expressed in the volume of website traffic or the number of potential customers who sign up for mailings, among other things.

Fast development can be measured in terms of the size of the application (change) per unit of time. The size of the application can be expressed in technical units, such as lines of code, or functional units, such as story points or function points. Comparing productivity should be done cautiously, as it depends on many factors: for example, the non-functional requirements are not reflected in story points. Comparing productivity makes more sense when criteria are specified for a certain combination of applications and teams.

Organizations often focus on the fast development of the application because that is what provides the functionality and value. However, it is equally important that related components are also developed or delivered quickly. These include service requests, such as provision of equipment, provision of access rights, setting up new mailboxes, or business intelligence (BI) reports.

It can also be useful to track the predictability of expected development times; for example, by recording when deviations from planning are reported. Managers should encourage people to report potential delays as soon as possible; this is desired behaviour.

Under business pressure to realize value quickly, Agile development teams deliver potentially deployable software increments as frequently as possible. Unfortunately, these releases often have to wait for days, weeks, or sometimes months for actual deployment, which is often the longest delay in the value stream. This usually occurs because of a widespread cautious approach to approval and deployment. Typically, part of the approval process is an assessment by a change advisory board that only meets on scheduled days. The actual deployment also often happens according to a schedule. Therefore, there is a potential conflict between fast development and resilient operations.

The thinking behind this approach is that change is potentially disruptive and therefore should be carefully controlled. Because fast changes and careful changes are diametrically opposed targets, development teams and IT operations often do not collaborate effectively, and IT and service management practitioners have to close the gap.

This tension is based on a familiar mental model in which change disrupts stability, and stability controls change: the fewer the changes, the lower the risk to stability. More recently, a different way of thinking has emerged. A reduction in the size of change reduces the risk of disruption. Smaller changes also mean that change can happen more frequently. By changing more frequently, the organization's capability of changing is improved. Increased capability of change leads, in turn, to lower risk of disruption. Figure 4.6 demonstrates the effect of the size of change.

The DevOps community has embraced this way of thinking, and has developed appropriate practices and supporting technology. Research has found that high performance in terms of frequency of deployment, lead time for change, and change failure rate is correlated with version control, continuous delivery, and automated testing. Research also suggests that change approval based on peer review within the team is no less risky than using a change advisory board.

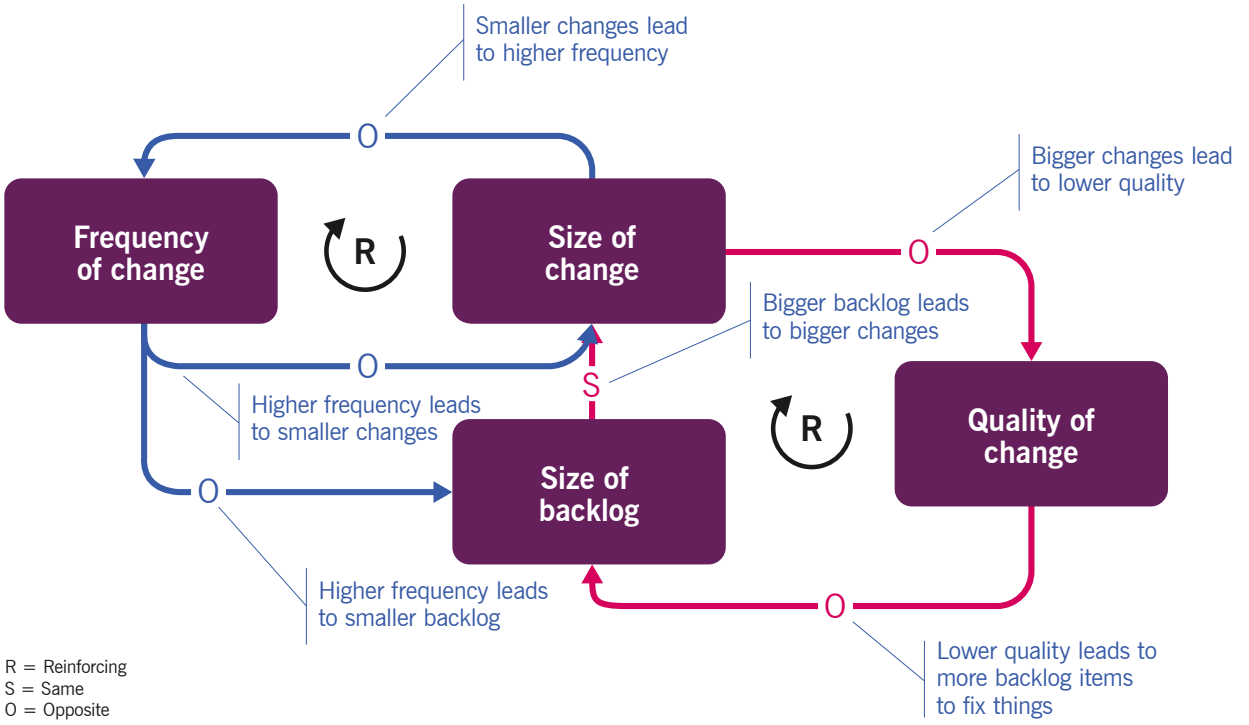

Figure 4.6 Effect of the size of change

Reproduced by kind permission of Oleg Skrynnik

Techniques that can be used to achieve fast development include:

- infrastructure as code
- loosely coupled information system architecture
- reviews
- continual business analysis
- continuous integration/continuous delivery
- continuous testing
- Kanban.

The ITIL story: Techniques for fast development

Solmaz: *We develop new app functionality in increments, releasing improvements and changes regularly and frequently. This helps us to realize value earlier and receive feedback sooner. It also allows us to prioritize the development of new features and support work. Because the changes are small, they require less support and the risk of service disruption is less.*

4.2.1 Infrastructure as code

Infrastructure as code (IaC) enables faster provisioning of environments, contributing to faster development and more resilient operations. Virtualization and hypervisor technology (often provided via the cloud) allows infrastructure items to be created, modified, and removed remotely via programming interfaces. Today, it is common practice to build and configure servers using scripts and configuration files.

IaC is a way of managing and provisioning IT infrastructure and platforms by using machine-readable definition files rather than physically configuring hardware components. These files can then be stored in a version control system (see version control in section 4.3.4).

The concept of idempotence is key to IaC. This means that a deployment command always configures the target environment into the specified state, regardless of what it was previously. This can be achieved by reconfiguring the target environment or replacing it with a new one. In order to do this, changes are made to the environment description and a new version of the configuration model is created. The code is validated and tested to prevent common deployment issues. The release pipeline executes the configuration model, resulting in newly (re)configured target environments. If changes are needed, the source is edited, not the target. Tools such as Vagrant, Ansible, Puppet, and Docker support the whole process. Cloud-based solutions such as infrastructure as a service (IaaS) and platform as a service (PaaS) use IaC definitions to dynamically provision and remove environments. Teams can then provision multiple test environments reliably and on demand. This enables them to test applications in production-like environments earlier in the development cycle.

The choice to use IaC rather than a traditional physical configuration is an architectural design decision with far-reaching consequences. The essence of architecture is choosing which building blocks, or resources, to use, and how to use them. The digitization of infrastructure and platforms enables faster and more reliable provisioning of all necessary environments, such as development, test, and production, contributing to fast development and deployment. It is equally relevant for achieving resilient operations, as it enables quicker recovery from some incidents and prevents others by reducing human error, such as manual misconfiguration and configuration drift. IaC is also more efficient because it has fewer repeatable manual operations, thereby contributing to valuable investments by reducing costs of infrastructure.

Figure 4.7 shows the contribution of IaC to the service value chain.

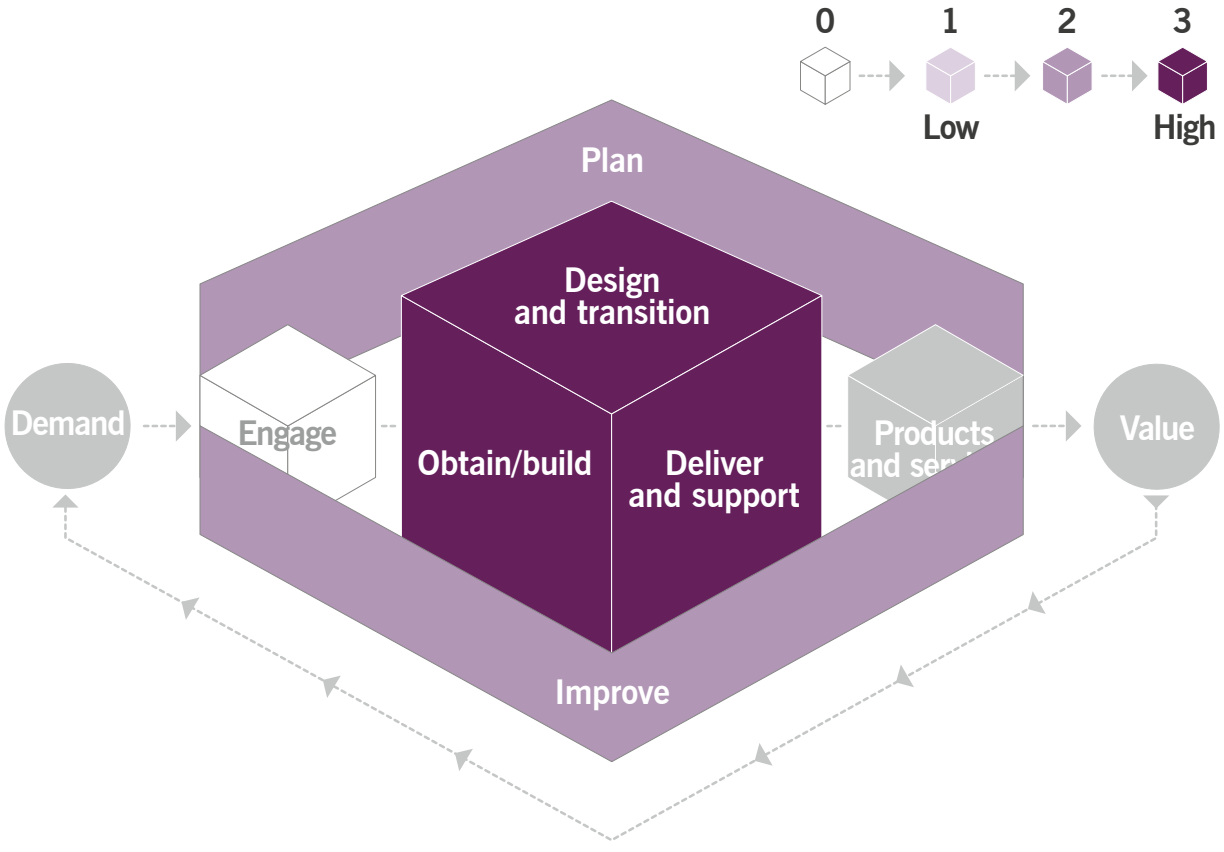

Figure 4.7 Heat map of the contribution of infrastructure as code to the service value chain

Table 4.5 Practices for which infrastructure as code is relevant

ITIL management practice	Activities/resources associated with infrastructure as code	Impact
Architecture management	Verifying architectural decisions and comparing infrastructure solutions.	H
Change enablement	Enabling the fast provisioning or decommissioning of virtual infrastructure components in order to balance speed of delivery with governance, risk, and compliance needs.	H
Deployment management	Automating the deployment of infrastructure, ensuring a faster and more repeatable and reliable deployment of both infrastructure and applications.	H
Information security management	Designing and enforcing security policies in virtual infrastructure components.	H
IT asset management	Tracking the use of commercial software licences assigned to virtual infrastructure components that are often provisioned or decommissioned quickly.	H
Knowledge management	Storing virtual server configuration files and making them available for IaC automation.	H
Service configuration management	Designing and maintaining configuration management databases and relationships at the appropriate level of granularity to track virtual infrastructure components that are often provisioned or decommissioned quickly.	H
Service financial management	Shifting funding from capital expenditure to operating expenditure due to reduced investment in physical infrastructure.	H
Service request management	Automating the provisioning or decommissioning of virtual infrastructure components.	H
Service validation and testing	Designing and maintaining test cases to ensure that virtual IaC meets organizational requirements and policies.	H
Software development and management	Developing software architecture and code to leverage fast delivery/provisioning of virtual hardware infrastructure.	H
Incident management	Automating (where appropriate) incident recovery tasks by leveraging IaC capabilities.	M
Infrastructure and platform management	Fast delivery/provisioning of virtual hardware infrastructure.	M
Problem management	Problem detection and error control leveraging IaC capabilities.	M
Service continuity management	Designing appropriate continuity plans to reflect the organization's use of IaC capabilities.	M
Supplier management	Selecting suppliers who can provide IaC capabilities, or leverage the organization's investments in IaC.	M
Risk management	Recognizing risks introduced or mitigated through the use of IaC.	L

Table 4.5 outlines the practices for which infrastructure as code is relevant.

The ITIL story: Infrastructure as code

Marco: *In test mode, we created several test environments using hypervisor technology on a virtual machine. We wanted to emulate the use of the app on multiple platforms. Because we validated the code at every stage in the development cycle, we knew that the app would continue to work on different devices as it grew.*

4.2.2 Loosely coupled information system architecture

Loosely coupled information system architecture is based on relatively small, independent components. This architecture enables work to be done in small, relatively independent, product- or service-based teams and platform-based teams. By breaking a system down into parts that can be developed and managed relatively independently, teams can focus on their own part and limit their interactions with other teams. The product- or

service-based teams comprise developers and engineers, along with product/service owners, who represent the consumer perspective. This closer collaboration is beneficial for fast development, as well as co-creating value. It also contributes to valuable investments, fast development, and resilient operations (where IT operations are also represented in the team).

One of the biggest problems with tightly coupled architectures (such as in monolithic information systems) is the very low speed of change, as many changes require several parts of the system to be redesigned and redeveloped. Adding more teams and staff to work on the same system may actually reduce speed, as it can result in too many deep interconnections between components on an architectural level.

A closely related technique is application strangulation. This can be applied to gradually create a new application around an old one, slowly replacing old code with new code. This is useful when it is unfeasible to refactor an application (restructuring code without changing its functionality) and it has to be replaced instead. This technique can compromise performance and reliability, so careful monitoring is required if it is used.

Techniques such as microservices and containerization are often associated with loosely coupled architecture.

Definitions

- **Microservices** A variation of the service-oriented architecture in which an application is designed and developed as a set of small, loosely coupled services, each running in its own process and using lightweight mechanisms to communicate.
- **Containerization** The technique of packaging software into standardized lightweight, stand-alone, executable units for development, shipment, and deployment.

The benefits of loosely coupled information system architecture include:

- It facilitates faster development and more frequent changes.
- It enables evolutionary architecture.
- It is more efficient, and leads to less work being needed on redesigning and redeveloping a product or service.
- It enables a switch from a team structure focused on components to a product- or feature-based team structure.

Figure 4.8 shows the contribution of loosely coupled information system architecture to the service value chain.

Table 4.6 outlines those practices for which loosely coupled information system architecture is relevant.

The ITIL story: Loosely coupled information system architecture

Su: *We used a loosely coupled information system architecture for the app. By treating the system as relatively small and independent components, the teams working on each component could work independently because they understood the inputs the component would receive and the outputs that would be required by subsequent components in the workflow. This reduced duplication, complexity, and interdependencies between the different components of the app.*

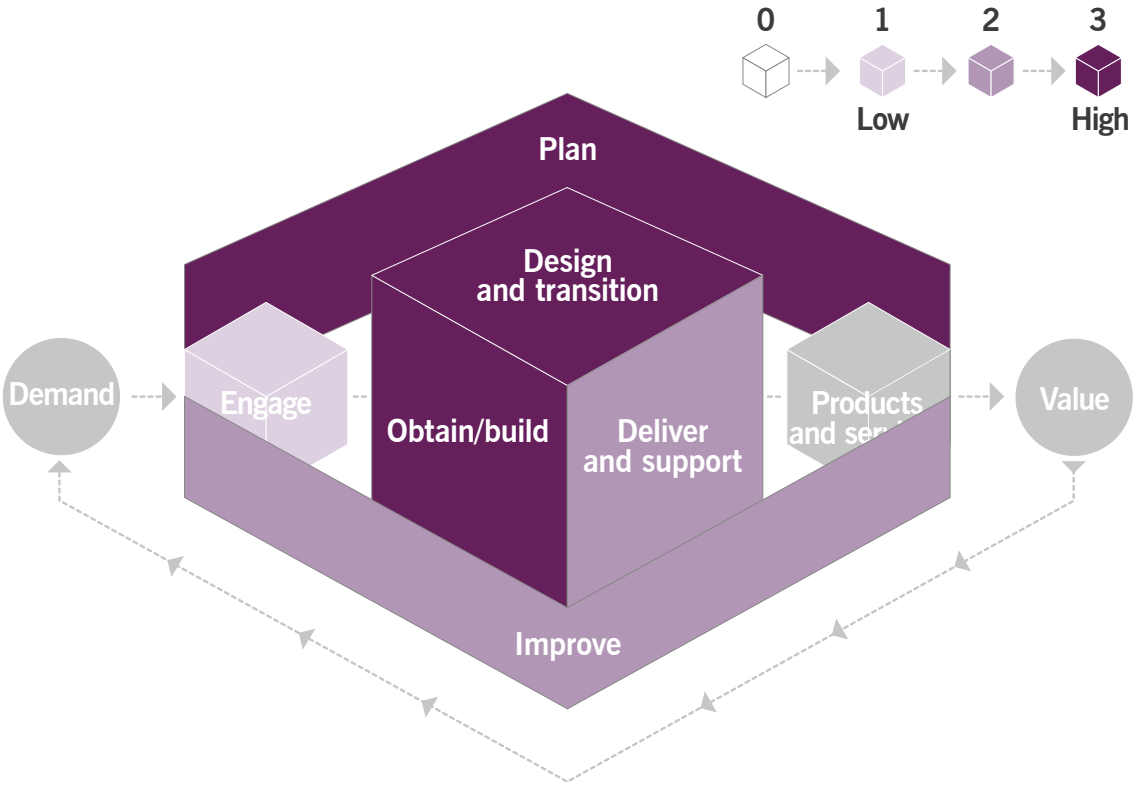

Figure 4.8 Heat map of the contribution of loosely coupled information system architecture to the service value chain

Table 4.6 Practices for which loosely coupled information system architecture is relevant

ITIL management practice	Activities/resources associated with loosely coupled information system architecture	Impact
Architecture management	Designing loosely coupled service, technical, and information architectures.	H
Business analysis	Understanding consumer needs and translating them into detailed requirements for each component of a loosely coupled service architecture.	H
Deployment management	The scope of deployments and deployment patterns is reduced with the decoupling of system architecture; this makes deployments easier to manage and replicate, and strong candidates for automation.	H
Information security management	Designing and managing information security policies for loosely coupled services and service components.	H
Infrastructure and platform management	The detailed design, building, running, and management of loosely coupled infrastructure and platform components.	H
Problem management	Investigating problems and designing error controls that span multiple loosely coupled systems.	H
Service configuration management	Designing and maintaining information on various services and service components and their interrelationships.	H
Service level management	Designing and aligning service levels from a loosely coupled architecture with consumer expectations at the point of service consumption.	H
Software development and management	The detailed design, building, running, and management of loosely coupled software components.	H
Change enablement	Loosely coupled architecture lowers the risks associated with changes to applications and infrastructure, should those changes fail. The lower risk profile results in changes being approved at the team level, rather than at the divisional or organizational level, reducing bureaucracy.	M
Incident management	When incidents are isolated, their investigation and resolution can be planned and performed in a less stressful and more efficient way. The impact of service disruptions in loosely coupled architecture is generally constrained to the unserviceable configuration item (CI) and not the other CIs. This is due to the design principle that system components should be resilient and not depend on available services from other CIs. As a result, incidents have a lesser impact on customers.	M

ITIL management practice	Activities/resources associated with loosely coupled information system architecture	Impact
Service financial management	Loosely coupled system architecture allows the coupling and decoupling of third-party services more easily, enabling the service provider to leverage price reduction and new offers. This flexibility also requires the service provider to employ an Agile costing accounting model to efficiently switch providers.	M
Supplier management	Establishing contracts and managing performance when some components of a loosely coupled architecture are provided by suppliers or external service providers.	M
Strategy management	Decoupling tightly coupled architecture is a strategic-level decision, due to the investment required and the potential operating model implications of taking advantage of it (e.g. introducing autonomous teams). An example of this architecture is service-oriented architecture that may incorporate third-party services as part of the end-to-end service.	L

4.2.3 Reviews

Progressing iteratively with feedback implies having regular reviews of the achievements, to identify lessons to be learned and correct the course of actions if necessary. However, these reviews should not slow down progress or introduce excessive controls.

4.2.3.1 Retrospectives

In Agile, a retrospective is a team meeting at the end of an iteration (or 'sprint') or project to discuss what went well, what could be improved, and how to benefit from the findings in the future. This fast and frequent feedback contributes to fast development. Retrospectives can be applied to most scenarios, and in any area that needs continual improvement. To be effective, they should be scheduled regularly and a facilitator should smooth the contributions of participants. Actions that are input for continual improvement should be identified, and responsibilities delegated. Figure 4.9 shows the contribution of retrospectives to the service value chain.

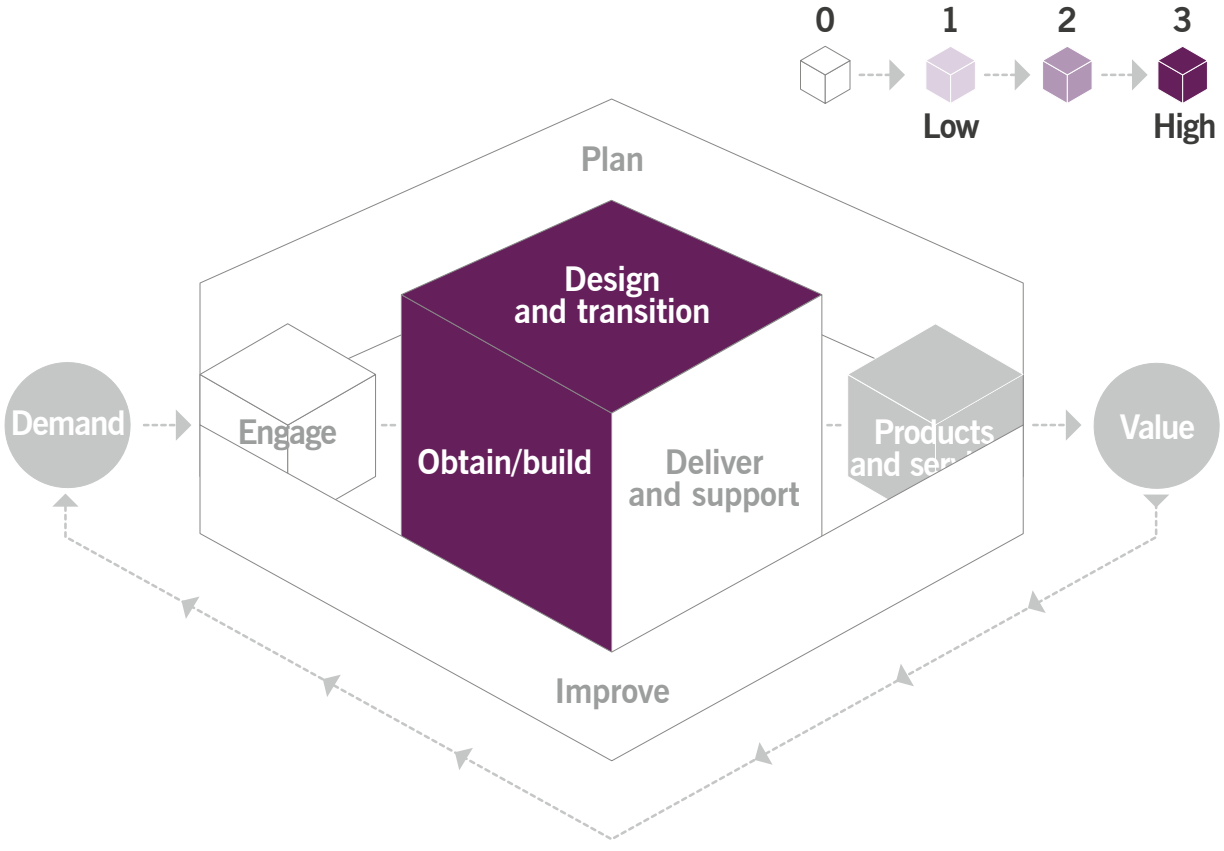

Figure 4.9 Heat map of the contribution of retrospectives to the service value chain

Table 4.7 Practices for which retrospectives are relevant

ITIL management practice	Activities/resources associated with retrospectives	Impact
Continual improvement	Activities and techniques to review continual improvement initiatives periodically or upon completion, to understand whether intended outcomes are being achieved.	H
Project management	Activities and techniques to review project work and learn lessons that can benefit future similar projects.	H
Service level management	Activities and techniques to review and, if needed, modify service levels periodically to understand how to improve.	H
Software development and management	Activities and techniques to review work periodically to understand how to improve.	H
Change enablement	Activities and techniques to review the effectiveness and efficiency of making changes periodically to understand how to improve (including decisions on creating or suspending standard change models).	M
Incident management	Activities and techniques to review incident management work periodically or after major incidents to understand how to improve.	M
Problem management	Activities and techniques to review problem and error controls periodically to understand how to improve.	M
Relationship management	Activities and techniques to regularly review the status and direction of existing relationships with various stakeholders.	M
Risk management	Activities and techniques to review risk mitigations periodically, or after a mitigation has been triggered, to understand how to improve.	M
Service continuity management	Activities and techniques to review the efficiency and effectiveness of continuity plans after recovering from a disaster.	M
Service desk	Activities and techniques to review service desk interactions periodically to understand how to improve.	M
Supplier management	Activities and techniques to regularly review the status and direction of existing relationships with various partners and suppliers.	M

Table 4.7 outlines the practices for which retrospectives are relevant.

