ITpreneurs™
Effective Learning Solutions

DevOps in an ITSM Environment

First Aid Kit

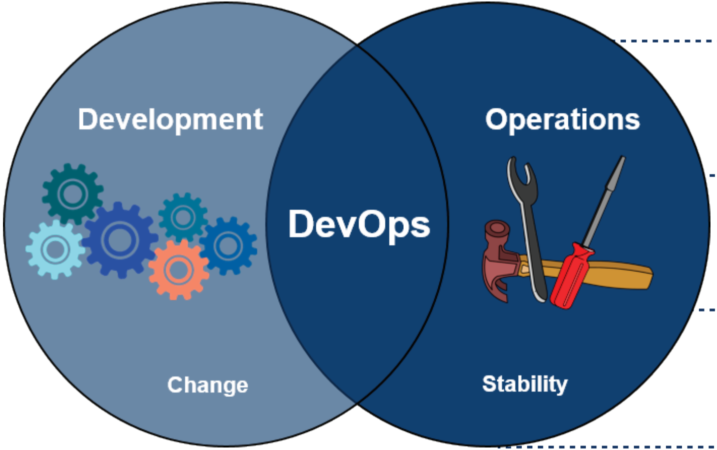
The Best of Two Worlds

DevOps in an ITSM Environment

First Aid Kit

The DevOps and ITSM First Aid Kit is an essential toolkit for the effective utilization of ITSM skill sets in any DevOps organization. The Kit provides essential knowledge about the two different worlds, ITSM and DevOps, and how they complement each other. It helps you know how DevOps movement fits perfectly with ITSM. Since DevOps is gaining momentum, think of this kit before looking elsewhere for the DevOps movement in your organization. However, it does not provide any one-size-fits-all solutions or formulae that ensure a successful DevOps implementation.

Introducing DevOps



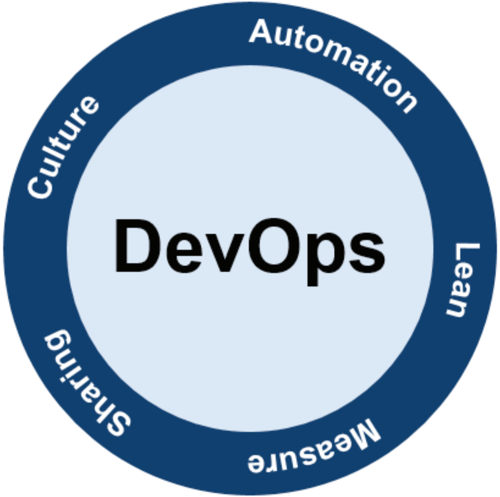
Every organization wants better products and services, easier releases, higher quality, and more fun. DevOps helps organizations delivering quality products and meeting customer's expectations.

The core of DevOps is to improve collaboration between Development and Operations teams. The collaboration helps remove the Wall of Confusion between the two teams.

DevOps does not happen overnight. It might require a separate team to be piloted in an organization. It also requires automation and investment in technology.

You can start practicing DevOps in your organization even in the absence of a separate DevOps team, automation, and investment. You can do this by implementing (some of) the DevOps principles.

Defining DevOps

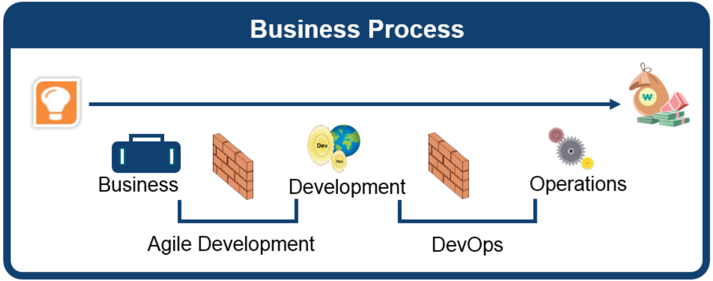


"DevOps shifts the tradition of how IT is organized, how engineers interact. It brings a set of best practices that guides how engineers and IT works that is markedly different than a traditional set of principles. It's a culture of automating, measuring, and sharing in the name of increased efficiencies throughout the software development life cycle."

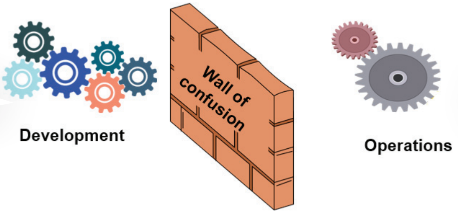
—Jason Hand, DevOps Evangelist at VictorOps, organizer of DevOps Days.

Position DevOps in an Organization

The business demands faster and continuous delivery. However, it is not easy due to various challenges and contracting goals of the Development and Operations teams.



The Development team builds software and applies changes to incorporate new features and fulfill all requirements.



The Operations team focuses on stability, reliability, security, and performance of the systems maintained by them.

DevOps breaks down the Wall of Confusion: One Team, One Goal!

DevOps: Value Proposition

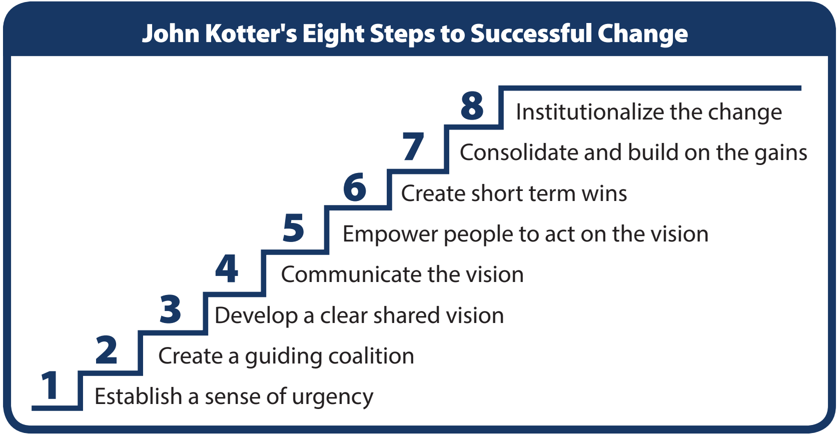
Aligns IT performance and responsiveness to business.

Deploys more robust and reliable releases at a fast pace.

DevOps: An Organizational Change

Does the organizational change start or end with a culture change?

- DevOps introduces not only tools, methods, and technology but requires a different way of working in an organization.
- Organizational Change Management (OCM) supports such a change. One of the models that helps in successful change management is Kotter's 8-step Change Model.



John Kotter's Eight Steps to Successful Change

8 Institutionalize the change

7 Consolidate and build on the gains

6 Create short term wins

5 Empower people to act on the vision