4.2.3.2 Blameless post-mortems

Digital technology has increasingly significant social and economic consequences, particularly in HVIT environments. It is therefore increasingly important to prevent incidents. However, complex systems are inherently hazardous: despite all efforts, they will fail. It is therefore also increasingly important to learn from failure.

A post-mortem is a formal record of an incident in terms of its impact, resolution/mitigation efforts, causes, and measures to prevent recurrence. The quality of a post-mortem is better when the participants feel able to share their knowledge and options without fear for their reputation and position. This is why it is crucial to focus on what, rather than who, caused an incident. This blameless culture originated in the healthcare and airline industries, where lives are at risk. Blameless post-mortems are strongly related to safety culture and psychological safety (see section 3.2.2.2).

Definition: Blameless post-mortem

A non-judgemental description and analysis of the circumstances and events that preceded an incident.

Post-mortems are not only useful for the people involved in the incident, but should be shared with and read by others who may benefit from the results. It is also important to share the findings with the broader community, including consumers, as transparency and vulnerability establish trust and help to co-create value.

Good behaviours relevant to post-mortems include:

- conducting post-mortems as part of 'business as usual', not as exceptions
- focusing on what caused an incident, not who
- sharing results broadly and transparently in order to learn and foster trust.

Figure 4.10 shows the contribution of blameless post-mortems to the service value chain.

Table 4.8 outlines the practices for which blameless post-mortems are relevant.

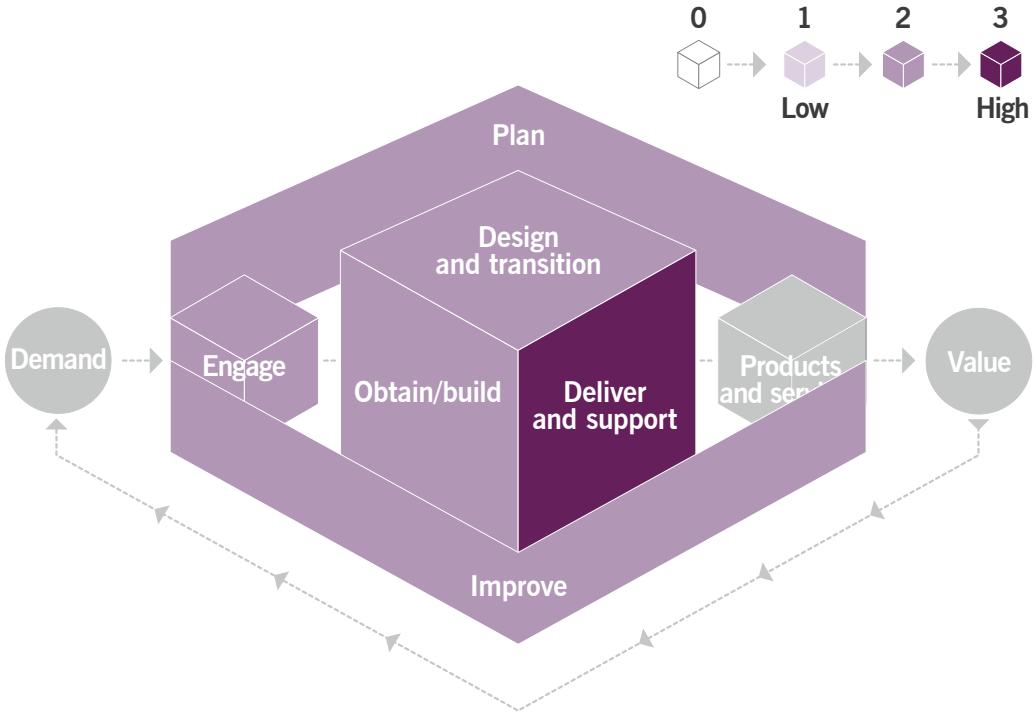

Figure 4.10 Heat map of the contribution of blameless post-mortems to the service value chain

The ITIL story: Reviews

Marco: *Working on the app can be demanding, but we benefit from retrospectives. We regularly analyse finished work in order to understand what lessons can be learned for use in the next sprint. These are blameless post-mortems: we discuss our work openly and honestly, without fear that the cause of an incident will be assigned to any one team member.*

Table 4.8 Practices for which blameless post-mortems are relevant

ITIL management practice	Activities/resources associated with blameless post-mortems	Impact
Change enablement	Investigating successful and failed changes to identify opportunities to improve the success of future changes.	H
Deployment management	Investigating successful and failed deployments of service components to identify opportunities to improve the success of future deployments.	H
Incident management	Investigating incident management (and major incident management) activities to identify opportunities to improve.	H
Problem management	Adjusting the leadership approach to examine the system rather than the people. Blameless post-mortems help organizations to obtain more information about the circumstances related to incidents. This provides better information for problem identification and investigation.	H
Project management	Investigating project activities to identify lessons learned and opportunities to improve future projects.	H
Release management	Investigating successful and failed releases of services to identify opportunities to improve the success of future releases.	H
Continual improvement	Investigating continual improvement activities to identify lessons learned, understand changes in the system, and identify opportunities to increase or maintain the success of future improvements.	M
Information security management	Obtaining more information about the circumstances related to security breaches and security incidents.	M
Risk management	Assessing risk mitigations after they have been triggered to understand how to improve mitigations and responses to current and future risks. Amending risk registers and exploring new areas of possible risks.	M
Service desk	Investigating engagement with external stakeholders to identify opportunities to improve interactions and ongoing communications.	M
Service validation and testing	Investigating successful and failed validation and testing activities to identify opportunities to improve the success of future efforts.	M
Software development and management	Capturing lessons learned and ideas for improvements from sprints. Involving partners and suppliers to gather feedback for the learnings register.	M

4.2.4 Continual business analysis

Rapidly changing, unpredictable HVIT environments require continual adjustments in response to fluctuating market demand. This has implications for business analysis.

When an investment decision has been made, it is crucial to verify any initial assumptions that were made about product or service specifications, characteristics, and features. Some providers ignore the need to interact with real users as soon as possible, and spend several months or even years on development before presenting their solution to customers and users. Often, when this approach is taken, some features of the product or service are unnecessary, some need significant readjustment, and others that would be highly valuable are missing. However, the organization's resources and time have already been spent, and marketing opportunities are shrinking.

A better way to develop is to implement feedback loops as early as possible. Proper feedback loops can provide valuable information about the direction in which the product or service should be developed. For a feedback loop to be useful, it is important to develop iteratively. A minimum viable approach (see section 4.1.2) could be used for this purpose. Figure 4.11 illustrates this concept.

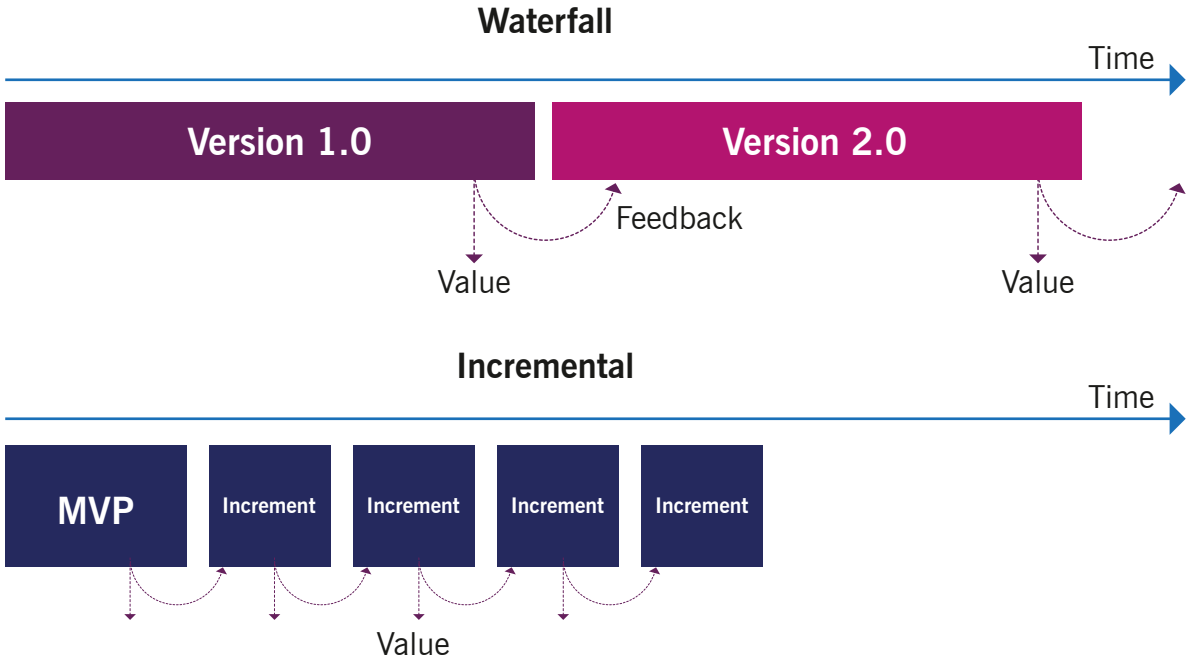

Figure 4.11 Faster value realization with an iterative approach

In the past, requirements were always defined at the start of a project in formal discussions between the stakeholders and a business analyst who had been assigned a temporary, project-based task. These requirements would then become the basis for development activities.

Increasingly, however, products are developed and improved based on feedback. The feedback can be reported by users or observed indirectly. Analysis and the specification of requirements happens on an ongoing basis. Business analysis is often no longer performed by a dedicated business analyst; it is a role that can be fulfilled in combination with other roles. When this approach is used, further development can be challenging because the product's architecture has already been established. Therefore, pre-development analysis should consider these future concerns and create a flexible architecture. Post-development analysis differs from pre-development in the sense that it is less focused on creating an architecture and more focused on working effectively within the system's architectural constraints.

Figure 4.12 shows the contribution of continual business analysis to the service value chain.

Table 4.9 outlines the practices for which continual business analysis is relevant.

The ITIL story: Continual business analysis

Radhika: *The ITIL guiding principles are useful for every team in the business. For example, the focus on value principle fits neatly onto business analysis. Circumstances continually change, and features that are in development may become lower or higher priority, depending on how their value is affected. For example, new legislation that affects app use or data storage would automatically become a high priority.*

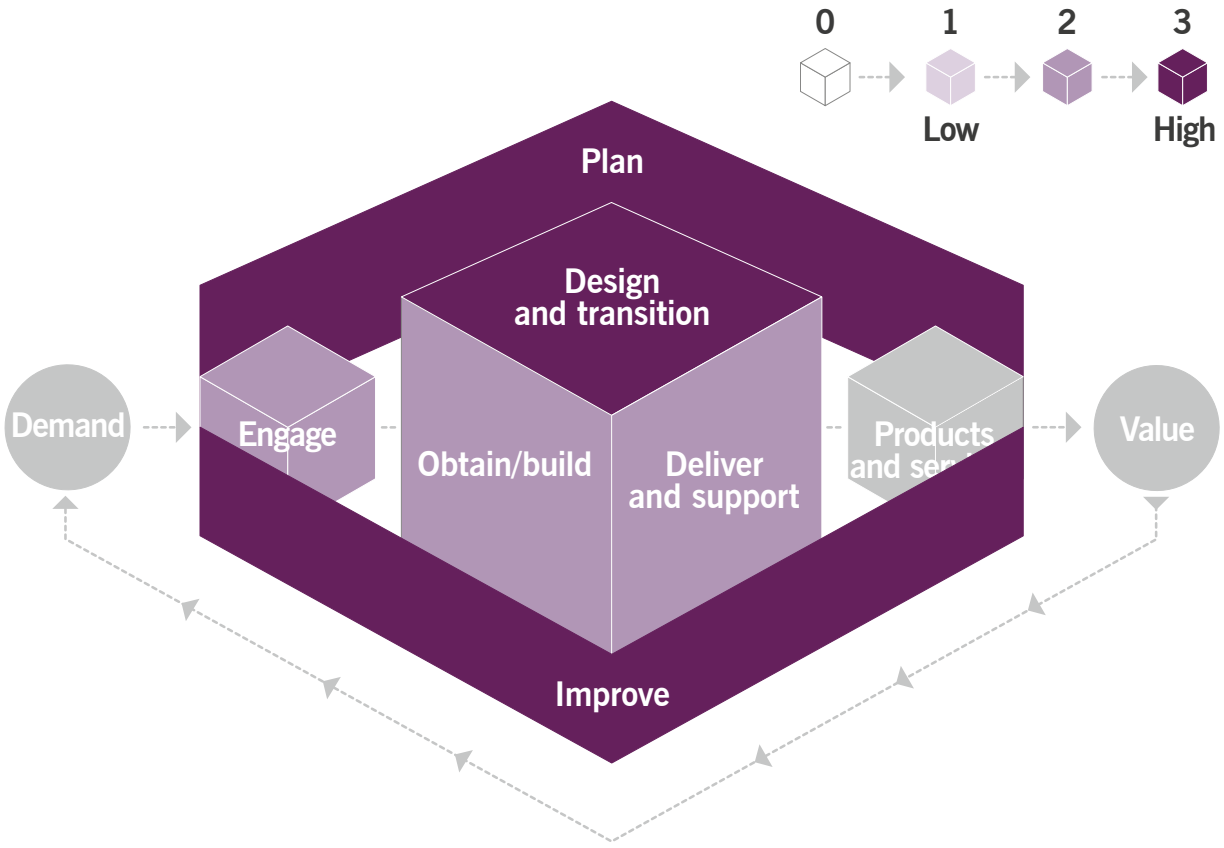

Figure 4.12 Heat map of the contribution of continual business analysis to the service value chain

Table 4.9 Practices for which continual business analysis is relevant

ITIL management practice	Activities/resources associated with continual business analysis	Impact
Business analysis	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's products and services.	H
Infrastructure and platform management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's infrastructure and platform products.	H
Portfolio management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the way the organization invests in various products and services. Monitoring ongoing portfolio investments to identify value leakage or to verify value co-creation.	H
Project management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's projects and associated business cases.	H
Relationship management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's relationships with external stakeholders.	H
Risk management	Continually scanning internal and external corporate risks to assess and understand their impact on the organization's products and services, and to design appropriate mitigations and countermeasures.	H
Service design	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the customers' and users' experiences of the organization's products and services.	H
Software development and management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's software products and the prioritization of ongoing software development work.	H
Strategy management	Continually monitoring and evaluating customers, market conditions, and the broader ecosystem to understand their impacts on the organization's products and services, and adjust the strategy accordingly.	H

ITIL management practice	Activities/resources associated with continual business analysis	Impact
Architecture management	Continually scanning the use of technology within the organization and by external stakeholders to understand its impacts on the organization's technical, service, and information architecture.	M
Knowledge management	Continually scanning customers, market conditions, and the broader ecosystem to create a shared understanding among relevant stakeholders.	M
Service continuity management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's continuity and disaster recovery measures.	M
Supplier management	Continually scanning customers, market conditions, and the broader ecosystem to understand their impacts on the organization's relationships with partners and suppliers.	M

4.2.5 Continuous integration, continuous delivery, and continuous deployment

Continuous integration, continuous delivery, and continuous deployment (CI/CD) are descriptive terms for a collection of practices primarily associated with software engineering, which are central to the philosophy of Lean and Agile software development. The adoption of these practices has grown rapidly, and it is important to understand the defining characteristics of CI/CD and the wider context of evolving system development practices when implementing services that are underpinned by software development.

Definitions

- **Continuous integration** An approach to integrating, building, and testing code within the software development environment.
- **Continuous delivery** An approach to software development in which software can be released to production at any time. Frequent deployments are possible, but deployment decisions are taken case by case, usually because organizations prefer a slower rate of deployment.
- **Continuous deployment** An approach to software development in which changes go through the pipeline and are automatically put into the production environment, enabling multiple production deployments per day. Continuous deployment relies on continuous delivery.

CI/CD contributes to fast development and, because the deployments are more reliable, to resilient operations.

Continuous integration is the automation achieved in the build and code testing process each time the code is committed, which results in changes to version control. Continuous integration promotes the practice of code sharing in the software development community by merging changes into a centrally shared code repository. The code commit triggers the automated-build functionality to extract the latest code commit from the shared version control repository and build, test, and validate the full master branch (that is, the trunk).

Continuous integration emerged as a practice because of the isolated nature of software development, whereby developers require constant integration points between their changes and the rest of the development team's code base. Long waiting cycles cause merge conflicts, bugs with difficult resolution paths, diverging code strategies, and duplicated efforts. One of the main aims of continuous integration is to keep the master branch free of exceptions and errors. Software development teams can establish branch policies to ensure that the master branch meets certain pre-agreed and pre-authorized quality criteria.

Continuous delivery is focused on the process of building and testing new releases and promoting them to the production environment. This requires staging environments, which become part of a release pipeline to automate infrastructure provisioning and the deployment of new releases.

Continuous delivery is a Lean practice; its goal is to keep an incident-free production environment running well. This is achieved by discovering the shortest path from the availability of new code in the version control repository to packaging management for deployment purposes.

Continuous delivery relies on the concept of progressive deployment rings for a deployment model. The first deployment ring is often a 'canary release' in a production environment that is accessible to the organization's internal IT team or other relevant groups of internal users. Consecutive deployment rings welcome wider user populations. Those subsequent deployment rings may require an approval point by a key decision-maker. If an organization uses these rings for deployment, it usually follows the same approach for release management.

Another deployment model uses the concept of 'blue/green deployment', whereby the existing version keeps running in one production environment (known as 'blue') while a new version is deployed to an identical 'green' environment. Typically, load balancing is used to redirect a limited number of users to the 'green' version. If an incident occurs, traffic can be re-routed to the 'blue' version. Otherwise the 'green' environment becomes the default one and the 'blue' environment is used for the next deployment.

Figure 4.13 shows the contribution of CI/CD to the service value chain.

Table 4.10 outlines the practices for which CI/CD are most relevant.

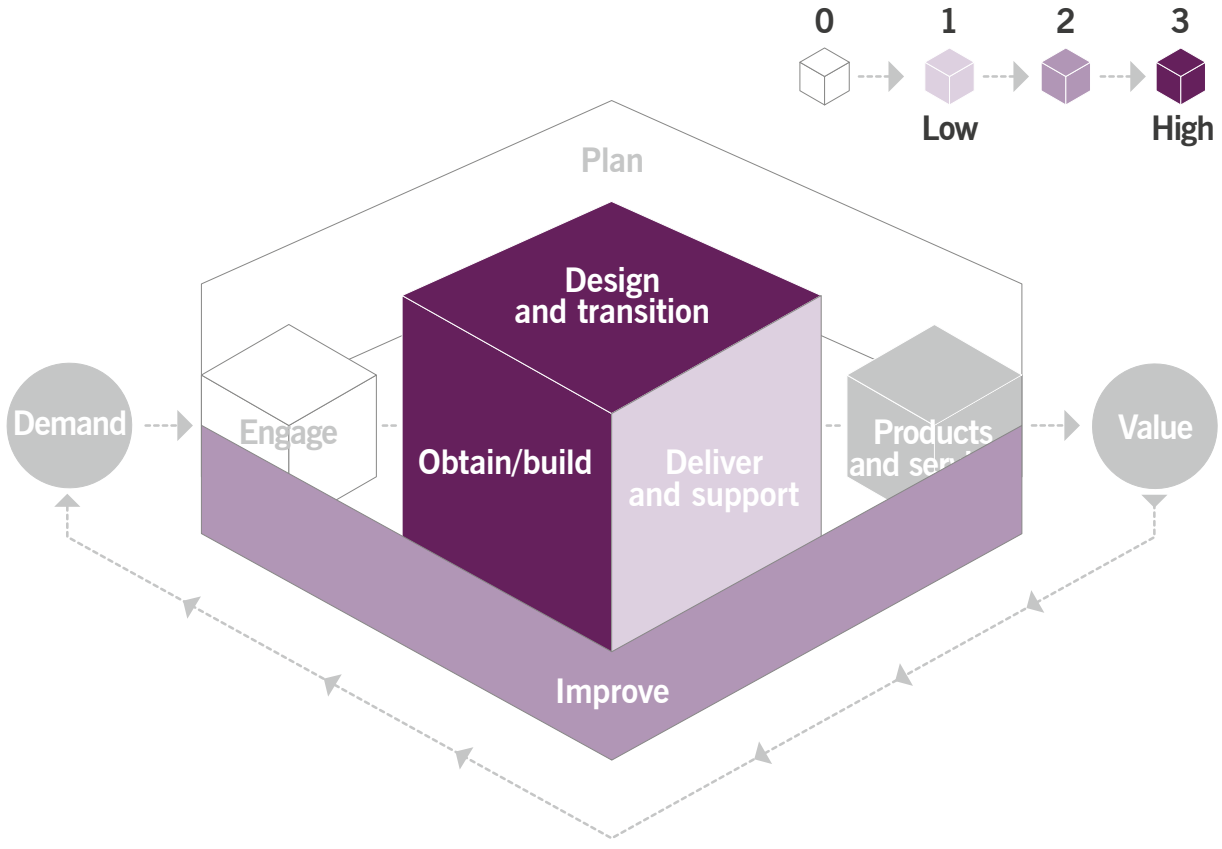

Figure 4.13 Heat map of the contribution of CI/CD to the service value chain

Table 4.10 Practices for which CI/CD are most relevant

ITIL management practice	Activities/resources associated with CI/CD	Impact
Change enablement	The rate of change to the organization's products and services can be adjusted by using a CI/CD pipeline in line with business needs and expectations.	H
Deployment management	The systematic/automatic installation of specific versions or packages of software to a predetermined environment (integration, user acceptance testing, production).	H
Infrastructure and platform management	Use of CI/CD techniques for digital infrastructure.	H
Release management	Recognition that the deployment of software and the release of functionality are often distinct activities that help to plan and manage releases.	H
Service configuration management	Managing code repositories and the associated tools that form parts of a CI/CD toolchain, and continually updating CMDB(s) to reflect when significant (CI-level) changes have been made.	H
Service validation and testing	Developing automated test cases to support continual integration activities.	H
Software development and management	Use of CI/CD techniques that are used for application software.	H
Architecture management	Designing and improving service, technical, and information architecture to leverage CI/CD capabilities. Containerization is an architectural choice that supports CI/CD.	M
Information security management	Ensuring compliance with information security policies by reducing manual work. Increasing the scale and scope of automation by leveraging CI/CD tools may help to ensure compliance with information security policies by reducing manual work and improving the traceability of changes.	M
Knowledge management	Continually updating knowledge bases to ensure that the organization maintains an up-to-date understanding of the software being built and deployed through CI/CD pipelines.	M
Service continuity management	CI/CD pipelines should be set up to push software components to continuity and disaster recovery systems.	M
Risk management	Reducing the impacts of certain types of enterprise risks through the use of CI/CD automation.	L

The ITIL story: Continuous integration, continuous delivery, and continuous deployment

Marco: *We have created identical build, test, and live environments for the app, which allow us to continually integrate and deliver new code that is compatible with the existing codebase. As a result, we can develop the app at a high cadence with code that already works. We also have fewer incidents caused by bugs that could have difficult resolution paths.*

4.2.6 Continuous testing

Software testing is not just about testing software that is developed and operational. Testing is something that should be conducted throughout the entire software development lifecycle. Table 4.11 outlines the different types of testing.

Table 4.11 Types of software testing

Testing the ideas	All software originates with an idea, often an attempt to solve a problem. Testing the idea helps to determine its quality from a customer perspective (based on wants and needs) and a business perspective (based on metrics, including growth, revenue, conversion, and user base).
Testing the artefacts	Artefacts, such as epics, user stories, acceptance criteria, data flow diagrams, and process flow diagrams, should be tested. Inspecting the information within the artefacts can help to identify ambiguities, assumptions, and information relating to risks and variables. This information can be used as feedback to help review and improve the artefacts.
Testing the user experience and user interface designs	Artefacts are used for design activities, programming activities, and testing activities. The design activities produce more design artefacts than can be tested. Interfaces and experience are tested using relevant artefacts, and test results may affect other artefacts.
Testing the architecture and code designs	When designing software architecture and discussing how to build it and new features, exploratory testing can enhance the designs and ideas.
Testing the code	Code reviews are an important part of building a high-quality product. Anyone should be able to read the code and test it from different perspectives. Unit testing verifies that each unit of the software performs as it should. Unit tests are often automated. It is important to note that this is the first point within the software development lifecycle where there should be something tangible to measure against.
Testing the operational software	Testing the operational software is the most common activity that is associated with testing. Many Agile teams include 'testing' as a status within their workflow after development. However, testing should not be a status in a workflow; it should be a series of structured activities conducted throughout the software development lifecycle.
Testing in different environments	When it comes to test environments, a risk-based approach may be appropriate. There are many risks that can be tested for. Some of these cannot be tested for in a development environment. For others, a stringently integrated environment is needed for testing. Some risks can only be tested for in the real production environment.
Testing the release pipelines	Pipeline processes can be tested, and teams often do so implicitly in order to enhance their pipelines to be as efficient and fast as possible. This requires a good understanding of the structure behind pipeline testing.
Testing the system in production	Some risks must be tested for in the production environment, such as performance risks (particularly user load and user stress risks), user acceptance testing (where users use the system as they would the live software), and observability risks (to test the effectiveness of observability solutions and implementations). When testing in production, the aim should be to minimize the risk of testing affecting the customers. Strategies to achieve this include: ● **Canary releases** The new feature is initially released to a small, targeted group of users, gradually increasing to all users. ● **Feature toggles** Features can be hidden behind a feature flag, easily enabled or disabled by toggling the flag on or off. ● **Blue/Green deployments** Two identical production environments run simultaneously (one 'blue' and one 'green'), with only one of the environments being live, serving all production traffic, whereas the other is used for deployment of the new versions. ● **Automated roll-back strategies** Automation tools can quickly revert the release to its previous version in the event of an incident.
Testing the services in production	There are many services, activities, and technologies that are implemented for released production software, which can all be tested. From a service delivery perspective, it is important to test processes. For example, when testing whether a user can access support, it is important to ask, how easy is it for users to get support? What opportunities do users have within the software to do that?

To achieve continuous testing, a number of principles[12] should be in place, including:

- Tests should be written at the lowest level possible.
- Tests with the fewest external dependencies should be favoured.
- Write once, run anywhere, including in the production system.
- Products should be designed for testability.
- Test code is product code; only reliable tests survive.
- Testing infrastructure is a shared service.

- Test ownership follows product ownership.
- Fault injections and chaos engineering in production should be practised to test service resilience.
- Unreliable tests should be eliminated.

Figure 4.14 shows the contribution of continuous testing to the service value chain.

Table 4.12 outlines the practices for which continuous testing is most relevant.

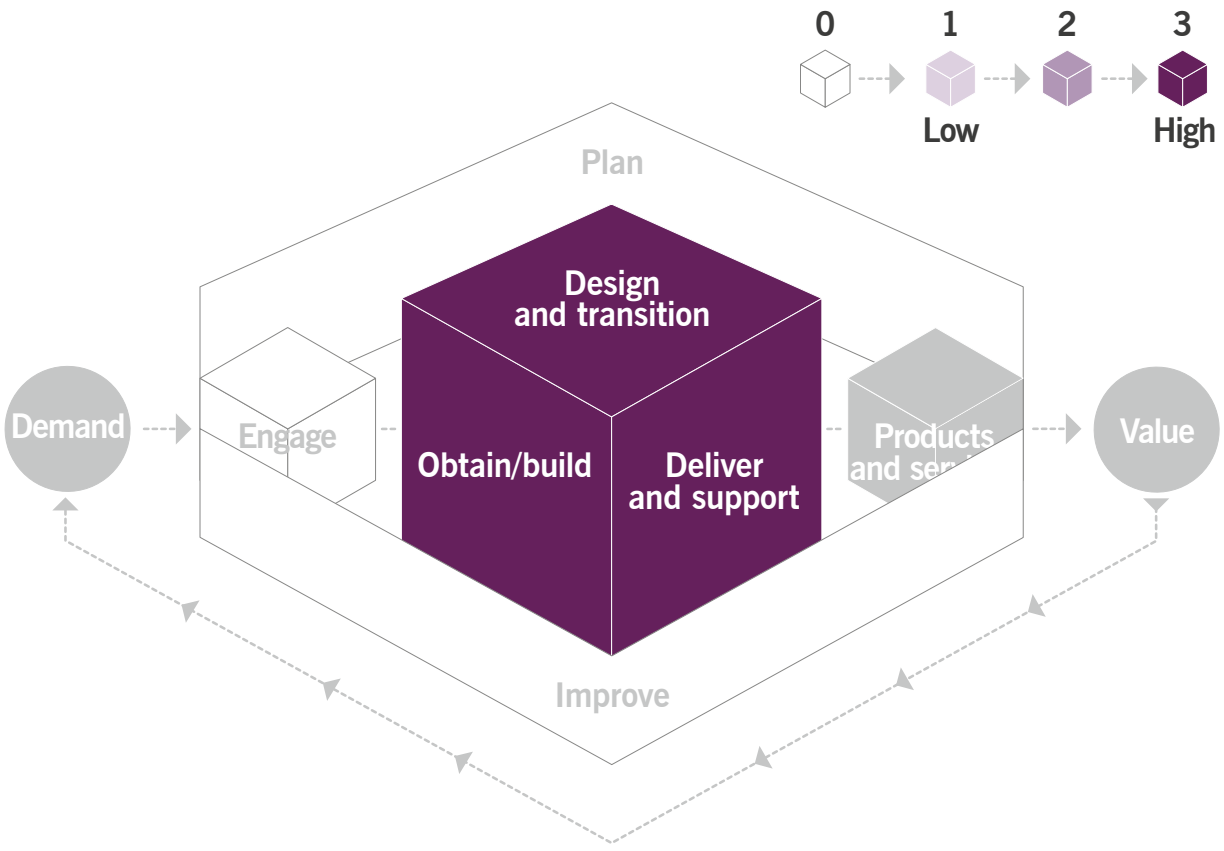

Figure 4.14 Heat map of the contribution of continuous testing to the service value chain

Table 4.12 Practices for which continuous testing is most relevant

ITIL management practice	Activities/resources associated with continuous testing	Impact
Architecture management	Designing and improving service, technical, and information architecture to leverage CI/CD capabilities.	H
Service validation and testing	Unit, integration, and regression testing is conducted on an ongoing basis throughout the development lifecycle. This includes application unit testing, infrastructure service testing, functional/non-functional testing, canary releases, blue/green testing, and infrastructure security testing.	H
Deployment management	Changes or deployments that cause the continuous testing to fail trigger the team's Andon cord. The team members then swarm to resolve the issue.	M
Information security management	Ensuring compliance with information security policies by reducing manual work. Leveraging automated testing tools may help to ensure compliance with information security policies by reducing manual work and improving the traceability of changes.	M
Problem management	Automated tests help to verify problem resolution, the presence of known errors, or the effectiveness of workarounds.	M
Service continuity management	Automated testing can accelerate the provisioning of technical resources needed to recover from a disaster.	M
Risk management	Reducing the impact of certain types of enterprise risks using test automation.	L

The ITIL story: Continuous testing

Su: *Every product we release to market is thoroughly tested to ensure that it is fit for purpose and fit for use. The app improvements are no different. We tested:*

- *the initial ideas for the new functionality to ensure that they had the potential to achieve our objectives*
- *epics, user stories, and acceptance criteria*
- *improvements to the interface design to ensure it was intuitive and easy to use*
- *code designs and software architecture to ensure they were robust*
- *code to fix bugs and glitches that were introduced during development*
- *the system and the software in production.*

At every stage, we revisited previous decisions if tests suggested that we had introduced significant inefficiencies or defects.

4.2.7 Kanban

Kanban is a set of principles, practices, and regular activities aimed at developing and managing a predictable, rhythmic, and constant flow of work. When applied properly, it can significantly accelerate the development of high-quality products and services. The pull-based triggering mechanism allows customers to pull work through the value stream. Pull-based working also has the advantage that work is not pushed onto people, burdening them unnecessarily and unproductively. This is valuable in Lean teams, where overburden is a form of waste.

Definition: Kanban

A Lean method based on a highly visualized pull-based workflow that manages and improves work across human systems by balancing demands with available capacity, and by improving the handling of system-level bottlenecks.

The main practices of Kanban are:

- visualizing work
- limiting work in progress
- managing flow
- making process policies explicit
- implementing feedback loops
- improving collaboration
- evolving experimentally.

Sometimes organizations only use Kanban boards to visualize work in progress. Although using a board is important, it is a limited application of Kanban. The power of Kanban relies upon holistic implementation and constant attention to the workflow. An example of a Kanban board is shown in Figure 4.15.

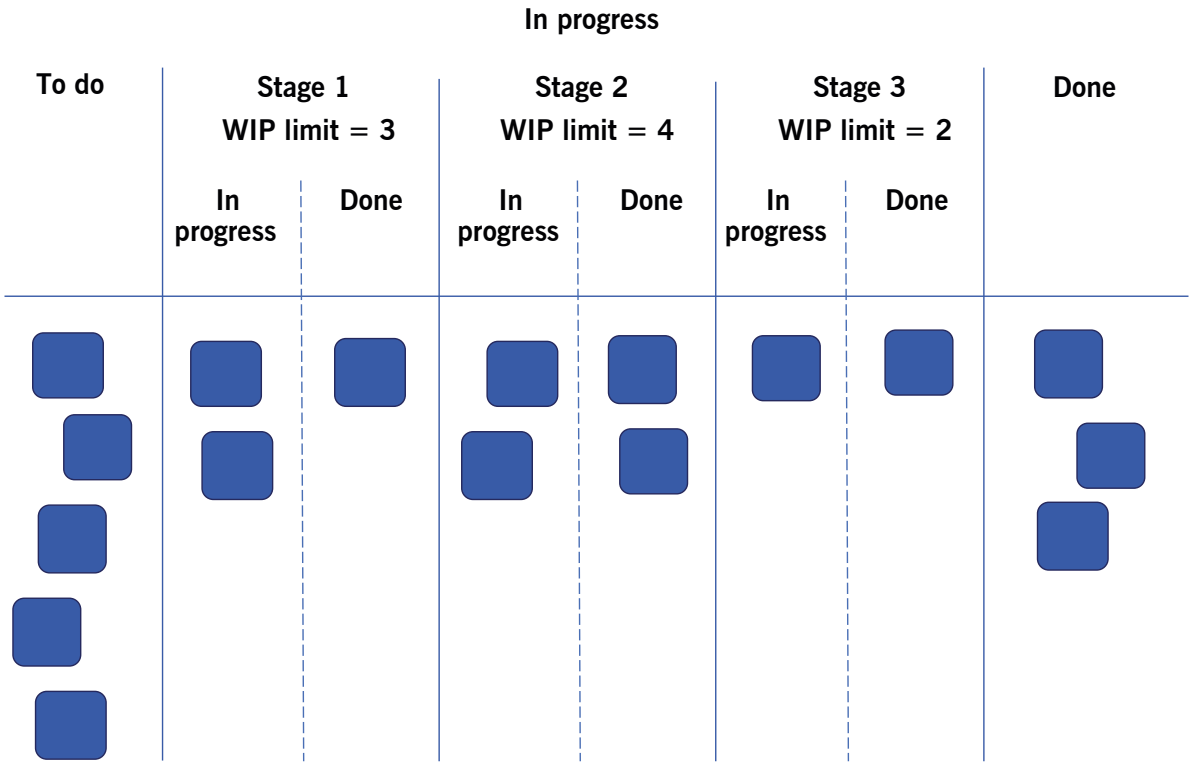

Figure 4.15 An example of a Kanban board

Kanban suggests a system of regular meetings that ensure effective communication. This system is often referred to as 'Kanban cadences'. The frequency of cadences is specific to an organization and should be defined and adjusted case by case. The meetings are:

- strategy review
- operations review
- risk review
- service delivery review
- replenishment meeting
- delivery planning meeting.

Figure 4.16 shows the contribution of Kanban to the service value chain.

Table 4.13 outlines the practices for which Kanban is relevant.

The ITIL story: Kanban

Radhika: *We used a Kanban board to visualize workflows for our app development so that we could track bottlenecks. These can often be eliminated with extra resources or by redesigning the workstream. The visual nature of a Kanban board enables colleagues and stakeholders outside the core development team to understand how work is progressing, which allows them to plan better and create more value.*

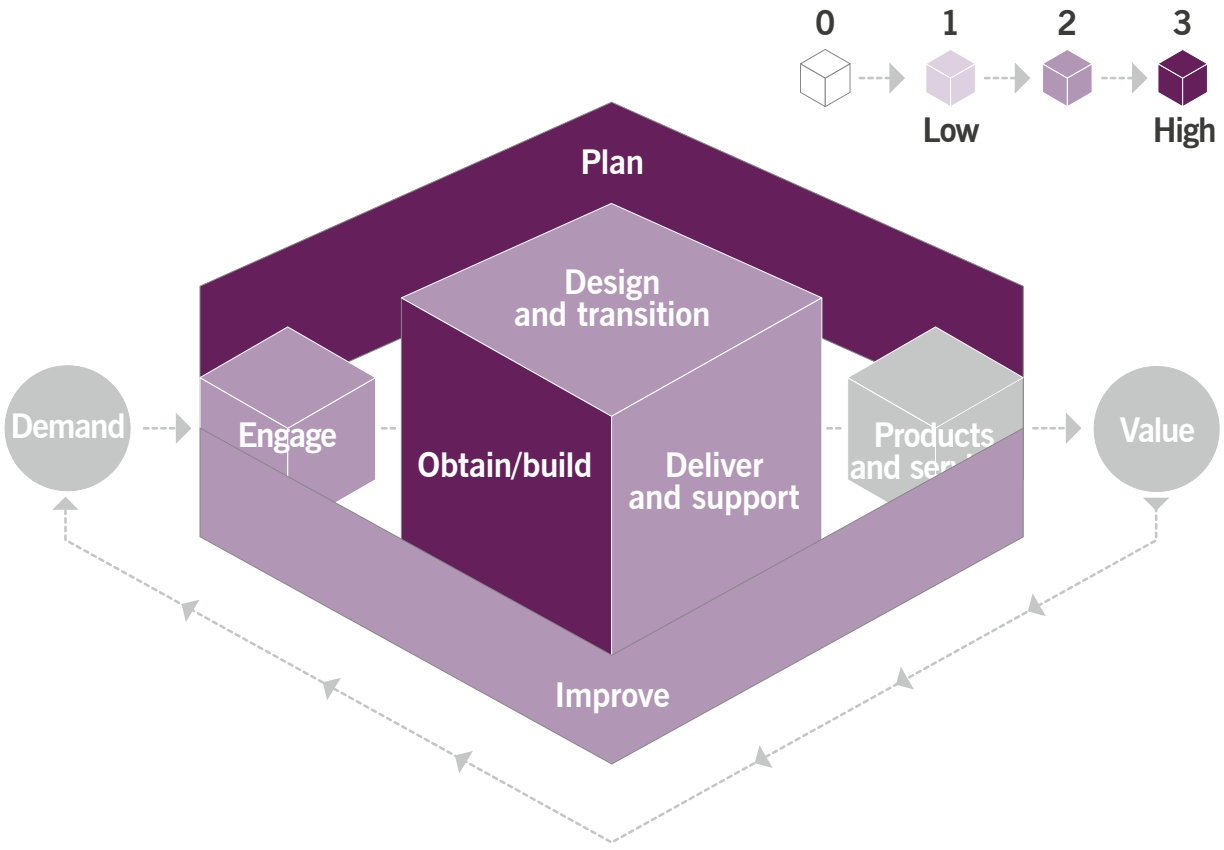

Figure 4.16 Heat map of the contribution of Kanban to the service value chain

Table 4.13 Practices for which Kanban is relevant

ITIL management practice	Activities/resources associated with Kanban	Impact
Change enablement	Visualizing and improving the flow of changes to services and service components by limiting work in progress.	H
Continual improvement	Visualizing and improving the flow of improvements into the SVS.	H
Project management	Visualizing and improving the flow of work across projects and teams.	H
Release management	Visualizing and improving the quality of releases to consumers.	H
Software development and management	Visualizing and improving the flow of new or changed software components into live environments.	H
Incident management	Visualizing and improving the speed and quality of incident resolution by limiting work in progress.	M
Portfolio management	Visualizing and improving the flow of investments across the portfolio pipeline(s).	M
Problem management	Visualizing and improving problem and error control by limiting work in progress.	M
Supplier management	Visualizing supplier onboarding/offboarding progress.	M

4.3 Techniques for resilient operations

The resilient operations objective involves ensuring that digital products are available for use whenever needed.

The potential value of digital investments can only be realized when the digital products and services invested in are available for use. Fulfilment of the non-functional requirements provides warranty, and reduces the risk that issues will adversely affect the utility of the products and services.

Information systems increasingly rely on so many components that behaviour often cannot be predicted or guaranteed. Failsafe systems are an illusion; organizations must be prepared for inevitable and unexpected failure. The emphasis is no longer on maintaining a long interval between failures; it is on restoring service quickly when inevitable issues do occur. This reduces the disruption to business operations.

Resilience applies to all parts of the systems stack, and also to the organization that manages these component parts. It is only when every component is resilient that the consumer-facing parts are resilient. Resilient operations add nothing to the potential value of an investment; rather, they ensure that its potential value can be realized. Because information systems are complex and therefore intrinsically error-prone, resilience is about damage limitation. Depending on the nature of the systems, the damage can be expressed in terms of lost revenue, lower prices, incurred costs, and reputational harm. For example, when an e-commerce website suffers from poor availability or performance:

- revenue is lost if customers turn to other providers
- there is pressure to lower prices because of reduced customer satisfaction
- costs are incurred for restoring the service, implementing workarounds, and communicating with consumers
- reputational harm is incurred when the provider is perceived to deal with incidents poorly; for example, by not taking appropriate action, not caring, withholding information, or not learning from incidents.

The consumerization of IT has led to an expectation that corporate IT systems will be available at any time and everywhere. Resilience can be drastically improved by the cloud capabilities that are now available at a fraction of the cost of on-premise systems. Resilient systems and services are no longer a choice, they are a realistic expectation due to the cloud and the consumerization of IT.

Resilient operations can be measured in terms of availability, performance, and security. Availability is measured as a percentage alongside a specification of the mean time between failure and the mean time to restore service. Availability is notoriously unreliable as an indicator of its effect on service consumers. It is also difficult to measure; for example, when a system is partially available.

Performance is measured in various ways; for example, the time for a web page to load, a data query to execute, or a batch process to complete. Security can be measured in terms of security breaches, but this only measures the breaches that have been detected. Better indicators are the maturity of the monitoring of controls, and the ability to analyse log information to identify risks and breaches.

Techniques that can be used to achieve resilient operations include:

- technical debt
- chaos engineering
- definition of done
- version control
- AIOps
- ChatOps
- site reliability engineering.

The ITIL story: Techniques for resilient operations

Henri: *Our app must be reliable and consistent, or it will be regarded as defective by our customers. We also need to make sure that our team is resilient and can adapt to different conditions should their ways of working need to change.*

4.3.1 Technical debt

Definition: Technical debt

The total rework backlog accumulated by choosing workarounds instead of system solutions that would take longer.

In software development and management, technical debt is the total backlog of rework needed to repair substandard (changes to) software. Generally, when software is used, enhancements and repairs will follow. When these changes are applied, the quality of the software will decline unless effort is made to limit the damage. This is known as 'software entropy', which is particularly relevant to HVIT because of the large investments in software, the frequency of changes, and the need to change quickly to keep up with market demand.

Technical debt is the backlog of work that is needed to fix damage that is incurred while enhancing or repairing the software. Such damage is usually caused by a combination of limited time, money, knowledge, and skills. For example, organizations that are under pressure to restore normal service when software fails may apply an easy workaround to quickly resolve the issue. There is often intent to fix it properly later, but in many cases, this does not happen. The intended 'proper fix' may not be formally recorded and will be subsequently forgotten.

Technical debt can also occur during the initial development of an application, where deadlines or budgetary constraints lead to shortcuts. Almost every project has a legacy, which sometimes surprises its benefactors.

Technical interest is the sum of the cost involved in applying changes to software that is difficult to change because of the compromised quality, plus the cost of incidents caused by these imperfections. As long as the interest is not prohibitive, technical debt can be perfectly acceptable. However, it tends to accumulate, and the higher the debt, the more risk is involved. This risk must be managed.

Reducing technical debt results in fewer incidents, and therefore contributes to resilient operations. It also contributes to valuable investments and fast development. A balance between these benefits should be achieved and maintained. It is therefore prudent to recognize the existence of technical debt, qualify and quantify it as best possible, assess the associated short-term and long-term risks, and make appropriate decisions as part of (software) product management, including allocating a budget to 'repay' the technical debt (see also the concept of error budget in section 4.3.7).

Figure 4.17 shows the contribution of technical debt to the service value chain.

Table 4.14 outlines the practices for which technical debt is relevant.

The ITIL story: Technical debt

Henri: *The app development work will re-use much existing code; therefore we will accrue some technical debt. As our app grows, we may need to implement workarounds to accelerate launches, but these could make the code susceptible to incompatibilities at a later date.*

Marco: *The more work we do to improve the original code, the more resilient it will be and the less technical debt we will accrue.*

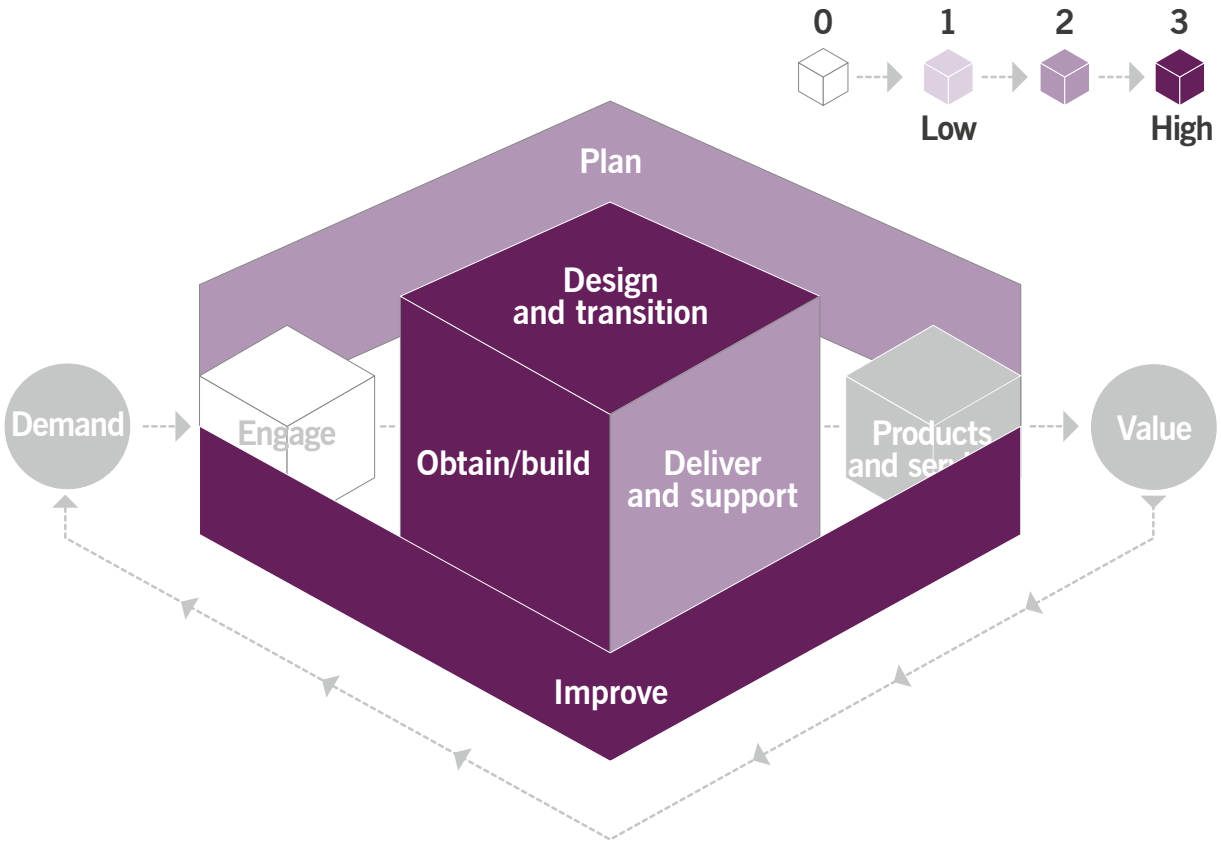

Figure 4.17 Heat map of the contribution of technical debt to the service value chain

Table 4.14 Practices for which technical debt is relevant

ITIL management practice	Activities/resources associated with technical debt	Impact
Incident management	Incident resolution and management requires knowledge of existing technical debt and the efforts planned to resolve it.	H
Infrastructure and platform management	Identifying and reducing technical debt by creating or modifying infrastructure and platform service components.	H
Knowledge management	Ensuring that all relevant stakeholders have access to up-to-date information.	H
Portfolio management	Deciding whether to invest resources to fix the technical debt present in live products and services, and understanding the impact on investments towards future products and services. Assessing technical debt so that new portfolio items can be introduced to the existing pool. Assessing the technical debt of current portfolio items to prevent value drain.	H
Problem management	Applying problem control and error control methods to manage technical debt.	H
Software development and management	Identifying and reducing technical debt by creating or modifying infrastructure and platform service components.	H
Business analysis	Understanding the impact of technical debt on the articulation of requirements and solutions.	M
Continual improvement	Identifying, prioritizing, and managing efforts to reduce technical debt.	M
Information security management	The design, implementation, and improvement of information security controls are influenced by existing technical debt. Information security controls can also result in the creation of technical debt, which needs to be acknowledged and communicated to all relevant stakeholders.	M
Project management	Planning and executing projects is influenced by existing technical debt. Projects can also result in the creation of technical debt, which needs to be acknowledged and communicated to all relevant stakeholders.	M
Risk management	Recognizing the impact of technical debt on new or existing enterprise risks; risk mitigations may create technical debt that needs to be acknowledged and communicated to all relevant stakeholders.	M
Service desk	Communicating with external users who need assistance with incidents and requests requires knowledge of existing technical debt and efforts planned to resolve it.	M

4.3.2 Chaos engineering

Definition: Chaos engineering

The discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production.

In order to address uncertainty of distributed systems, chaos engineering relies on four basic steps[13]:

- Define the steady state; this will be the output of a normal behaviour.
- Hypothesize that this steady state will continue.
- Introduce variables that reflect real-world events.
- Try to disprove the hypothesis.

The principles for an ideal application of chaos engineering are:

- **Build and hypothesize around steady state behaviour** Focus on measurable system outputs, rather than attributes of the system.
- **Vary real-world events** Chaos variables reflect real-world events. Prioritization must be defined by considering the impact or estimated frequency of the events.
- **Run experiments in production** Chaos prefers to experiment directly on production traffic.
- **Automate experiments to run continuously** Build automation into the systems to drive orchestration and analysis.
- **Minimize the blast radius** The fallout from experiments should be minimized and contained.

Some of the benefits of chaos engineering are that it:

- prepares the team for random instance failures
- encourages redundancy
- makes systems stronger, thereby instilling confidence to move quickly in complex systems
- introduces realistic conditions into a controlled run, uncovering weaknesses before they cause problems
- addresses the most significant weaknesses proactively, before they affect customers in production.

Chaos monkey is one of the tools in the so-called Simian army,[14] the best-known toolset for chaos engineering. The Simian army is a collection of open-source cloud testing tools created by Netflix®. Chaos monkey tests responses to system failures by disabling certain components to see how the remaining systems will respond. Although chaos monkey may disrupt operations, it also contributes to their longer-term resilience.

Definition: Chaos monkey

A tool that tests the resilience of IT systems by intentionally disabling components in production to test how remaining systems respond to the outage.

Deployment of the chaos monkey results in the termination of a randomly selected system within an identified group of systems. This creates a situation close to a natural incident and tests the ability of the systems to sustain failures. Another member of the Simian army, conformity monkey, checks each service in order to fulfil architect-defined best practices.

Other members of the Simian army include:

- **Latency monkey** Induces artificial delays to simulate service degradation and check whether dependent services respond adequately.
- **Doctor monkey** Performs health checks to determine unhealthy instances and proactively shut them down if owners do not fix the root cause.
- **Security monkey** Finds and terminates instances of security violations or vulnerabilities.
- **Janitor monkey** Ensures that the cloud environment is free of clutter and waste.

Figure 4.18 shows the contribution of chaos engineering to the service value chain.

Table 4.15 outlines the practices for which chaos engineering is relevant.

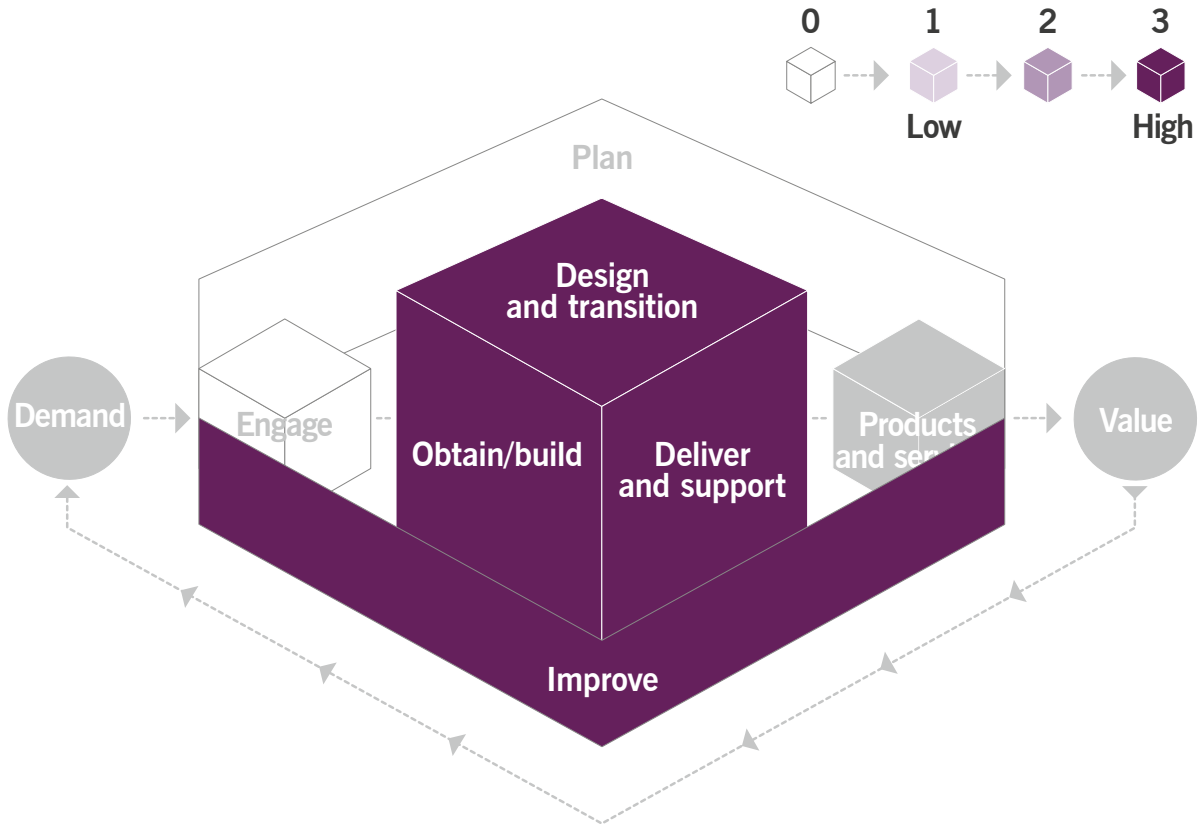

Figure 4.18 Heat map of the contribution of chaos engineering to the service value chain

Table 4.15 Practices for which chaos engineering is relevant

ITIL management practice	Activities/resources associated with chaos engineering	Impact
Continual improvement	Using chaos engineering as one of the most effective tools for improving service quality.	H
Infrastructure and platform management	Designing infrastructure and platforms for sufficient resilience and redundancy to deal with the unexpected outages caused by chaos engineering tools. Providing information for chaos engineering regarding service components and backup activities.	H
Service continuity management	Designing service continuity measures with sufficient resilience and redundancy to cope with the unexpected outages caused by chaos engineering tools. Continually monitoring continuity plans, measures, and mechanisms for resilience.	H
Service level management	Tests must be designed and run considering business continuity strategy, service level agreements, and clear criteria established for service degradation in case an artificial disruption exceeds acceptable levels.	H
Software development and management	Chaos engineering tools are themselves software applications that need to be developed (or configured) and managed. Software should be designed and architected with sufficient resilience and redundancy.	H
Architecture management	The building of resilient infrastructure, which is promoted by chaos engineering. Considering interactions between services and components in order to support demand.	M
Capacity and performance management	When running this type of test, performance information should be captured. As a result, improvements should be identified that will ensure services are designed for optimum performance, scalability, and capacity.	M
Incident management	Teams can practise responding to and recovering from outages by using chaos engineering tools. They must be prepared to manage incidents without impacting users. Redundancy and automation should be built into the processes.	M
Measurement and reporting	Chaos engineering tests involve experiments and hypotheses, and will help to collect and analyse data for planning and forecasting. The results support continuity business strategy.	M
Monitoring and event management	Monitoring and event management tools can be set up to flag outages orchestrated by chaos engineering tools, or to monitor the quality of service rather than the technical components.	M
Organizational change management	Chaos engineering will help to ensure engagement and cooperation in the live environment.	M
Problem management	Proactively detecting problems by introducing random failures and looking for potential flaws in services/components. The data gathered from chaos engineering tools can help to identify underlying problems that require investigation and remediation.	M
Service configuration management	CMDBs and code repositories should have high availability and accurate information (aligned with the recovery point objectives defined by service continuity management) to help the organization recover quickly from outages.	M
Service design	Chaos engineering testing principles can help architects design more resilient systems and improve user experience.	M
Service desk	The service desk team must be informed about the test, and prepared for managing incidents without impacting users.	M
Service validation and testing	Chaos engineering testing principles can help to evaluate service reliability. Architects should focus on service interruption.	M
Risk management	Certain types of organizational risks can be mitigated by using chaos engineering tools and methods to increase organizational resilience and robustness.	L

The ITIL story: Chaos engineering

Radhika: *We needed to test the app's resilience. For example, what would happen if the membership functionality ceased to work? Would customers still be able to book a car? Could the booking still be assigned to their account retrospectively?*

Solmaz: *We used the chaos monkey tool to understand how the app would work under duress. It allowed us to see where the system could break down, which meant that we could amend the code and software architecture to reduce or eliminate weak areas.*

4.3.3 Definition of done

Definition of done

A checklist of the agreed criteria for a proposed product or service.

The 'definition of done' is commonly used by application developers employing the Agile Scrum methodology. The Scrum community defines 'done' as 'having produced potentially releasable software increments'. In DevOps circles, this has been extended to 'released to production'. This much-needed extension is still inadequate when seen from an end-to-end perspective. Releasing the required functionality is not useful if users use it poorly, interpret data incorrectly, make sub-optimal decisions, or fail to act on good decisions. A narrowly focused IT function may not see preventing this as its responsibility, but as the organization's responsibility. Therefore, a more complete definition of done includes criteria for operations and use. The technique is broadly applicable and is often encountered in HVIT environments.

A definition of done describes the functional criteria that contribute to a product's or service's utility, and the non-functional criteria that contribute to its warranty. These non-functional criteria should be defined and agreed with whoever is responsible for operations. A definition of done that includes non-functional criteria therefore contributes to resilient operations and to value co-created through improved usability, and also to faster development because less rework is needed.

Non-functional criteria specify the quality needed by the people who will operate, maintain, and enhance the system, and by those who will ensure security and regulatory compliance. These criteria address the qualities needed of execution and evolution. Functional criteria describe 'what', whereas non-functional criteria describe 'how'. Quality attributes are specified in the definition of done, so that the development and support teams can consider them in early phases. For development, they are software design constraints.

In Agile software development, 'done' often means having software increments that are potentially deployable. DevOps extends this definition to three categories: deployed, released, and available for use. From a co-creational service perspective, a better way to define work as 'done' is when users have achieved the desired outcomes from their investments. Regardless of the approach chosen, the definition of done should be holistic and focused on value.

The following aspects should be considered when creating a definition of done:

- **Environments should be prepared for use** The continuous integration framework should be verified and working.
- **Deliverables for support should be complete** All criteria, user stories, and tests should have been accepted.
- **Metrics should be available** It is important to measure every release in order to verify that it fulfils user stories and criteria (better known as 'story points' in the Agile world).
- **Code must be understandable, maintainable, and ready to support future changes**.

It is also useful to consider the definition of ready, which describes when a backlog item is ready to be worked on. The acronym 'INVEST' represents useful criteria. Items should be:

- **Independent** Self-contained, and not dependent on other backlog items.
- **Negotiable** Discussion and fine-tuning are possible.

- **Valuable** There should be clarity around how stakeholders would benefit.
- **Estimatable** Scoped enough for an acceptable approximation.
- **Small** Small enough to estimate and to plan into a timebox.
- **Testable** With clear acceptance criteria.

Resilience tests verify that the system fulfils the agreed non-functional criteria in different areas, such as availability, capacity, efficiency, maintainability, performance, privacy, reliability, recoverability, and security.

Figure 4.19 shows the contribution of definition of done to the service value chain.

Table 4.16 outlines the practices for which the definition of done is relevant.

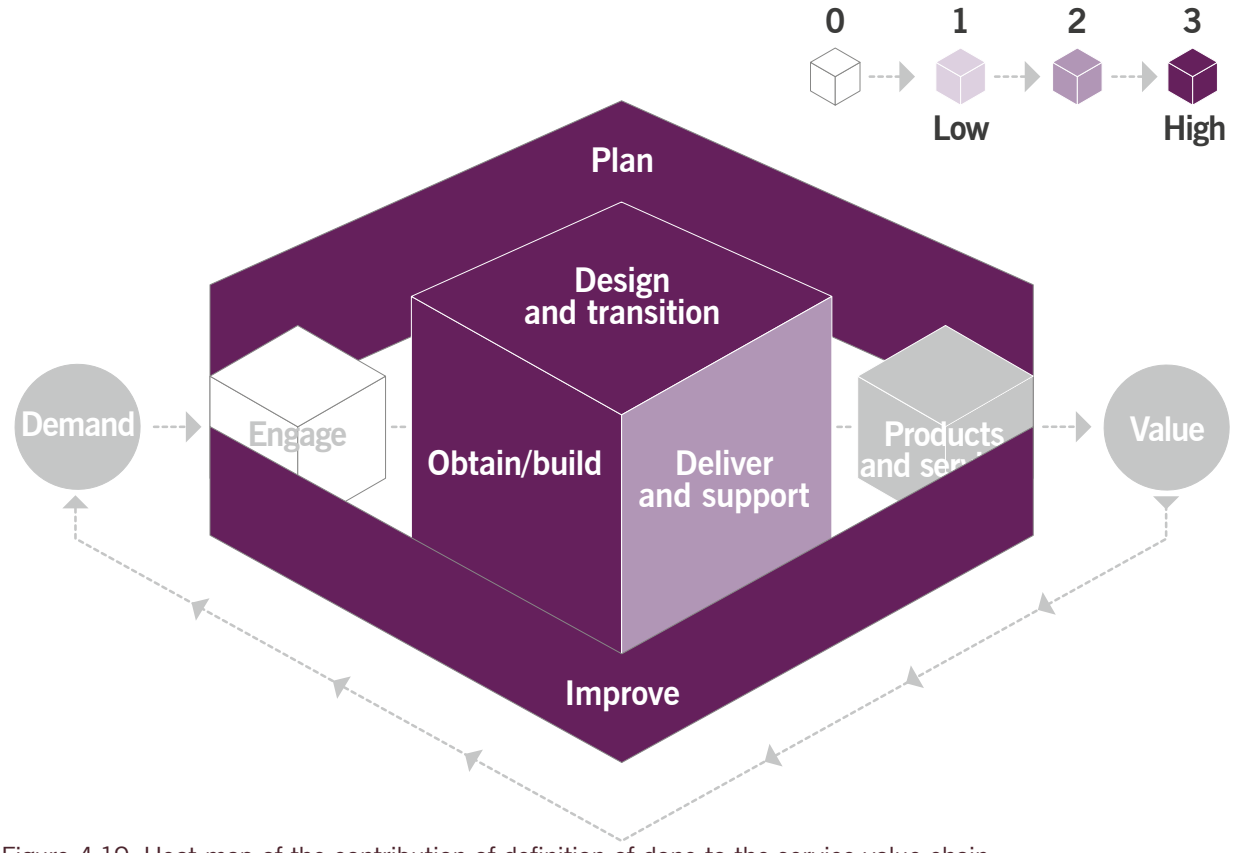

Figure 4.19 Heat map of the contribution of definition of done to the service value chain

Table 4.16 Practices for which the definition of done is relevant

ITIL management practice	Activities/resources associated with definition of done	Impact
Availability management	Detailed warranty requirements for the new or changed service should be negotiated and agreed with stakeholders.	H
Capacity and performance management	A definition of done checklist must consider capacity requirements, demand forecasting, and performance for managing business and customer expectations.	H
Change enablement	Change enablement activities can be structured around a definition of done; for example, to create a boundary with release management or deployment management.	H
Continual improvement	The definition of done can be used to scope and structure continual improvement activities, and to check whether outcomes have been achieved.	H
Deployment management	Deployment management activities can be structured around a definition of done; for example, to create a boundary with release management. When moving releases to live environments, teams should verify that deliverables for support are complete: all requirements, user stories, and tests should be accepted.	H

ITIL management practice	Activities/resources associated with definition of done	Impact
Incident management	Incident management activities can be structured around a definition of done; for example, to create a boundary with problem management.	H
Information security management	Security tests such as vulnerability, penetration, or policy compliance should be considered in the definition of done for resilient products and services.	H
Project management	Project tasks or outputs can define success or completion criteria using a definition of done approach.	H
Release management	Release management activities can be structured around a definition of done; for example, to create a boundary with change enablement or deployment management. The releases must be designed to match with business, customer, and user expectations. It is important to measure every release to verify that it fulfils user stories and requirements.	H
Service level management	A definition of done can articulate service actions for providers and consumers and can be used as the basis for monitoring actual performance against expected performance.	H
Service request management	Service request management activities, such as logging or fulfilling requests, can be structured using a definition of done approach.	H
Service validation and testing	Testing activities can be structured around a definition of done to ensure that multiple types of tests are conducted.	H
Software development and management	Software can be developed (or configured) to meet a definition of done before it is deployed into live environments, ensuring that code is understandable, maintainable, and ready to support future changes.	H
Business analysis	Functional and non-functional requirements for warranty and utility must be captured in order to fulfil customers' needs and expectations.	M
Service design	The definition of done should be customer-centric, in order to make design methods easier to adopt and ensure that services will be maintainable and cost-effective.	M
Service catalogue management	When the new functionality, product, or service is released, the service catalogue must be updated.	L
Service desk	Quality attributes are specified in the definition of done so that the development and support teams can consider them in early phases.	L

The ITIL story: Definition of done

Su: *The delivery team for the app includes people from many departments within Axle Car Hire. The traditional definition of done, when the developer hands over the working code, is not the most efficient or accurate. We want to guarantee the app's resilience, warranty, maintainability, utility, and usability. For us, 'done' means:*

- *the production and test environments are ready for use*
- *the continuous integration framework is prepared*
- *the user stories and tests have been acknowledged*
- *the measurements and metrics have been accepted and can be tested*
- *the software is readable, usable, and adaptable.*

4.3.4 Version control

Definition: Version control

The administrative management of sources and artefacts of information systems, products, and services.

Version control is of particular relevance to HVIT, because a strong correlation has been established between good version control and high IT performance: it provides for a quicker lead time for change, more frequent deployment, and a quicker mean time to restore service.

Version control tracks which versions of sources and artefacts exist in which environments. The common practice of storing software source code in a version control system can be expanded to include almost every other significant component of an IT system, product, or service, such as:

- scripts to create and configure infrastructure for every environment, including development, testing, and production
- acceptance criteria, test cases, and tests themselves
- external and internal libraries, modules, and components
- information about interdependencies
- scripts and configuration files for deployment pipelines
- scripts for operational tasks, such as recurring operations
- documentation, including architectural decisions
- artefacts that are needed to build and run the system
- contracts, agreements, and so on.

Version control therefore applies not only to service components (developed/engineered and managed in the obtain/build value chain activity), but also at the level of services. Proper version control allows the gathering of valuable information about the current and previous states of every component needed for a product or service. This information includes the initial state, the changes, the time and date of change, the person who made the change, and any additional clarifying and supporting information. The benefits of version control include:

- Version control supports the infrastructure as code technique (see section 4.2.1).
- Applying version control to sources and artefacts correlates to a quicker lead time for change, more frequent deployment, and a quicker mean time to restore service.
- Automated testing that is supported by version control correlates to a quicker lead time for change.
- Version control is a prerequisite to continuous delivery, which correlates to more frequent deployments.

Version control therefore contributes to resilient operations and fast development. It can also be considered an enabler for evolutionary architecture: a practice in which architectural decisions are not restrictive, but rather enable the continual evolution and improvement of products and services without redesigning and redeveloping previous solutions.

Figure 4.20 shows the contribution of version control to the service value chain.

Table 4.17 outlines the practices for which version control is relevant.

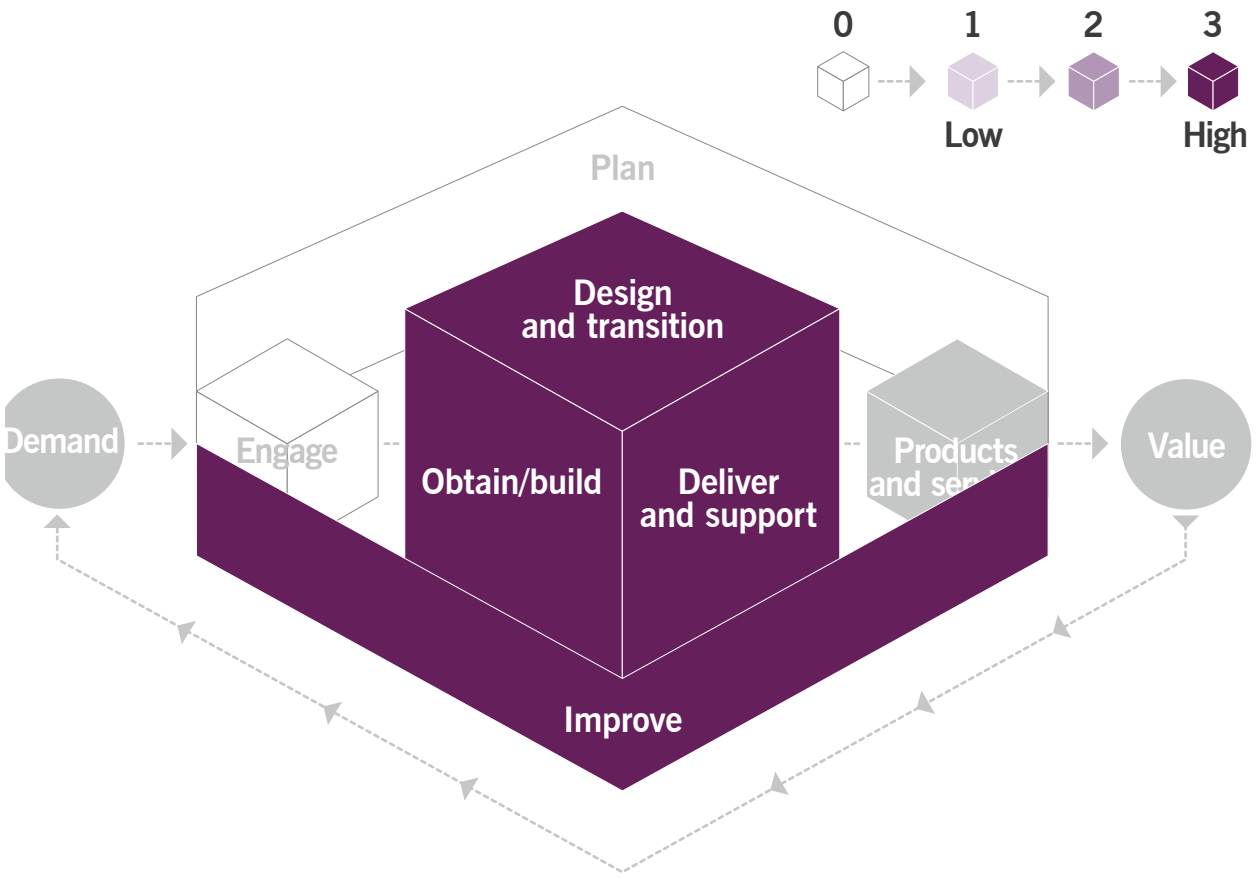

Figure 4.20 Heat map of the contribution of version control to the service value chain

Table 4.17 Practices for which version control is relevant

ITIL management practice	Activities/resources associated with version control	Impact
Deployment management	Using version-controlled repositories to deploy new or changed service components or return to a previous version.	H
Information security management	Addressing or closing information security risks by flagging vulnerable versions of service components.	H
Infrastructure and platform management	Infrastructure components, configuration settings, and virtual and physical infrastructure components can be formally stored and managed using a version-controlled repository.	H
Service configuration management	CMDBs can be federated, leveraging version-controlled code repositories, infrastructure-as-code configuration files, and even a store of physical devices and other hardware. Check-ins should occur multiple times each day, and environment specifications should be managed and versioned.	H
Software development and management	Code, and even configuration settings for other software components, can be formally managed using a version-controlled repository to house the outputs of software development and management work.	H
Continual improvement	Creating a baseline of the current environment, and updating the baseline once improvements have been made.	M
Incident management	Using a version-controlled repository of software or hardware components to resolve an incident.	M
Knowledge management	Updating knowledge repositories and communicating information when versions of service components change.	M
Service continuity management	Understanding the impact of new versions of service components; and, if viable, propagating them into service continuity and disaster recovery plans.	M
Service request management	Using a version-controlled repository of software or hardware components to quickly fulfil requests.	M

The ITIL story: Version control

Marco: *We practise continuous integration and continuous delivery. We utilize version control to systematically keep a record of each iteration of the app we release. If a release is unstable, we can swiftly restore the service by returning it to the previous stable release.*

4.3.5 AIOps

Definition: AIOps

The application of machine learning and big data to IT operations to receive continuous insights which provide continuous fixes and improvements via automation. Also referred to as 'artificial intelligence for IT operations' or 'algorithmic IT operations'.

AIOps aims to bring AI to IT operations, addressing the challenges posed by modern trends in the ongoing evolution of infrastructure, such as the growth of software-defined systems. The implications of these new technologies, such as the increase in the rate at which infrastructure is reconfigured and reshaped, necessitates more automated and dynamic management technologies, which may have a significant impact on an organization's digital services.

AIOps platforms are used to enhance and partially replace many primary IT operations functions, such as availability and performance monitoring, failure recognition, predictive analysis, and event correlation and analysis.

AIOps harnesses data platforms and machine learning, collecting observational data (for example, events, log files, operating metrics) and engagement data (for example, customer request and service desk tickets), and drawing insights by applying cognitive or algorithmic processing to this data.

These insights may be used to drive some or all of a range of common outputs, such as:

- **Issue detection and prediction** Helping the service organization to respond more quickly to incidents.
- **Proactive system maintenance and tuning** Reducing human effort and potential errors.
- **Threshold analysis** Enabling a more accurate picture of a system's normal range of operation.

The application of AIOps depends heavily on the availability of data to be analysed and whether the data lends itself for analysis, which it may not if it has an extremely complex nature and a weak correlation between symptoms, causes, and effects.

Some organizations have also started to use AIOps beyond IT operations, to give business managers real-time insights into the impact of IT on business. This keeps them informed and enables them to make decisions based on real-time, relevant data.

Figure 4.21 shows the contribution of AIOps to the service value chain.

Table 4.18 outlines the practices for which AIOps is relevant.

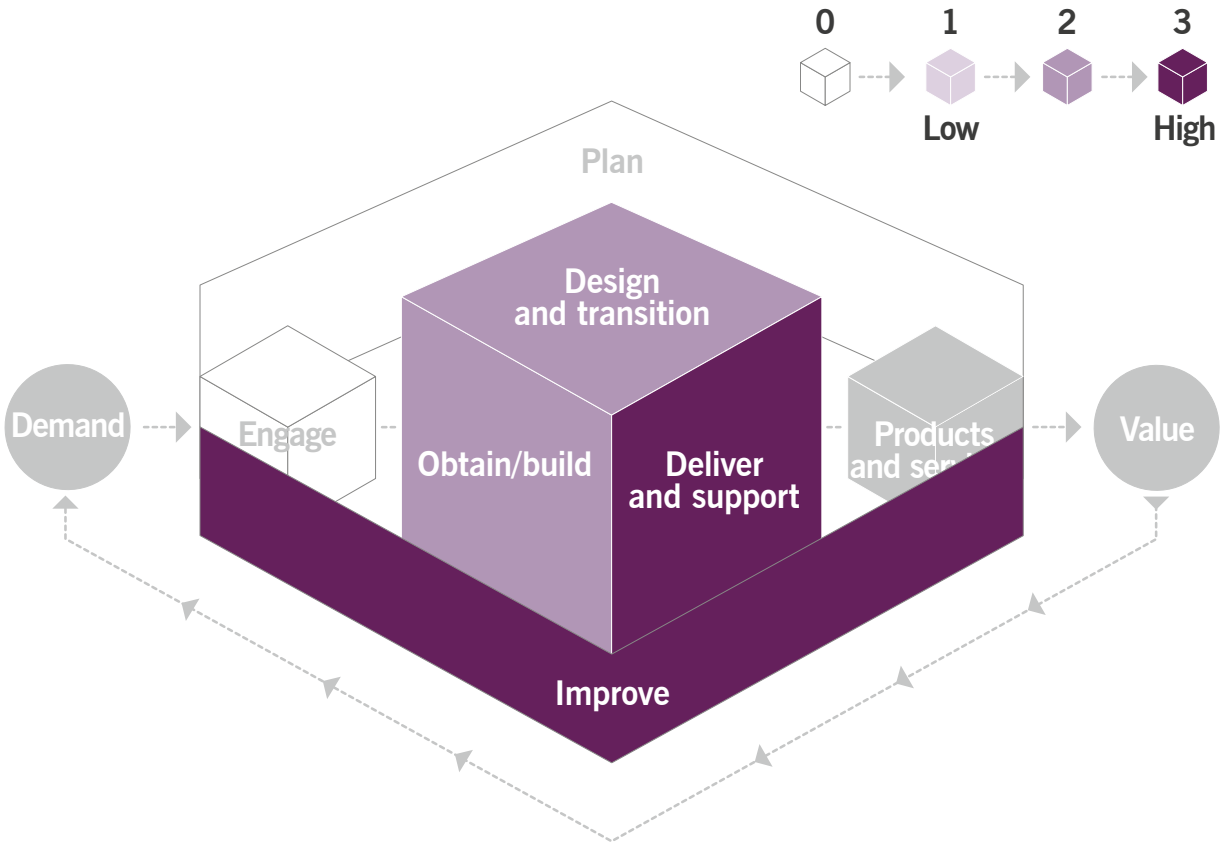

Figure 4.21 Heat map of the contribution of AIOps to the service value chain

Table 4.18 Practices for which AIOps is relevant

ITIL management practice	Activities/resources associated with AIOps	Impact
Capacity and performance management	AIOps provides capabilities for identifying patterns and anomalies, determining the capacity and utilization of assets, and planning the capacity of future products or services.	H
Incident management	Incident management data can benefit from the highly automated capabilities provided by AIOps tools that augment manual work. Resolving correlating incidents with contextual pre-analysed data merged from different systems.	H
Infrastructure and platform management	AIOps tools can automate much of the day-to-day management of infrastructure and platform resources.	H
Monitoring and event management	AIOps tools can help to correlate vast data sets from across multiple monitoring tools. They create a better understanding of the IT environment. AIOps enables value co-creation through an integrated set of business and operational metrics, thereby reducing the frequency of operational events or incidents because they are predicted and prevented. AIOps helps to optimize IT and reduce IT costs by replacing silo-focused IT monitoring tools, and by monitoring the health and performance of applications of all tiers in value streams.	H
Change enablement	AIOPs supports the visualization of dependency details at every device level.	M
IT asset management	AIOps may collect dynamic inventory information with logical and physical attributes.	M
Measurement and reporting	AIOps provides data for metrics to evaluate performance and regulatory compliance. It also helps to automate the reporting task.	M
Problem management	Information from AIOps tools can aid in identifying and investigating problems and errors, and in automating and monitoring the application of workarounds. They can also help with proactive problem detection based on pre-processed and merged data.	M
Service configuration management	AIOps data can be used to detect changes to configuration items, helping to identify unauthorized changes.	M

Table continues

Table 4.18 *continued*

ITIL management practice	Activities/resources associated with AIOps	Impact
Service desk	Information from AIOps tools can support engagement with external stakeholders. AIOps helps organizations to proactively plan, identifying issues and their business impacts before they occur. AIOps also enables the informed triage of user queries based on merged data and identified trends.	M
Workforce and talent management	Organizations that implement AIOps breakdown silos across their IT teams enable less-experienced staff to be more productive, developing skills and efficiencies.	M
Knowledge management	The combination of knowledge of IT processes, operations, performance results, and data processing algorithms supports critical business functions.	L

The ITIL story: AIOps

Radhika: *Thousands of customers use the app and hire our vehicles. These transactions generate a huge amount of data, which is a rich source of information about customer demand.*

Su: *We have created scripts to analyse the data, find patterns of use, and optimize the infrastructure of the service. For example, if the data suggests that users of electric cars are reaching the end of the battery charge, the scripts automatically highlight a tip explaining how to recharge the battery, with a map of the nearest recharge facility.*

4.3.6 ChatOps

ChatOps is a model in which people, tools, process, and automation are connected in a transparent flow. This model helps to control pipelines and collaboration. It is a close-fitting integration of instant communications with operational execution: an emerging movement that promotes the integration of several teams, tools, and DevOps platforms. Development is driven by bringing tools and platforms into conversations. When bots are team members, it is possible to send requests to them and get instant responses.

ChatOps enables collaborative communication among humans and tools, reducing incident response time by eliminating repetitive requests for information and automating some regular IT operations actions.

The elements of ChatOps include:

- **Chat platform** The service that connects stakeholders, teams, and the systems they work on.
- **Bots** The core of the ChatOps model. Bots exist and work between the collaboration tool and DevOps tools. They receive requests from team members, then retrieve information from the integrated systems by executing scripts.
- **Integration and automation services** The third-party elements in ChatOps; for example, for issue tracking: version control systems, infrastructure as code, continuous integration servers, or monitoring tools.

This model is becoming more popular. Organizations are connecting their chat platforms to their build systems in order to receive notifications and execute and query processes on their continuous integration servers. The same model can be applied to quality assurance teams. The ChatOps workflow considers:

- work needed
- work happening
- work done.

This model promotes feedback, improves communication and cross-training, and enhances team collaboration. While 'chatting', people collaborate and innovate, driving progress. ChatOps humanizes work by capturing knowledge and identifying requirements to deliver services as planned or expected.

This type of tool has highlighted the need for instant collaboration between tools, operation teams, and messaging tools. ChatOps is the evolution of traditional chats, as it enables systems to join the conversation. For example, DevOps or IT and service management tools can notify support groups of an incident or event.[15]

Figure 4.22 shows the contribution of ChatOps to the service value chain.

Table 4.19 outlines the practices for which ChatOps is relevant.

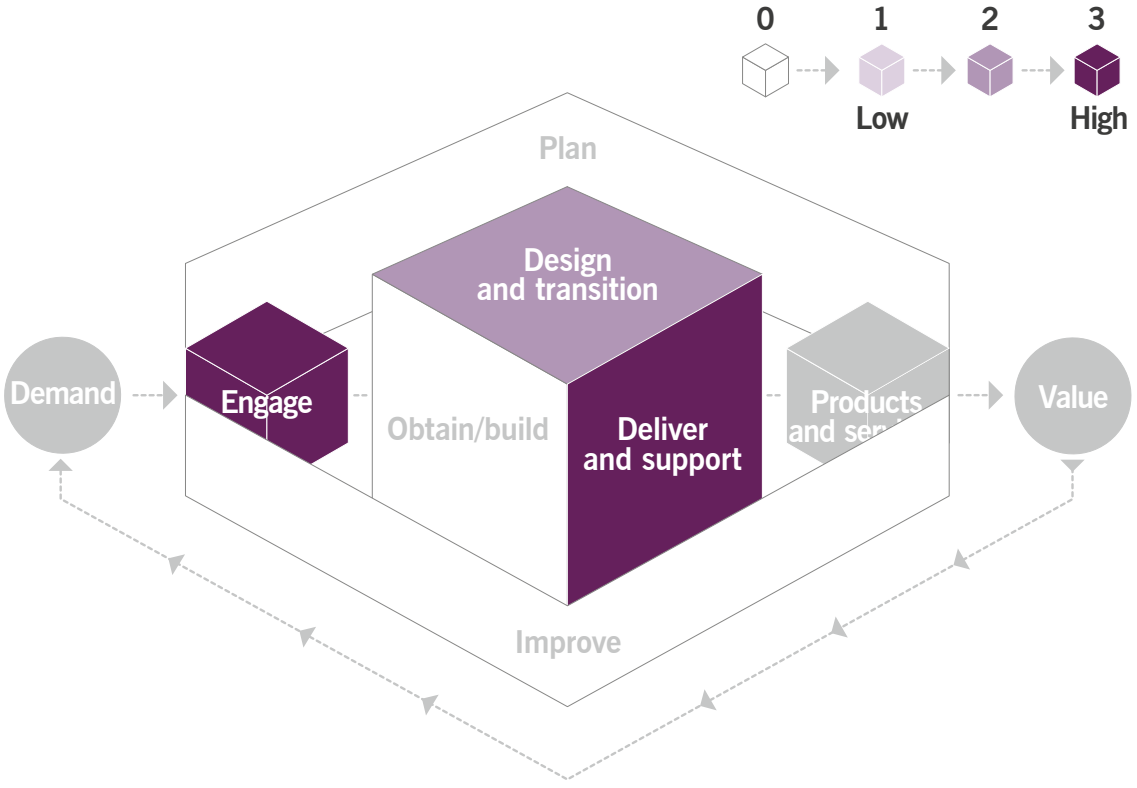

Figure 4.22 Heat map of the contribution of ChatOps to the service value chain

Table 4.19 Practices for which ChatOps is relevant

ITIL management practice	Activities/resources associated with ChatOps	Impact
Service desk	Communicating and coordinating with users to better manage incidents and requests.	H
Change enablement	Communicating and coordinating between all teams involved in managing changes to services and service components. Some ChatOps tools can integrate with other IT and service management tools. ChatOps provides a channel for communicating with users and team members about new or changed services, thereby humanizing the way of working.	M
Continual improvement	Meeting goals of continual improvement initiatives to improve communication and coordination between teams.	M
Deployment management	Communicating and coordinating between all teams involved in deploying new or changed service components. Some ChatOps tools can integrate with deployment tools.	M
Incident management	Communicating and coordinating between external stakeholders and various teams involved with incident management activities. Some ChatOps tools can integrate with other IT and service management tools. ChatOps helps IT teams in support activities, such as registering, analysing, and diagnosing, thereby reducing response times and eliminating repetitive tasks.	M

Table continues

Table 4.19 *continued*

ITIL management practice	Activities/resources associated with ChatOps	Impact
Knowledge management	Searching for unstructured knowledge in chat logs. Capturing knowledge and identifying requirements to deliver services as planned or expected. Collecting feedback to support continual improvement.	M
Problem management	Running root-cause analyses and post-mortems.	M
Release management	Communicating and coordinating between all teams involved in managing changes to services.	M
Risk management	Storing data and information in a searchable format.	L

4.3.7 Site reliability engineering

Definition: Site reliability engineering

A discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems with the goal of creating ultra-scalable and highly reliable software systems.

Because highly digitally enabled organizations demand highly resilient operations, site reliability engineering (SRE) is particularly relevant to HVIT.

SRE applies a software development mindset to IT operations, and helps to align development and operations. SRE teams split their time between executing IT operations, coaching IT operations teams, and developing software that increases the resilience and performance of IT systems. They tend to spend less than half of their time on toil (otherwise, this indicates a problematic system).

Toil is defined[16] as work that is:

- **Manual** It requires hands-on time from humans.
- **Repetitive** It is being done over and over again.
- **Automatable** It could be achieved by a machine because it does not require specific human judgement.
- **Tactical** It is interrupt-driven and reactive, rather than strategy-driven and proactive.
- **Devoid of enduring value** It does not permanently improve the service.
- **Linearly scaling** It scales in proportion to the service size, traffic volume, or user count.

SRE is based on experience that suggests that systems are complex and will therefore fail. Effort is therefore balanced between preventing failures and reducing the impacts of unpreventable failures, such as by designing systems to degrade gradually rather than fail abruptly. Failure is regarded as an opportunity to learn, recognizing that people are the adaptable element of complex systems, and that their expertise is changing just as much as the other parts of the system.

Learning from failures is not focused on the identification of a root cause, as in complex systems this is not effective or even possible (see section 3.2.3.1). Rather, failures are used to improve the team's collective knowledge. They help people to continually calibrate their mental models, so that they will recognize hazards more easily and act to keep the system within the boundaries of acceptable performance. Practitioners assess when to apply a standard fix and when improvisation is required. They are never certain of the consequences of their actions, and they learn from feedback about how the system's performance changes according to their actions. Because the team should be taking risks, it is crucial that it is not part of a blame culture. Although

algorithms have their place, in machine learning, for example, heuristics are more effective in dealing with complex systems, so human judgement is crucial. This requires intelligence, experience, and the motivation to act; and a workplace that encourages the desired behaviour.

In availability management, it is important to balance outage prevention and incident resolution. The corresponding key availability metrics are MTBF and MTRS. The more complex the system, the more failures are inevitable, meaning that the focus should shift from preventing failure to restoring service rapidly. The correct balance will vary from service to service, depending on the service's warranty requirements and the underlying system's characteristics. In HVIT environments, systems will usually be more complex, and it will therefore be more effective to focus on reducing MTRS than increasing MTBF.

It can be challenging to balance investing in new features and in service reliability. An error budget is a powerful SRE tool that helps in this area. Changes tend to cause system incidents. Development work for features and development work for stability need to be balanced. An error budget is a control mechanism that allocates appropriate capacity to development work for stability, ensuring the right balance. When a service is close to its error budget, the product team should focus on improvements rather than new features.

An error budget is expressed as 100 per cent minus the service level objective (SLO) of the service. A 99.9 per cent SLO service has a 0.1 per cent error budget. This amount should be spent on improving stability. Error budgets allow teams to regulate themselves within policies, with consequences if the error budgets are exceeded. It is important that SLOs are expressed in terms that affect the service consumer experience, rather than being internal system KPIs.

The improvements that SRE brings to availability, latency, and performance all contribute to resilient operations.

Figure 4.23 shows the contribution of SRE to the service value chain.

Table 4.20 outlines the practices for which SRE is relevant.

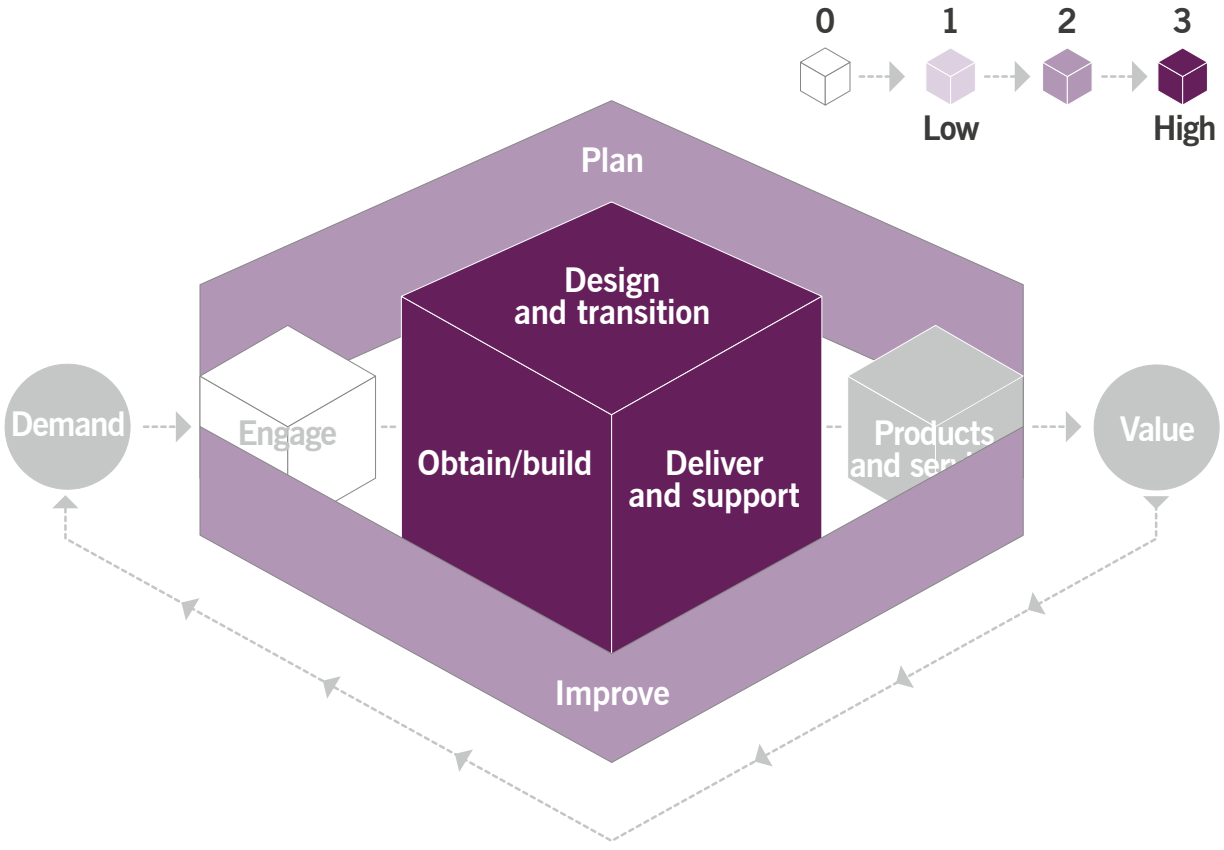

Figure 4.23 Heat map of the contribution of SRE to the service value chain

Table 4.20 Practices for which SRE is relevant

ITIL management practice	Activities/resources associated with SRE	Impact
Availability management	Using SRE techniques and tooling to improve visibility of a system in order to judge the service health and diagnose issues. Tracking the 'technical' MTBF and (more crucially) MTRS metrics, such as user outage minutes, number of lost transactions, lost business value, and user satisfaction. Using error budgets to balance service reliability and innovation.	H
Capacity and performance management	Using SRE techniques and tooling to improve visibility of a system in order to judge the service health and diagnose issues. Monitoring systems and defined SLOs must be accounted for and measured. Improving monitoring in order to better understand the system when things go wrong.	H
Change enablement	Using SRE techniques and tooling to enable changes to service components and the roll-back of failed changes.	H
Incident management	Using SRE techniques and tooling to manage incidents in the infrastructure or platform layers.	H
Infrastructure and platform management	Using SRE techniques and tooling to help architect and design infrastructure and platform capabilities to meet the organization's needs.	H
Monitoring and event management	Using SRE techniques and tooling to improve visibility of a system in order to judge the service health and diagnose issues.	H
Problem management	Data from SRE tools can help to identify problems, ensuring workarounds are applied quickly through the use of automation. Automating IT processes improves resilience and reduces toil. Use of post-mortems.	H
Service design	SRE collaboration during the design phase can prevent a variety of problems or incidents from occurring later in production. Although design decisions can be reversed or rectified later in the development lifecycle, such changes come at a high cost in terms of effort and complexity.	H
Software development and management	Providing requirements to SRE teams and acting on feedback.	H
Deployment management	The deployment process should align with the risk process described in service design.	M
Organizational change management	An SRE team has a core responsibility to prepare its teams for rapid innovation.	M
Release management	With SRE, techniques used for releasing software are applied to digitized infrastructure.	M
Service configuration management	With SRE, automated discovery and version control can be applied to infrastructure components.	M
Service validation and testing	For release engineering in SRE, it is recommended that continuous build test targets correspond to the same test targets that gate the project release.	M

The ITIL story: Site reliability engineering

Su: *The more features we add to the app, the more complex it becomes and the more likely that the code within it will fail. Failure is an inevitable feature of any software platform. The way the app fails can teach us how to recalibrate it to be more resilient.*

Radhika: *Site reliability engineering balances our need to reduce the occurrence of failure with the need to repair the service after it fails. The more we can automate work and reduce repetitive manual actions, the stronger the code is.*

4.4 Techniques for co-created value

The co-created value objective involves co-creating value from digital products through the close collaboration of the service provider and the service consumer.

Co-created value is about the service consumer using the service provider's products and services effectively and benefiting from their utility and warranty. A return on digital investments is only realized when decision-making, whether done by people, automation, or AI, is improved by information derived from automated information systems. Users, therefore, have to understand the digital products and information, and their uses in their context. They should understand the functionality well enough to use it appropriately, and be able to interpret the information correctly in order to improve decision-making. Finally, people or things have to act on these decisions; only then is value realized.

Example

Someone uses a ride-hailing app to get transport to the airport. They misinterpret the indicated arrival time as a guarantee instead of an estimate. They make a misinformed decision to wait. They then have a nervous ride to the airport because they may miss their flight. Correctly interpreted, the information would have led them to decide to hail a regular taxi, and would therefore have had value.

Many users do not use the functionality of information systems effectively. They also misinterpret the meaning of the data that a system provides, subsequently make sub-optimal decisions, and therefore do not achieve the required return on their IT investment. Not only is the potential value not realized, but productivity often declines when issues occur. Some of the resulting loss is due to issues related to resilient operations, and some is lost due to misuse by the service consumer. This means that proactive, functional user support should be prioritized. In a business environment, this form of support is more suited to a co-located role, such as a proactive super user who acts as a 'value realization coach', than to a distant service desk.

'Co-created value' refers to the value for the service consumer, service provider, and other stakeholders. For the service consumer, the value is the outcome that the service output facilitates. For the service provider, depending on the nature of the investment, value can be expressed in different terms, such as revenue from digital products and services, website traffic, and reduced costs and risks.

In HVIT environments, where service consumers use digital products and services regularly, expectations are higher. Consumers demand a more intuitive and responsive user experience and customer experience. The better the service providers understand how service consumers use the IT service or digital product, the better equipped they are to support them. Equally, the more that service consumers understand how service providers provide IT services or digital products, the better equipped they are to interact effectively with them. These concepts illustrate the symbiotic, co-creational nature of services.

Although the digital experience will, by definition, be algorithmic, seasoned digital service consumers expect that it should also be sophisticated and as finely tuned to their circumstances as possible. If this is too difficult for the digital service to accomplish, service consumers expect a sophisticated physical/analogue/human experience with the service provider's staff. Where the digital experience relies on advanced algorithms, the human experience relies on advanced heuristics that are needed to deal with unpredictable situations. Such service interactions are social interactions, and a service can be ruined by small details in human interaction. Experienced service providers and consumers acknowledge contextual constraints and the human nature of the

other party. This results in the mutual experience that the possible has been accomplished, with mutual respect for each other's position.

Co-creation is not only about the service interaction, where value is actually realized. It is also about the consumer's involvement in service design and further development. The collaboration between a Scrum product owner and a development team is a good example of the close collaboration between IT practitioners, business people, and, in some cases, customers and consumers. Such self-contained product- or service-oriented teams are very effective. In many cases, this construct can only be realized when the information system is designed with this way of working in mind, enabling various small teams to work on relatively isolated parts of the bigger system without much need for interaction. This requires a loosely coupled information system architecture (see section 4.2.2).

An important technique to support value co-creation is service experience.

The ITIL story: Techniques for co-created value

Henri: *Our goal is to co-create value for all stakeholders, so we need to ensure that the interface we offer for booking cars is consistent and intuitive, regardless of any changes that are happening under the surface. The app should respond seamlessly to customer requirements, to ensure that users receive an optimized service that provides optimal value.*

4.4.1 Service experience

'Service experience' refers to the fact that service consumers value a service that is based on a combination of the 'technical' output of the service and how it is perceived from a human perspective. This means that service providers should be increasingly aware of consumer requirements and the resources they have at their disposal to co-create value. Services are not passively received: value co-creation requires effort from the consumer. The service provider and consumer have to dynamically respond to each other's behaviour and accommodate exceptions as much as possible.

When, in some digitally enabled organizations, business and IT converge into a single organizational entity, there are no longer business and IT entities that need to be aligned. There is, therefore, also no longer a business–IT relationship to manage. 'Business people' and 'IT people' report to the same management, have the same targets, and are often physically co-located. When an Agile or Scrum way of working has been adopted, with a relatively independent team dedicated to a single product, business people and IT people are in the same team, and the product owner represents business interests. The product owner often manages the relationships with external customers and other stakeholders. These other stakeholders include other product owners with whom synergies are explored; for example, knowledge and resource sharing.

Data analytics and machine learning can contribute significantly to relationship management. Information security is also important, as are ethics. Customer experience management and customer journeys are other topics that deserve consideration.

For further details on service relationships and service experience, see *ITIL® 4: Drive Stakeholder Value*.

Figure 4.24 shows the contribution of service experience to the service value chain.

Table 4.21 outlines the practices for which service experience is relevant.

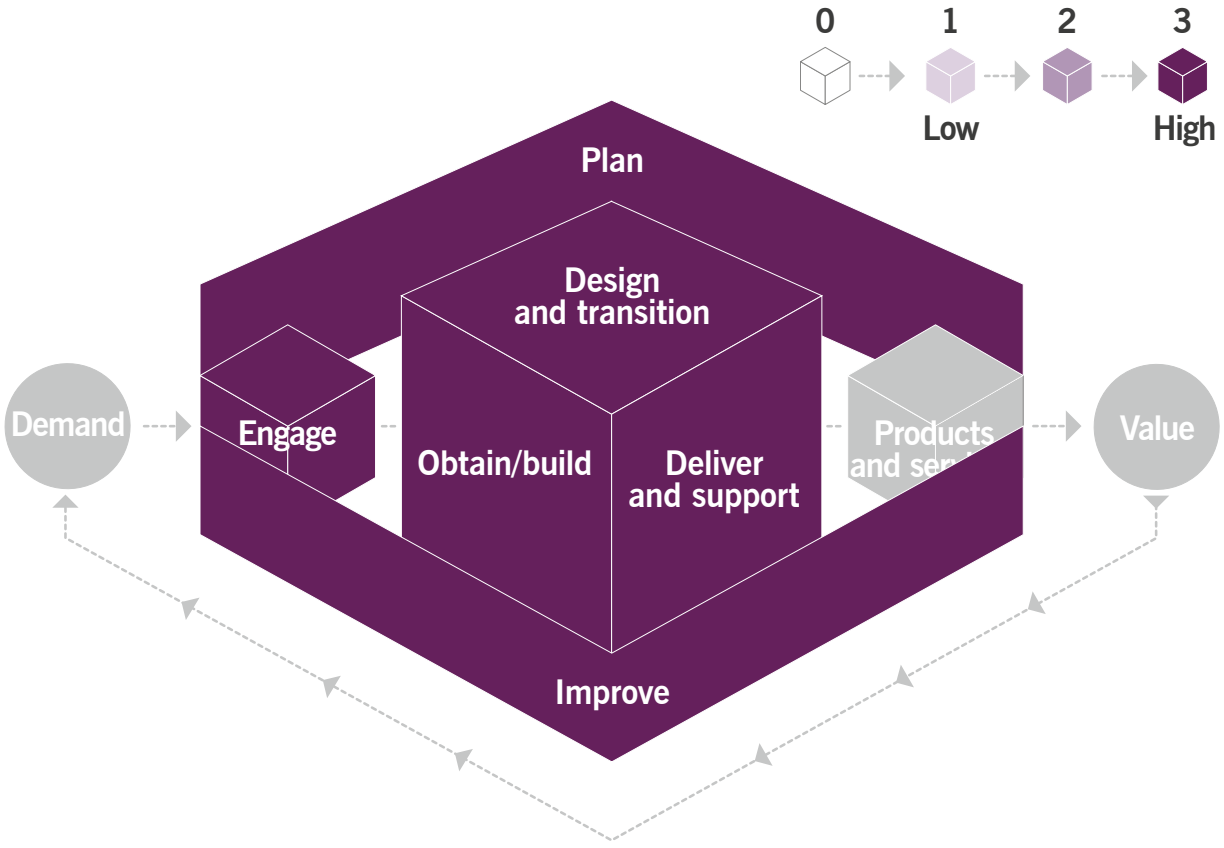

Figure 4.24 Heat map of the contribution of service experience to the service value chain

Table 4.21 Practices for which service experience is relevant

ITIL management practice	Activities/resources associated with service experience	Impact
Business analysis	Understanding user needs and translating them into customer experience or user experience requirements, in addition to traditional requirements regarding utility and warranty.	H
Service catalogue management	Describing services and offerings in terms of technical as well as experiential aspects.	H
Service design	Articulating customer experience and user experience needs beyond a basic experience.	H
Service desk	Being empathetic and having the emotional intelligence to understand users' experiential needs. Giving users a choice of communication channels. Service experience requires technology and information enablers, such as self-service tools, online portals, mobile applications, call centre tools, and chat. Using user satisfaction as a KPI. Assessing user experience, while choosing a tool for two-way communication with users. Gathering service experience data (rough estimates of users happy/not happy with the service).	H
Service level management	Promoting a good understanding of the psychographics of the service consumer and the (emotional) effect of service interactions on the consumer.	H
Software development and management	The desired service experience informs the design of the user interface.	H
Monitoring and event management	Developing and configuring tools and techniques to monitor service experience and associated events, in addition to technical monitoring and event management.	M
Relationship management	Being empathetic and emotionally intelligent to understand consumers' experiential needs.	M
Service validation and testing	Developing and maintaining tests of the service experience.	M
Supplier management	Engaging and managing suppliers based on both subjective and objective agreements.	M

The ITIL story: Service experience

Su: *At Axle Car Hire, there is no divide between the business and IT. The development team collaborates to deliver a service experience that is responsive to customer requirements. We use data from both the app and vehicles to guide the optimization and automation of the service. The app is customizable, so the user can optimize the service according to their needs.*

4.5 Techniques for assured conformance

The assured conformance objective involves ensuring that service provision and service consumption comply with corporate and regulatory directives with respect to governance, risk, and compliance. Beyond ensuring conformance, it is also important to assure the accountable people that conformance has been achieved.

Although external requirements may remain the same, there may be alternative and more appropriate ways for digitally enabled organizations to fulfil them.

High velocity is often associated with taking risks, and, from a commercial perspective, these risks may be necessary. Paradoxically, one of the biggest risks an organization can take is not taking enough risks. Nevertheless, risks must be justified, and there are internal rules and external regulations that organizations must comply with. Governing bodies must be assured that their directives have been followed. Assured conformance reassures those people who are accountable for, and affected by, governance, risk, and compliance issues, as they feel more confident knowing that the organization acts within these constraints.

Regarding governance, the practitioner does not govern but is governed. They operate within a governance framework, and must understand the applicable constraints and how to act within that framework. The practitioner's insight and judgement influence how they act. The more insight they have and the better their judgement skills, the more the practitioner will be able to judge when it might be appropriate to deviate from the rules when the associated benefits and risks are justifiable. This requires the practitioner to understand the thinking behind the constraints.

Assured conformance can be measured by (the lack of) security breaches, fines by regulators, bad publicity, actions required by internal and external auditors, and the cost of the measures to ensure conformance with governance, risk, and compliance issues.

Techniques that can be used to achieve assured conformance include:

- DevOps Audit Defense Toolkit
- DevSecOps
- peer review.

The ITIL story: Techniques for assured conformance

Henri: *Like all ethical businesses, Axle fully complies with laws and regulations. We utilize techniques that assure conformance because sometimes IT progresses so swiftly that compliance requirements can be overlooked or delayed. Our dedicated governance team is just one of the ways we keep a lookout for changes in compliance requirements.*

4.5.1 DevOps Audit Defense Toolkit

The DevOps Audit Defense Toolkit[17] is guidance that addresses the tension between IT and audit that is caused by new, more fluid patterns of work found in the DevOps community. It helps to demonstrate to auditors that the IT function understands the business risks and is properly mitigating them. The Toolkit suggests techniques to mitigate risk and to create a common perspective and shared understanding between the IT function and auditors. It therefore contributes to assured conformance. By reducing unnecessary bureaucracy, it also contributes to fast development.

The DevOps Audit Defense Toolkit is relevant to HVIT because some of HVIT's principles and techniques appear to contradict conventional compliance requirements. Usually, however, this is a case of finding other ways of achieving the desired results. Internal regulations are derived from external requirements, and, often, alternative internal regulations can be found. It is essential, however, to involve auditors in this process.

Figure 4.25 shows the contribution of DevOps Audit Defense Toolkit to the service value chain.

Table 4.22 outlines the practices for which the DevOps Audit Defense Toolkit is relevant.

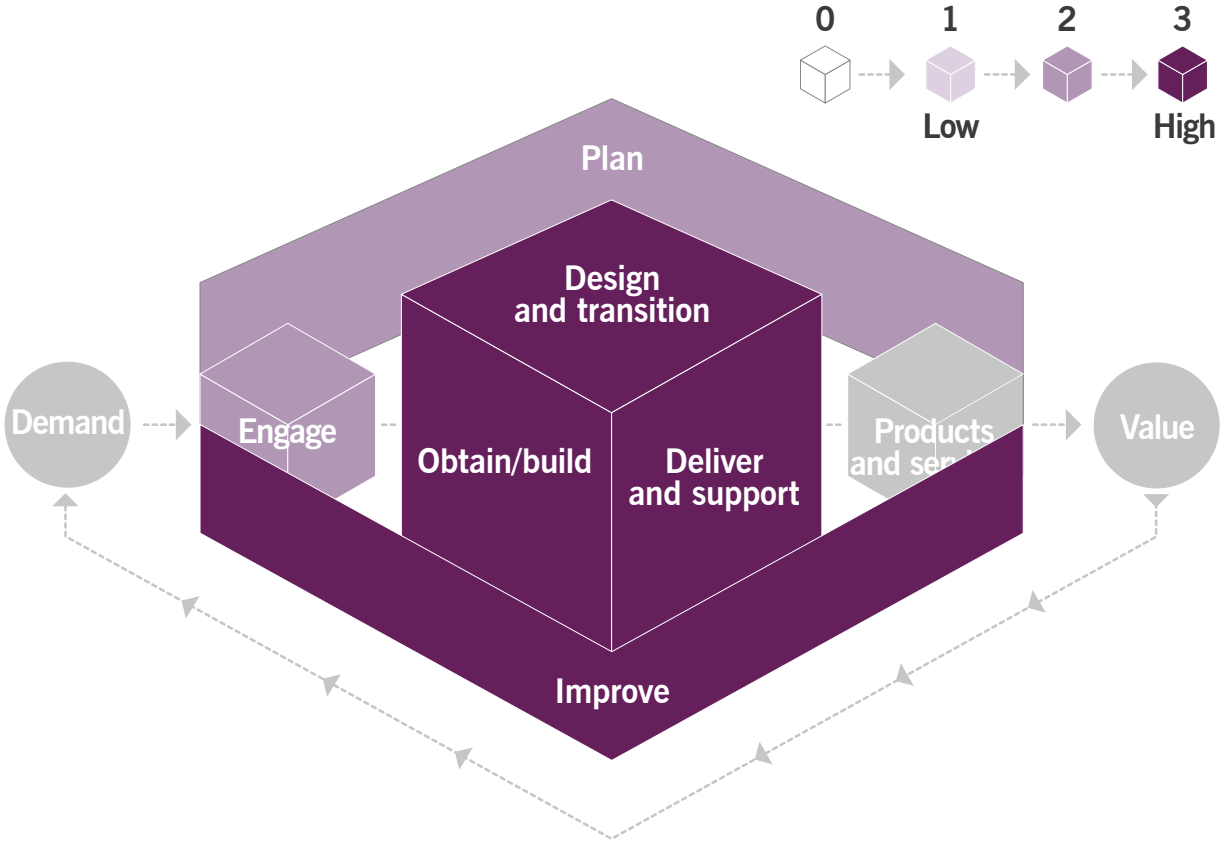

Figure 4.25 Heat map of the contribution of DevOps Audit Defense Toolkit to the service value chain

Table 4.22 Practices for which the DevOps Audit Defense Toolkit is relevant

ITIL management practice	Activities/resources associated with the DevOps Audit Defense Toolkit	Impact
Continual improvement	Auditing provides new information or opportunities for improvement that are formally registered, prioritized, and managed.	H
Information security management	Designing and implementing controls in the product lifecycle to provide extensive traceability and joint accountability.	H
Monitoring and event management	Operational data warehouses consolidating performance and event data provide a rich repository of information to audit implementation and performance of controls.	H
Service configuration management	Standardized configurations to support security and audit requirements.	H
Knowledge management	Giving staff and other key stakeholders access to relevant policy documentation and previous audit reports.	M
Risk management	Creating a balanced and practical approach between enterprise risk management, technical risk management, and new ways of working.	M
Workforce and talent management	Training staff on their obligations and duties to ensure compliance with all relevant policies and regulations.	M
Business analysis	Incorporating audit findings and suggested remediation into the product backlog.	L
Strategy management	Incorporating regular external or internal audits into a service's roadmap in order to provide independent governance of the service.	L

4.5.2 DevSecOps

Most organizations have a dedicated information security team, which performs risk assessments and defines policies, procedures, and controls. In high-velocity environments, information security is integrated as much as possible into the daily work of development and operations, and it shifts the reliance on control of process towards verifying preconditions, such as employees' expertise and integrity. The security officer's role shifts from 'policing' to enabling others to take necessary measures.

'DevSecOps' refers to this integration of security-related activities into the daily work of application development and IT operations. Security is built into the entire DevOps process and across the four pillars of culture, automation, metrics, and sharing (CAMS, or CALMS with the addition of Lean).

Definition: Integration of duties

Having a task that is prone to fraud or error performed by one person because other controls have been applied. This serves as an alternative to separation (or segregation) of duties.

Traditionally, duties are separated in order to reduce the risk of fraud and error; for example, the risk that untested and unauthorized code will be deployed into production. This can, however, lead to delay, and frustration at the perceived bureaucracy. The separation of duties is not an objective in itself: it is a method of reaching an objective. Other means are available to achieve the same objective, so duties can be integrated while maintaining the same level of assurance.

Information security is critically dependent on the behaviour of people throughout the organization. Staff who have been trained well, and who follow information security policies and other controls, can help to detect, prevent, and correct security incidents. Poorly trained or insufficiently motivated staff can be a major vulnerability.

Many processes and procedures are required to support information security management. These include:

- a security incident management process
- a risk management process
- a control review and audit process
- an identity and access management process
- event management
- procedures for penetration testing, vulnerability scanning, and so on
- procedures for managing security-related changes, such as firewall configuration changes.

This integrated way of approaching security contributes to assured conformance.

Case study

A large music-streaming service provider relies on being able to deliver quickly. It continually improves to maintain its leading position. Its way of working, its operating model, is based on speed and continual improvement. An impending change in its legal status introduces new compliance requirements, triggering a change to its operating model.

In particular, its financial systems are impacted by required controls regarding the segregation of duties and audit trails. Processes and the associated tools need to be changed. This is initially met with resistance from the autonomous teams, who were proud of the ways of working that they had developed.

To overcome this resistance, the teams are given a challenge and ownership of the solution. The new regulations are presented as a fact of life: a natural result of the growth of the enterprise. Because the teams are used to working with a great deal of autonomy, they are trusted to discover how to comply without compromising flow and agility. They are offered expert help from the internal audit team and the process tool team.

Each team develops its own process flow and process tool configuration, and interacts with key stakeholders. Although the diversity in approaches is probably less efficient than a one-size-fits-all solution for all teams, the benefits are evident. There is better flow because each team follows its own specific process. And, significantly, each team takes full responsibility for complying with the controls. This creates sustainable benefits.

The key learnings from this case are:

- **Play to the strengths of the teams** In this case, responsibility was welcomed. With less-autonomous teams, another approach might be needed.
- **Focus on external regulations, not on how they have been translated into internal policies and constraints** There are often alternative ways of achieving the same compliance. This requires flexibility from the internal audit team.

Figure 4.26 shows the contribution of DevSecOps to the service value chain.

Table 4.23 outlines the practices for which DevSecOps is relevant.

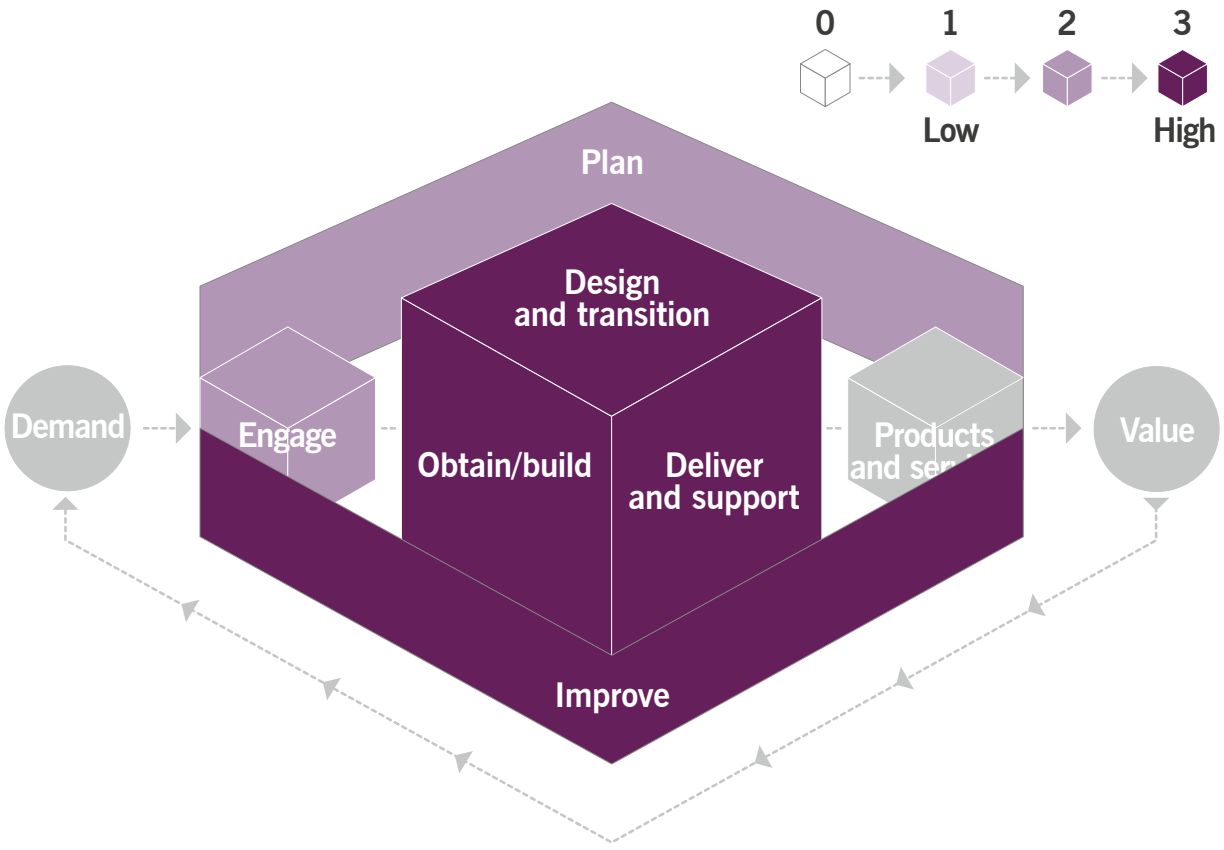

Figure 4.26 Heat map of the contribution of DevSecOps to the service value chain

Table 4.23 Practices for which DevSecOps is relevant

ITIL management practice	Activities/resources associated with DevSecOps	Impact
Continual improvement	Improvements to security controls and policies can be part of the learning and feedback incorporated by development and operations teams.	H
Information security management	Designing and implementing controls in a development lifecycle to provide extensive traceability and joint accountability. Integrating information security duties into the daily work of practitioners.	H
Monitoring and event management	Configuring monitoring tools to continually scan for threats and vulnerabilities so that they can be escalated to the appropriate teams.	H
Change enablement	Implementing a preventative control that automatically requires pre-authorization from security management before developers can make certain types of production data edits, including functions that they have entitlements to, based on certain defined criteria.	M
Deployment management	Security management provides guidance on key credential management, CD pipeline security checks, container security, automated penetration testing, and data and performance monitoring. Information security management and risk management should be an integral part of the daily work of practitioners.	M
Knowledge management	Giving staff and other key stakeholders access to relevant policy documentation.	M
Risk management	Creating a balanced and practical approach between enterprise risk management, technical risk management, and new ways of working. Identifying and removing dependencies on external teams/parties when changing IT services, which may involve delegating approval authority to the team's product/delivery manager. Investing in process automation (e.g. CI/CD) with defined and integrated controls to enforce requirements for the separation of duties. Further to this, employing independent third-party compliance software to suspend the production of deployments until approvals are provided.	M

ITIL management practice	Activities/resources associated with DevSecOps	Impact
Risk management *continued*	Detailing the requirements and risk controls in place in supplier contracts to support the integration of duties while adhering to the organization's security policy. Conducting value stream mapping to identify and minimize process handoffs and approvals.	M
Service validation and testing	Test data management is a key element that helps to ensure continued stability, reliability, availability, and security.	M
Strategy management	Integrating duties to balance regulatory requirements with speed of execution.	M
Workforce and talent management	Training and coaching staff and other relevant stakeholders in how to build security into development and operations work.	M
Business analysis	Understanding the security policies, standards, risks, potential threats, and vulnerabilities in internal and external environments, and translating them into requirements for development and operations teams. Incorporating security requirements into the product backlog.	L
Infrastructure and platform management	Security management can enhance infrastructure and platform management (especially when leveraging infrastructure as code) with guidance on secure standards and training, privacy reviews, threat modelling, credential management, and data security. Information security management and risk management should be an integral part of the daily work of practitioners.	L
Software development and management	Enhancing software development with guidance on secure coding standards and training, privacy reviews, threat modelling, code analysis, source code and credential management, and data security. Information security management and risk management should be an integral part of the daily work of practitioners.	L

The ITIL story: DevSecOps

Henri: *The integrity and security of data are fundamental to the way the teams at Axle Car Hire work. When working quickly to deliver new app features at a high cadence, there is a risk of introducing security loopholes which could be exploited.*

Marco: *All our staff are trained to be aware of how their behaviour can compromise our security. They follow security processes and can detect, prevent, and correct security incidents.*

4.5.3 Peer review

Definition: Peer review

A judgement on a piece of scientific or other professional work by others working in the same area. When applied in software development, a work product is examined by its developer and one or more colleagues in order to evaluate its technical content and quality. This contributes to assured conformance.

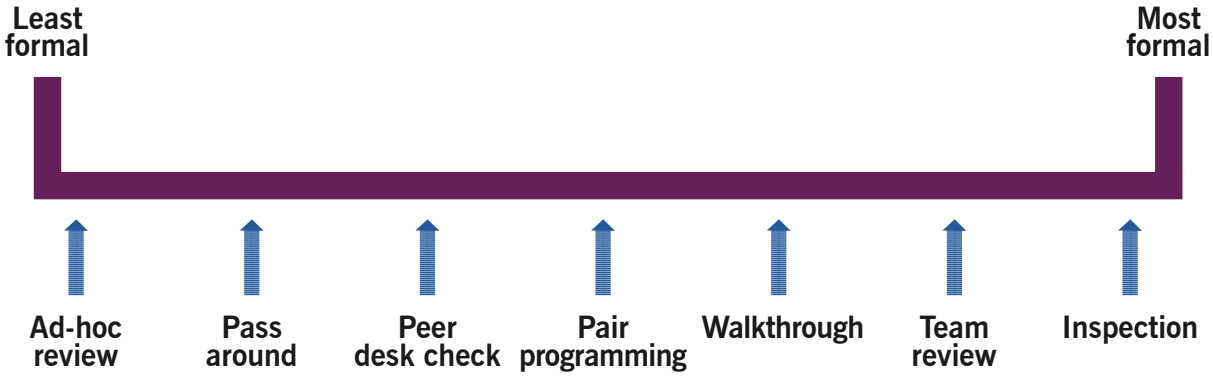

Figure 4.27 Peer review formality spectrum

Based on the demonstrated value of peer reviews in the engineering industry, numerous industry experts have identified it as a very desirable development practice. Experience has shown that problems (defects) are eliminated earlier if a development process incorporates peer reviews; these reviews are as effective as, or even more effective than, testing.[18]

Peer review provides a disciplined engineering practice for detecting and correcting defects in design artefacts. It has also been found to be one of the most effective ways of promoting the quality and productivity of design processes in software engineering and other engineering disciplines, including electrical, civil, mechanical, and fire protection engineering.

Data that is collected during the peer review process is used to correct defects, and to evaluate and improve the development process itself.

The peer review approach can consist of one or more of the following:

- inspection
- team review
- walkthrough
- pair programming
- peer desk check
- pass around
- ad-hoc review.

These approaches are illustrated in Figure 4.27 in order of formality. In addition, the activities typically included in different types of peer reviews are outlined in Table 4.24 (from Wiegers, 2002; reprinted by permission of Pearson Education Inc., NY).

Table 4.24 Activities in different peer review approaches

Review type	Activities				
	Planning	Preparation	Meeting	Correction	Verification
Inspection	Yes	Yes	Yes	Yes	Yes
Team review	Yes	Yes	Yes	Yes	No
Walkthrough	Yes	No	Yes	Yes	No
Pair programming	Yes	No	Continuous	Yes	Yes
Peer desk check, pass around	No	Yes	Possibly	Yes	No

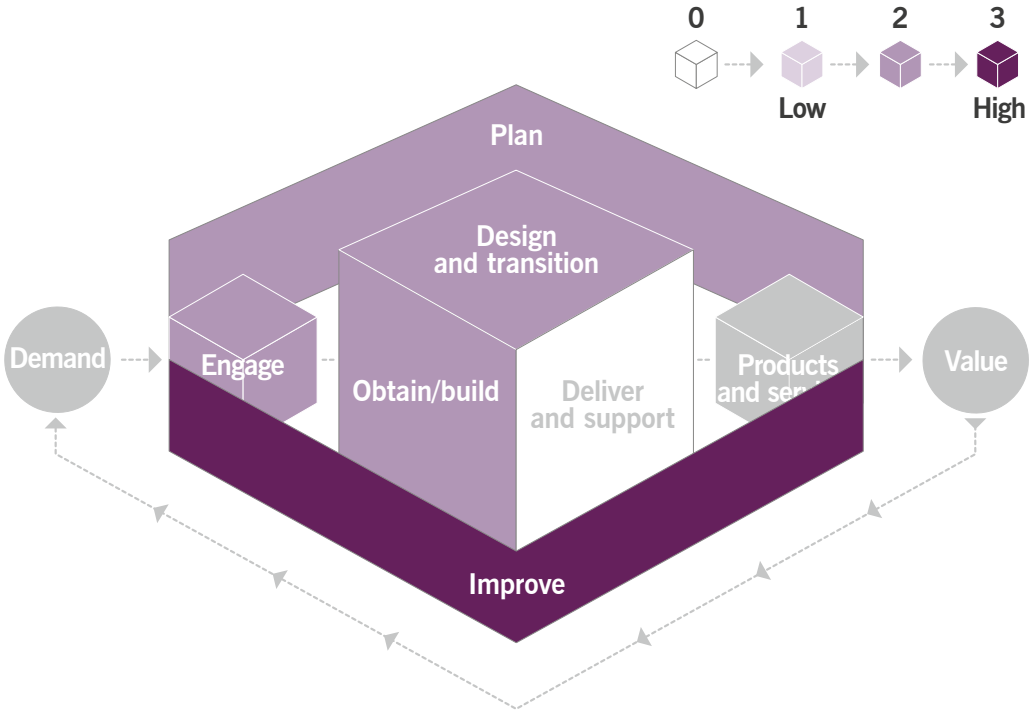

Figure 4.28 Heat map of the contribution of peer review to the service value chain

Ad-hoc review	No	No	Yes	Yes	No

Table 4.25 Practices for which peer review is relevant

ITIL management practice	Activities/resources associated with peer review	Impact
Risk management	Reducing the risk of an unauthorized change being developed and released into production. Cross-checking between identification and assessment of risks.	H
Software development and management	Inspecting development work between peers to increase the quality of code to ensure that it effectively satisfies demand and performance expectations.	H
Change enablement	Colleagues acting as change authorities by performing peer reviews on standard or low-risk changes. Authorizing some changes by peer review or initial assessment of change requests.	M
Continual improvement	Reviewing work done as part of continual improvement initiatives, to help increase the quality of the outcomes achieved.	M
Infrastructure and platform management	Inspecting infrastructure and platform components to increase their quality.	M
Knowledge management	Reviewing knowledge articles and similar documentation to help eliminate biases and increase the quality of communication across the organization.	M
Problem management	Reviewing workarounds and proposed fixes to errors to increase their quality.	M
Architecture management	Conducting walkthroughs of proposed changes to technology architecture to ensure that the changes align with agreed blueprints and roadmaps.	L

Figure 4.28 shows the contribution of peer review to the service value chain.

Table 4.25 outlines the practices for which peer review is relevant.

The ITIL story: Peer review

Su: *Our app development team works collaboratively and conducts regular scheduled peer reviews. We benefit from the expertise and experience of our colleagues, who review each other's work, and find and correct issues before they reach the live environment.*

Solmaz: *We promote an open and blame-free culture, which means that individuals feel comfortable when sharing their work with their counterparts. This helps to build strong, impactful services that create value for all stakeholders.*

4.6 Summary

In Chapter 2, five important organizational objectives enabling high-velocity IT were described. To support the achievement of these objectives, organizations may employ numerous techniques and models. Some of these have been developed recently, whereas others have been adapted from previously adopted operating models and management approaches. Chapter 4 explored some popular and important techniques.

In the chapter, the techniques are grouped around the high-velocity objectives; however, most of them contribute to achievement of multiple objectives to some extent.

The HVIT techniques are universally applicable in the context of many practices. To help with adopting them in the practices, heat maps of their relative contributions to the service value chain are provided.

Practitioners are invited to treat this chapter as a multifunctional toolset, applying the tools according to the context and work tasks being performed. Implementing the techniques described here should not become an objective on its own; they should always be treated as means to achieve the organization's objectives. This applies to the other chapters of this publication and to ITIL in general: the tools should be adopted and adapted to meet the organization's needs.

CHAPTER 5

CONCLUSION

5 Conclusion

Digital technology has disrupted business in many industries, introducing new opportunities and new challenges. Business products, services, and operations have all undergone significant change, known as digital transformation, and this change requires new approaches to the management of IT and business.

To meet these requirements, many methods, techniques, and tools have been developed. The number and variety of these, and deciding how best to use them, can present challenges, and it is not always easy to select the appropriate approach. Beyond changes to products, services, and operations, digital transformation also involves cultural and organizational changes, which come with their own difficulties.

Leaders and practitioners in business and IT should understand the landscape of digital transformation, and be able to define objectives, adopt effective behaviour patterns, and employ appropriate techniques in order to succeed.

This publication provides an overview of the key concepts of digital transformation and high-velocity business and IT management. It suggests a set of objectives and behavioural patterns that will help to transform a business, enabling it to get the most out of digital technology. Finally, it describes a collection of useful techniques and methods that may support each of the objectives. The ITIL SVS provides an overall structure that will help in the practical application of high-velocity IT.

To get the most out of *ITIL® 4: High-velocity IT*, it should be studied alongside the ITIL management practice guides, which are available online and provide detailed, practical recommendations for all 34 practices. They include hands-on guidance that can be applied in the context of all ITIL 4 publications.

All ITIL publications are holistic and focused on value. They address the four dimensions of service management and help to manage resources in a way that enables value creation for the organization, its customers, and other stakeholders.

ITIL® 4: Direct, Plan and Improve provides guidance on aligning product and service management with today's business requirements, driving successful organizational transformation, and embedding continual improvement into an organization's culture at every level.

ITIL® 4: Drive Stakeholder Value contains guidance on establishing, maintaining, and developing effective service relationships. It leads organizations on a service journey in their roles as service provider and service consumer, helping them to interact and communicate effectively at every step.

ITIL® 4: Create, Deliver and Support provides guidance on the cultural and team management aspects of product and service management, and an overview of the various tools and technologies that support service management. It demonstrates how to integrate management practices into end-to-end value streams.

END NOTE: THE ITIL STORY

End note: The ITIL story

The new functionality for Axle's booking app has proved to be a great success and more customers than ever before are using the app to make bookings for cars. By adopting a culture and techniques that support a high-velocity way of working, the development team were able to quickly and effectively implement improvements to the app that are creating real value for Axle.

Development of further improvements to the app is still ongoing as and when requirements for new features are identified. Following the success of the work on the new app functionality, Henri and the team are now looking at how they can adopt a high-velocity approach in other areas of the business to help Axle grow and improve as a digital organization. The techniques that were used for the app development work, along with the lessons learned, have been documented and shared with those teams working on similar projects. Such a positive experience means that Axle is now ready to take the next step on its journey of digital transformation.

FURTHER RESEARCH

Further research

AXELOS publications

AXELOS (2020) *ITIL® 4: Create, Deliver and Support*. TSO, London.

AXELOS (2020) *ITIL® 4: Direct, Plan and Improve*. TSO, London.

AXELOS (2020) *ITIL® 4: Drive Stakeholder Value*. TSO, London.

AXELOS (2019) *ITIL® Foundation: ITIL 4 Edition*. TSO, London.

AXELOS (2018) *A Guide to AgileSHIFT®*. TSO, London.

AXELOS (2017) *Managing Successful Projects with PRINCE2®*. TSO, London.

AXELOS (2016) *ITIL® Practitioner Guidance*. TSO, London.

AXELOS (2015) *PRINCE2 Agile®*. TSO, London.

AXELOS (2015) *RESILIA®: Cyber Resilience Best Practice*. TSO, London.

Cabinet Office (2011) *Managing Successful Programmes*. TSO, London.

Office of Government Commerce (2010) *Management of Risk: Guidance for Practitioners*. TSO, London.

Office of Government Commerce (2010) *Management of Value*. TSO, London.

End notes

1. AXELOS (2011) ITIL V3 and BiSL: Sound guidance for business IT alignment from a business perspective: https://www.axelos.com/case-studies-and-white-papers/itil-and-bisl-sound-guidance-for-business [accessed 14 January 2020]. This case study relates to ITIL V3, but the concepts discussed remain valid.

2. Westrum, R. (2014) A typology of organisational cultures: www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf [accessed 14 January 2020].

3. Derived from Lumiere, in Johnson, P. A. (2014) *Making Light Work* Fairday Books, Sheffield.

4. Dr Amy Edmondson, Harvard Business School, who coined the term 'psychological safety': www.youtube.com/watch?v=LhoLuui9gX8 [accessed 14 January 2020].

5. Cook, R. I. (2002) How complex systems fail. Cognitive Technologies Laboratory, University of Chicago: https://www.researchgate.net/publication/228797158_How_complex_systems_fail [accessed 14 January 2020].

6. The term 'mental health' is used here to clearly mean those with a long-term and possibly latent condition that is capable of clinical diagnosis. This is to distinguish it from situational stress that anyone can suffer during major changes but which is reduced once the situational aspect, such as fear of being made redundant, has been managed.

7. Heuristic ways of working contrast with algorithmic ways of working, where a set of unambiguous instructions are given for performing a task. Algorithms always produce the 'correct' result.

8. Wikipedia, Cynefin framework: https://en.wikipedia.org/wiki/Cynefin_framework [accessed 14 January 2020].

9. England, R.E. (2013) *Plus! The Standard+Case Approach: See Service Response in a New Light*. CreateSpace Independent Publishing Platform, California.

10. Adapted from *The Toyota Kata Practice Guide* by Mike Rother. Licensed under the Creative Commons Attribution 3.0 unported licence from http://www-personal.umich.edu/~mrother/Supporting_Materials.html [accessed 14 January 2020].

11. Wikipedia, OODA loop: https://en.wikipedia.org/wiki/OODA_loop [accessed 14 January 2020].

12. Microsoft, Shift left to make testing fast and reliable: https://docs.microsoft.com/en-us/azure/devops/learn/devops-at-microsoft/shift-left-make-testing-fast-reliable [accessed 14 January 2020].

13. Principles of Chaos Engineering: http://principlesofchaos.org/ [accessed 14 January 2020].

14. Netflix Tech Blog, The Netflix Simian Army: https://medium.com/netflix-techblog/the-netflix-simian-army-16e57fbab116 [accessed 14 January 2020].

15. Abdula, M., Averdunk, I., Barcia, R., Brown, K., and Emuchay, N. (2018) *The Cloud Adoption Playbook: Proven Strategies for Transforming Your Organization with the Cloud*. John Wiley & Sons, Indianopolis.

16. Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (editors) (2016) *Site Reliability Engineering: How Google Runs Production Systems*. Google Books.

17. DevOps Audit Defense Toolkit: https://itrevolution.com/devops-audit-defense-toolkit/ [accessed 14 January 2020].

18. Garousi, V. (2010) Applying peer reviews in software engineering education: an experiment and lessons learned. *IEEE Transactions on Education* (Volume 53, Issue 2, May): https://ieeexplore.ieee.org/document/5152941 [accessed 14 January 2020].

Other publications

Bell, S. C. and Orzen, M. A. (2010) *Lean IT: Enabling and Sustaining Your Lean Transformation*. Productivity Press.

Blackburn, S. (2003) *Ethics: A Very Short Introduction*. OUP, Oxford.

England, R (2013) *Plus! The Standard+Case Approach: See Service Response in a New Light*. CreateSpace Independent Publishing Platform, California.

Gianotten, M. (2017). *Digital Empathy: When Tech Meets Touch*. Giarte, Amsterdam.

Godin, S. (2018) *This is Marketing: You Can't be Seen Until You Learn to See.* Portfolio Penguin.

Humble, J. and Farley, D. (2010) *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison Wesley.

Johnson, P. A. (2014) *Making Light Work: Rethinking the Service Organisation.* Fairday Books. https://fairday.co.uk/product/making-light-work-rethinking-the-service-organisation-peter-a-johnson/ [accessed 30 January 2020].

Liker, J. (2004) *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer.* McGraw-Hill.

Murphy, N. R., Beyer, B., Jones, C. and Petoff, J. (2016) *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media.

Reinertsen, D. G. (2009) *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas.

Ries, E. (2011) *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*. Portfolio Penguin.

Rother, M. (2009) *Toyota Kata: Managing People for Improvement, Adaptiveness and Superior Results*. McGraw-Hill.

Snowden, D. (2011) The Cynefin framework. https://cognitive-edge.com/ [accessed 30 January 2020].

Wiegers, K. (2002) *Peer Reviews in Software: A Practical Guide*. Pearson Education, NY.

Williams, B. (1993) *Morality: An Introduction to Ethics*. Cambridge University Press.

Websites

Cognitive Edge: The Future, Backwards https://cognitive-edge.com/methods/the-future-backwards/ [accessed 30 January 2020].

DeLuccia IV, J., Gallimore, J., Kim, G. and Miller, B. (2015). The DevOps Audit Defense Toolkit. IT Revolution https://cdn2.hubspot.net/hubfs/228391/_archive/Corporate/DevOps_Audit_Defense_Toolkit_v1.0.pdf [accessed 30 January 2020].

Klein, G. (2007) Performing a Project Premortem *Harvard Business Review* September 2007. https://hbr.org/2007/09/performing-a-project-premortem [accessed 30 January 2020].

The Netflix Simian Army (2011) https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116 [accessed 30 January 2020].

Sigler, E. (2014) What is ChatOps? https://www.pagerduty.com/blog/what-is-chatops/ [accessed 30 January 2020].

Snowden, D. J. and Boone, M. E. (2007) A Leader's Framework for Decision Making. *Harvard Business Review* November 2007 https://hbr.org/2007/11/a-leaders-framework-for-decision-making [accessed 30 January 2020].

Zacarias, D. 20 Product Prioritization Techniques: A Map and Guided Tour https://foldingburritos.com/product-prioritization-techniques/ [accessed 30 January 2020].

GLOSSARY

Glossary

adaptive system

A system in which the behaviour of agents changes and they self-organize in response to events.

AIOps

The application of machine learning and big data to IT operations to receive continuous insights which provide continuous fixes and improvements via automation. Also referred to as 'artificial intelligence for IT operations' or 'algorithmic IT operations'.

Andon

A system, either manual or automated, used to notify workers and other parts of an organization of quality or process issues. The Andon system originated in manufacturing, but is now widely used in IT.

backlog

A list of new features for a product. The list may consist of user stories which are structured in a way that describes who wants the feature and why. It is also a generic term that can be defined in terms of releases, sprints, and products.

big data

The use of very large volumes of structured and unstructured data from a variety of sources to gain new insights.

blameless post-mortem

A non-judgemental description and analysis of the circumstances and events that preceded an incident.

blockchain

An open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way.

business case

A justification for the expenditure of organizational resources, providing information about costs, benefits, options, risks, and issues.

chaos engineering

The discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production.

chaos monkey

A tool that tests the resilience of IT systems by intentionally disabling components in production to test how remaining systems respond to the outage.

complex adaptive systems

Systems that adapt in, and co-evolve with, a changing environment, resulting in:

- behaviour that is not predicted by the behaviour of parts of the system
- the inability to examine the system in isolation from the other systems in its environment.

complex system

A system in which agents' interactions are dynamic and often unpredictable.

complexity thinking

A systems thinking approach based on the recognition and understanding of the various levels of complexity inherent in the systems and the context in which they operate.

compliance

Both the act and result of ensuring that a standard or set of guidelines is followed, or that proper, consistent accounting or other practices are being employed.

configuration management database (CMDB)

A database used to store configuration records throughout their lifecycle. The CMDB also maintains the relationships between configuration records.

containerization

The technique of packaging software into standardized lightweight, stand-alone, executable units for development, shipment, and deployment.

continuous delivery

An approach to software development in which software can be released to production at any time. Frequent deployments are possible, but deployment decisions are taken case by case, usually because organizations prefer a slower rate of deployment.

continuous deployment

An approach to software development in which changes go through the pipeline and are automatically put into the production environment, enabling multiple production deployments per day. Continuous deployment relies on continuous delivery.

continuous integration

An approach to integrating, building, and testing code within the software development environment.

control

The means of managing a risk, ensuring that a business objective is achieved, or that a process is followed.

cost of delay

The benefits that are expected to be lost when the launch or update of a service offering is delayed.

cultural fit

The ability of an employee or a team to work comfortably in an environment that corresponds with their own beliefs, values, and needs.

customer experience (CX)

The sum of functional and emotional interactions with a service and service provider as perceived by a customer.

customer journey

The complete end-to-end experience that service customers have with one or more service providers and/or their products through touchpoints and service interactions.

customer orientation

An approach to sales and customer relations in which staff focus on helping customers to meet their long-term needs and wants.

cycle time

The amount of time required to complete a discrete unit of work, converting inputs into outputs.

data

Information that has been translated into a form that is efficient for movement or processing.

data analytics

A branch of data science focused on analysing raw data in order to draw conclusions about it, using highly automated techniques.

definition of done

A checklist of the agreed criteria for a proposed product or service.

design thinking

A practical and human-centred approach used by product and service designers to solve complex problems and find practical and creative solutions that meet the needs of an organization and its customers.

digital organization

An organization that is enabled by digital technology to do business significantly differently, or to do significantly different business.

digital product

A product is digital when digital technology plays a significant role in its goods, resources, or associated service interactions.

digital technology

Technology that digitizes something or processes digital data. Digital technology refers to information technology (IT) and the parts of operational technology (OT) that have been digitized. *See also* digitization.

digital transformation

The use of digital technology to enable a significant improvement in the realization of an organization's objectives that could not feasibly have been achieved by non-digital means.

digitization

The process of transforming something (e.g. text, sound, or images) from analogue to digital form by expressing the information in binary digits.

double-loop learning

A form of learning that takes place when examining the purpose and function of work being done, without taking the existing organizational structure for granted. Also known as 'reframing'.

early-life support (ELS)

A period of time associated with the release of service components to users, when additional resources are allocated to user support and service operations. Early-life support can also be applied to the onboarding or offboarding of users from a service.

emotional intelligence

The ability to understand the way people feel and react, and to use this skill to make good judgements and to avoid or solve conflicts.

epic

A high-level definition of a requirement that has not yet been sufficiently refined or understood. Eventually an epic will be refined or broken down into several user stories and requirements.

ethics

A system of principles that defines what is good for individuals and society.

gap analysis

An activity that compares two sets of data and identifies the differences; for example, comparing a set of requirements with the actual delivery, or the current state of an organization with a target future state.

governance

The means by which an organization is directed and controlled.

high-velocity IT

The application of digital technology for significant business enablement, where time to market, time to customer, time to change, and speed in general are crucial. High velocity is not restricted to fast development; it is required throughout the service value chain, from innovation at the start, through development and operations, to the actual realization of value.

high-velocity IT operating model

An IT operating model where digital technology plays a major role in the co-creation of value.

improvement

A deliberately introduced change that results in increased value for one or more stakeholders.

indicator

A metric that is used to assess and manage something.

information model

The construct of information, related to the taxonomy and relationships of data to other data, required to present and share content in a meaningful and representative way.

information technology

The application of digital technology to store, retrieve, transmit, and manipulate data (data processing), often in the context of a business or other kind of organization.

integration of duties

Having a task that is prone to fraud or error performed by one person because other controls have been applied. This serves as an alternative to separation (or segregation) of duties.

intelligent disobedience

Deliberately disobeying or disregarding rules in order to avoid a dangerous situation, or 'doing the right thing'.

ITIL continual improvement model

A model which provides organizations with a structured approach to implementing improvements.

ITIL guiding principles

Recommendations that can guide an organization in all circumstances, regardless of changes in its goals, strategies, type of work, or management structure.

ITIL service value chain

An operating model for service providers that covers all the key activities required to effectively manage products and services.

ITIL service value chain activity

A step of the value chain that an organization takes in the creation of value.

Kanban

A method for visualizing work, identifying potential blockages and resource conflicts, and managing work in progress.

lead time

The time taken to complete the execution of a process, usually measured from a specific perspective (e.g. that of the customer).

Lean culture

A work environment where trust, respect, curiosity, enquiry, playfulness, and intensity all co-exist to support learning and discovery.

machine learning

An applied form of artificial intelligence, based on the principle of systems responding to data, and adapting their actions and outputs as they are continually exposed to more of it.

measurement

A means of decreasing uncertainty based on one or more observations that are expressed in quantifiable units.

metric

A measurement or calculation that is monitored or reported for management and improvement.

microservices

A variation of the service-oriented architecture in which an application is designed and developed as a set of small, loosely coupled services, each running in its own process and using lightweight mechanisms to communicate.

minimum viable approach

A technique of providing users with the minimum set of capabilities to enable rapid assessment and learning. Minimum viable approaches can be applied to products, services, practices, processes, and process outputs.

multi-sourcing

The use of multiple service providers offering similar (if not the same) services, balancing the risks of relying on a single provider with the overhead of managing work across multiple providers.

operating model

A conceptual and/or visual representation of how an organization co-creates value with its customers and other stakeholders, as well as how the organization runs itself.

operation

The routine running and management of an activity, product, service, or other configuration item.

operational technology

The application of digital technology for detecting or causing changes in physical devices through monitoring and/or control.

organization

A person or a group of people that has its own functions with responsibilities, authorities, and relationships to achieve its objectives.

paradoxical intervention

A management technique that relies on asking staff for something in order to achieve the opposite result; for example, asking for more risks to be taken to decrease their impact.

peer review

A judgement on a piece of scientific or other professional work by others working in the same area. When applied in software development, a work product is examined by its developer and one or more colleagues in order to evaluate its technical content and quality. This contributes to assured conformance.

plan

The value chain activity that ensures a shared understanding of the vision, current status, and improvement direction for all four dimensions and all products and services across an organization.

product owner

A role in a Scrum team that is responsible for: defining user stories and acceptance criteria; prioritizing user stories in a backlog; clarifying requirements and answering questions from the development team; and assisting with demonstrations to customers.

RACI

A model used to help define roles and responsibilities. RACI stands for responsible, accountable, consulted, and informed.

reconstructing for service agility

An approach to organizing knowledge work and service provision that reflects its complex and social nature.

report

A detailed communication of information or knowledge about a topic or event.

rich data

Data that is accurate, precise, and subjected to rigorous quality control.

risk

A possible event that could cause harm or loss or make it more difficult to achieve objectives. Can also be defined as uncertainty of outcome and can be used in the context of measuring the probability of positive outcomes as well as negative outcomes.

role

A role is a set of responsibilities, activities, and authorizations granted to a person or team in a specific context.

safety culture

A climate in which people are comfortable being (and expressing) themselves.

Scrum

An iterative, timeboxed approach to product delivery that is described as 'a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value' (*The Scrum Guide* by Ken Schwaber and Jeff Sutherland, updated 2017).

service empathy

The ability to recognize, understand, predict, and project the interests, needs, intentions, and experiences of another party, in order to establish, maintain, and improve the service relationship.

service integration and management

The coordination and orchestration of work across all suppliers involved in the development and delivery of products and services.

service interaction

A reciprocal action between a service provider and a service consumer that co-creates value.

service level

One or more metrics that define expected or achieved service quality.

service level agreement (SLA)

A documented agreement between a service provider and a customer that identifies both the services required and the expected level of service.

service management

A set of specialized organizational capabilities for enabling value for customers in the form of services.

service mindset

An important component of the organizational culture that defines an organization's behaviour in service relationships. A service mindset includes the shared values and guiding principles adopted and followed by an organization.

service quality

The totality of a service's characteristics that are relevant to its ability to satisfy stated and implied needs.

service value system (SVS)

A model representing how all the components and activities of an organization work together to facilitate value creation.

service-dominant logic

A mental model of an (economic) exchange in which organizations co-create value by applying their competencies and other resources for the benefit of each other.

shift-left approach

An approach to managing work that focuses on moving activities closer to the source of the work, in order to avoid potentially expensive delays or escalations. In a software development context, a shift-left approach might be characterized by moving testing activities closer to (or integrated with) development activities. In a support context, a shift-left approach might be characterized by providing self-help tools to end-users.

Simian army

An open-source toolset for chaos engineering developed by Netflix®.

single-loop learning

The type of learning that takes place when fixing problems within the present organizational structure so that the system will function better without altering its structure.

site reliability engineering (SRE)

A discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems with the goal of creating ultra-scalable and highly reliable software systems.

sprint

A fixed timeframe (typically of 2–4 weeks) for creating selected features from the backlog.

stakeholder

A person or organization that has an interest or involvement in an organization, product, service, practice, or other entity.

strategy

A broad approach or course of action defined by an organization for achieving its objectives.

stress prevention

The prevention, monitoring, and remediation of unhealthy tension in the workplace.

systems thinking

A holistic approach to analysis and decision-making that focuses on the relationship between a system's components and the way the system works, both as a whole and within the context of larger systems.

technical debt

The total rework backlog accumulated by choosing workarounds instead of system solutions that would take longer.

theory of constraints

A methodology for identifying the most important limiting factor (i.e. constraint, often referred to as a bottleneck) that stands in the way of creating value, and then systematically correcting that constraint until it is no longer the limiting factor.

time value profile

A depiction of the change in value of a requirement, output, or outcome over time.

Toyota Kata

A mental model and behaviour pattern for scientific thinking and routines for practice and coaching.

user experience (UX)

The sum of the functional and emotional interactions with a service and service provider as perceived by a user.

user story

A technique in Agile software development that uses natural language to describe desired outcomes and benefits from the point of view of a specific persona (typically the end-user), usually in the form of 'who, what, and why'.

utility

The functionality offered by a product or service to meet a particular need. Utility can be summarized as 'what the service does' and can be used to determine whether a service is 'fit for purpose'. To have utility, a service must either support the performance of the consumer or remove constraints from the consumer. Many services do both.

value

The perceived benefits, usefulness, and importance of something.

value proposition

An explicit promise made by a service provider to its customers that it will deliver a particular bundle of benefits.

value stream

A series of steps an organization undertakes to create and deliver products and services to consumers.

value stream map

A visual representation of a service value stream which shows the flow of work, information, and resources.

value stream mapping

A Lean management technique to visualize the steps needed to convert demand into value, used to identify opportunities to improve.

version control

The administrative management of sources and artefacts of information systems, products, and services.

vision

A defined aspiration of what an organization would like to become in the future.

warranty

Assurance that a product or service will meet agreed requirements. Warranty can be summarized as 'how the service performs' and can be used to determine whether a service is 'fit for use'. Warranty often relates to service levels aligned with the needs of service consumers. This may be based on a formal agreement, or it may be a marketing message or brand image. Warranty typically addresses such areas as the availability of the service, its capacity, levels of security, and continuity. A service may be said to provide acceptable assurance, or 'warranty', if all defined and agreed conditions are met.

ACKNOWLEDGEMENTS

Acknowledgements

AXELOS Ltd is grateful to everyone who has contributed to the development of this guidance and in particular would like to thank the following people.

Lead editor

Mark Smalley

Mark Smalley, also known as the IT Paradigmologist, thinks, writes, and speaks about IT 'paradigms' (our changing perspectives on IT). Mark is a trainer/consultant at Smalley.IT and master trainer for GamingWorks' The Phoenix Project and MarsLander business simulations. He is global ambassador at the DevOps Agile Skills Association, and has contributed to many bodies of knowledge in the IT management domain.

Authoring team

Akshay Anand

Akshay is a product ambassador at AXELOS, working on the development of new guidance and research within the ITSM portfolio. With experience from the US, UK, and India, he previously advised Fortune 100 clients on their ITSM capabilities, implemented toolsets such as Remedy and ServiceNow, and headed up global ITSM activities at Macmillan Publishing. More recently, Akshay has focused on bringing together Agile development teams and ITSM professionals to address the challenges posed by emerging technologies. He tweets as @bloreboy.

Dan Ashby

Dan is a quality and testing specialist with experience in leading quality within companies such as eBay, AstraZeneca, and Photobox. He specializes in building and implementing software quality and testing strategies and teams, and is an enthusiastic blogger, writer, podcaster, and public speaker. A strong believer in creating safe spaces for people to learn about the craft of testing, Dan is deeply involved in the Testing Essentials meetups and training courses partnered with the Ministry of Testing to help grow practical testing skills and knowledge that can be applied in any context.

Ariana Bucio

Ariana is an experienced consultant, trainer, researcher, and auditor in ITSM. She has helped several organizations in Latin America to adapt and adopt ITSM best practices. She writes about ITSM and speaks at conferences and on webinars. Ariana contributes as a reviewer of the SDI certification standard. Her passion for service desks and best practices is shown as SDI lead auditor and consultant, promoting value creation for the organization and its customers.

Ana Yasmeen Chong Rosales

Ana is a passionate trainer, consultant, speaker, and professor in the Latin America region. Her experience leading various teams and projects has led her to believe that inspiring people is the most important action. She is CEO at BPGurus, a company that specializes in consulting and training in best practices for digital services and ITSM. Her career has focused on helping organizations to adopt integral solutions in ITSM and e-learning. She is a lead auditor at the Service Desk Institute and participates in several forums focused on the empowerment of women in IT.

James Finister

James is a partner in TCS's Enterprise Agility consulting team, focusing on implementing common lessons across both the IT and business ecosystems. He is a contributor to many sources of industry best practice, including ISO standards, is an honorary lifetime member of *it*SMF UK, and holder of its lifetime achievement award.

Reni Friis

Reni has more than 20 years of experience in ITSM, with a focus on making Lean make sense in an IT context. Reni is an accomplished IT management consultant, trainer, and international speaker on topics such as Lean IT, DevOps, SIAM, and leadership. She is a permanent member of the Lean IT Association content board, and has also contributed to the development and improvement of frameworks for both VeriSM, SIAM, and DevOps. Her style is characterized by humour, energy, and getting straight to the (value-adding) point of things.

Marco Gianotten

Marco is the founder of the boutique research-consulting firm Giarte. He is a public speaker, author of *Digital Empathy*, and initiator of the Xperience Level Agreement movement. Customer journeys, emotional requirements, and user experience will be the new normal in ITSM. He inspires organizations to rethink, transform, and simplify IT.

Rudolf Greger

Rudolf is a design philosopher with a mission to improve people's lives. With 30 years of experience, his vocation is designer and his profession is design thinking coach. He co-founded GP designpartners, a central European industrial design consultancy, and founded the design thinking tank to promote design thinking. Rudolf is a C-level sparring partner; he organizes design jams, writes about design, and speaks at conferences.

Jon Hall

Jon is an innovation-focused product manager with an extensive track record at the leading edge of ITIL-focused tooling and practice. He is well known in the DevOps community for his exploration of service management's necessary evolution to support and enable the success of DevOps through new ways of working. He is a recognized and cited expert on swarming as a transformational work practice. His work has been cited by DevOps luminaries such as Gene Kim, and he regularly speaks at both ITSM and DevOps events.

Ian Jones

Ian is a pragmatic architect and adviser of digital and ICT organizations who has successfully led many IT operating models and IT service management projects. He has worked with various organizations across all industry sectors, ranging from start-ups to enterprises with mission-critical services. His extensive, hands-on experience in integrating modern delivery practices (e.g. Agile, DevOps, Lean, and ITIL) has been recognized by his peers with a number of industry awards. Ian lives with his family in Brisbane, Australia.

Ola Källgården

Ola is passionate about striking the right balance between order and chaos. He bends and twists frameworks, models, and philosophies to make them more useful. He has experience in IT roles such as developer, business analyst, project manager, organizational change manager, and management consultant. Apart from guiding and coaching clients on behalf of Oligo Consulting, Ola is also a public speaker and an acclaimed and accredited ITIL and DevOps trainer.

Allan Kelly

Allan is a recognized expert in software development processes and management. As an Agile coach, consultant, and adviser, he helps teams and companies enhance their processes and improve their digital products. He invented Value Poker, Time-Value Profiles and Retrospective Dialogue Sheets. He is the author of the letter *Dear Customer, The Truth about IT Projects* and books including *Project Myopia*, *Continuous Digital*, *Xanpan* and his latest, *The Art of Agile Product Ownership*. His blog is at https://www.allankellyassociates.co.uk/blog/ and he tweets as @allankellynet.

Donna Knapp

Donna has more than 30 years' experience in the IT industry working as a practitioner, consultant, and educator. She is ITSM Academy's curriculum development manager. Donna's years of practical experience and love of learning make her a sought-after speaker on issues surrounding ITSM and the service desk. Donna is the author of *The ITSM Process Design Guide: Developing, Reengineering, and Improving IT Service Management*, as well as two college textbooks, *A Guide to Service Desk Concepts* and *A Guide to Customer Service Skills for Service Desk Professional*.

Dagfinn Krog

Dagfinn is an IT professional with 25 years' experience in the industry, having worked as an infrastructure, security, and service management consultant. He currently works as an IT security specialist in multi-sourced environments, bridging gaps between technology, service management, and business risk. He is also a long-serving board member of *it*SMF Norway.

Jeffrey Liker

Jeffrey is professor emeritus of industrial and operations engineering at the University of Michigan and president of Liker Lean Advisors. He is author of the international bestseller, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer* and has co-authored nine other books about Toyota, including *The Toyota Way to Service Excellence*, *Designing the Future*, and *The Toyota Way to Lean Leadership*. His articles and books have won 13 Shingo Prizes for Operational Excellence.

Krikor Maroukian

Krikor is a senior modern service management consultant at Microsoft® and the international representative of *it*SMF Hellas. He also has experience in project management and information security management. He focuses on high-performance service management research and development in EMEA. Krikor is a researcher at Henley Business School and is a BoD member of the British Computer Society (Hellenic section). He has authored academic papers on service management, project management, and model-driven business process engineering, and received the Emerald Literati Network Award for Excellence in 2017.

Simone Jo Moore

Simone is known as a service management mixologist and human API (application programming interface). People-connected, knowledge-shared, possibilities-discovered, and potential-realized are her active values. A recognized industry Top-25 thought leader, Simone is a senior consultant, master trainer, course author, podcast co-host, and mentor in frameworks such as VeriSM, ITIL, DevOps, CASM, KCS and SFIA. Simone combines these with a human resources, organizational change, and complementary health background for a deep leadership experience shared via training, workshops, and conferences.

Pasi Nikkanen

Pasi is passionate about driving employee happiness and productivity in service management. He is the chief product officer at HappySignals, podcast host on *HappyToday – The Employee Experience,* and a regular webinar and events speaker. He has worked with internal services and outsourcing at a large Nordic IT company and as a service designer for many large global enterprises.

Mike Orzen

Mike is considered a pioneer of Lean and technology. He is co-author of the award-winning books *Lean IT: Enabling and Sustaining Your Lean Enterprise* and the follow up *The Lean IT Field Guide*. Often referred to as 'the father of Lean IT', Mike is recognized internationally as a thought leader in Lean, Lean IT, and DevOps. He is a recipient of the Shingo Prize for Operational Excellence, a faculty member of the Lean Enterprise Institute, and a fellow, facilitator, and assessor at the Shingo Institute. Mike is also a certified Kundalini Yoga instructor and dedicated yoga and meditation practitioner.

Oleg Skrynnik

Oleg is a managing partner at Cleverics, having worked in IT for more than 20 years (mainly in management positions). He is experienced in establishing and transforming IT departments in large companies, and applies his professional experience in consulting projects, sharing the lessons learned with attendees of his training, masterclasses and business simulations. Oleg co-founded *it*SMF Russia and is a recognized speaker and author. He won first prize at the ITSM in Russia Award in 2014 and 2017 for his articles and wrote *DevOps: A Business Perspective*, on which the EXIN DevOps Foundation exam is based.

Dave Snowden

Dave is director of the Cynefin Centre which focuses on the application of natural science to social systems. He is a leading thinker and practitioner in the fields of complex adaptive systems and narrative theory. He has a long career which encompasses the design of decision support systems, general and strategic management, and (for the last 20+ years) research and development. In 2006 he was director of the EPSRC (UK) research programme on emergence, and in 2007 was appointed to an NSF (US) review panel on complexity science research. He has won numerous awards for his work and publications.

The ITIL storyline

Katrina Macdermid

Katrina is an ITIL 4 global ambassador, author, and creator of an innovative integrated framework, human-centred ITIL service design. She is an ITIL Master, with a solid background in designing and implementing IT operating models that put customer experience at the heart of all ITIL processes. Katrina has been responsible for the successful implementation of key strategic programmes and integrated service management in some of Australia's largest and most iconic organizations. She is a well-known conference speaker on human-centred ITIL service design.

Craig Smith

Craig is a poet and novelist, with three books to his name. He currently works as the managing editor of the content team at AXELOS. Craig commissioned and edited *ITIL Practitioner Guidance*, along with *ITIL and DevOps: Getting Started* and *Using Kanban in ITIL Operations*. Craig also co-wrote *ITIL Foundation Exam Tips*, the most widely read blog on the AXELOS site.

AXELOS ITIL 4 architect team

Akshay Anand	Product ambassador ITSM
Philip Hearsum	ITIL portfolio manager
Roman Jouravlev	ITIL portfolio development manager
Margo Leach	Chief product officer
Heigor Simões de Freitas	Product manager ITSM

ITIL 4 lead editors

Dinara Adyrbayeva	ITIL 4 practice library
David Cannon	*ITIL® 4: Digital and IT Strategy*
Erika Flora	*ITIL® 4: Digital and IT Strategy*
Lou Hunnebeck	*ITIL® 4: Direct, Plan and Improve*
Christian F. Nissen	*ITIL® 4: Drive Stakeholder Value*
Barclay Rae	*ITIL® 4: Create, Deliver and Support*
Stuart Rance	ITIL 4 examinations

Project team

David Atkins	Content delivery and production manager
Rachida Chekaf	Head of translations
Sarah Conner	Production editor
Adrian Crago-Graham	Head of PMO
Ricky Elizabeth	Brand and design manager
Emily Holbrook	Commissioning editor: ITIL supplementary content
James Lord	Senior qualifications and assessment specialist
Michael Macgregor	Project manager
Amy Metcalfe	Project editor
James Robertson	Senior qualifications and assessment specialist
Tom Young	Commissioning editor: ITIL core content

Contributors

Mark Bodman, Mike Fulton, Matthew Helm, Jessica Hinkel, John Kendrick, Caspar Millar, James Monek, Mark Toomey, Ole Westergaard.

AXELOS would like to thank the following for their valuable insight:

Arie van Bennekum, Daniel Billing, Erik van Busschbach, Andrew Campbell, James Cusick, Carmen DeArdo, Rob England, Robert Falkowitz, Dave Favelle, Dave Hahn, Julia Harrison, Dave van Herpen, Erica Morrison, Shuiji Tomita, Karel van Zeeland, Sun Zhenpeng.

INDEX

Index

W

Online Subscription

More ITIL® 4 content for even better value

- **Web-based service**
  access on your PC, laptop, mobile or tablet
- **Latest official guidance**
  all revisions and updates immediately included
- **Quick navigation**
  search for individual phrases or keywords
- **Dedicated support**
  we are here to ensure you get the most from your subscription

For more information please contact **digital@wlt.com** or visit **tiny.cc/ITIL4Subs**

Single-user subscription

- Expand your knowledge
- Explore more publications
- Access from anywhere

Multi-user subscription

- Empower your team
- Bespoke subscriptions
- Get company-wide access

in partnership with
HM Government

HM Government

ITIL® 4: High-velocity IT explores the ways in which digital organizations and operating models function in high-velocity environments, helping aspiring practitioners to understand digital transformation and manage it successfully. Written for service delivery managers and other IT practitioners, it focuses on the characteristics of high-velocity organizations, demonstrating how ITIL's operating model, principles, and practices can provide an approach that will help in the practical application of high-velocity IT.

ITIL® 4: High-velocity IT explores:

- the concepts of high-velocity IT, digital technology, and digital transformation
- what is needed to create a high-velocity IT culture, including key objectives and behavioural models
- a variety of techniques that can be used to achieve these key objectives.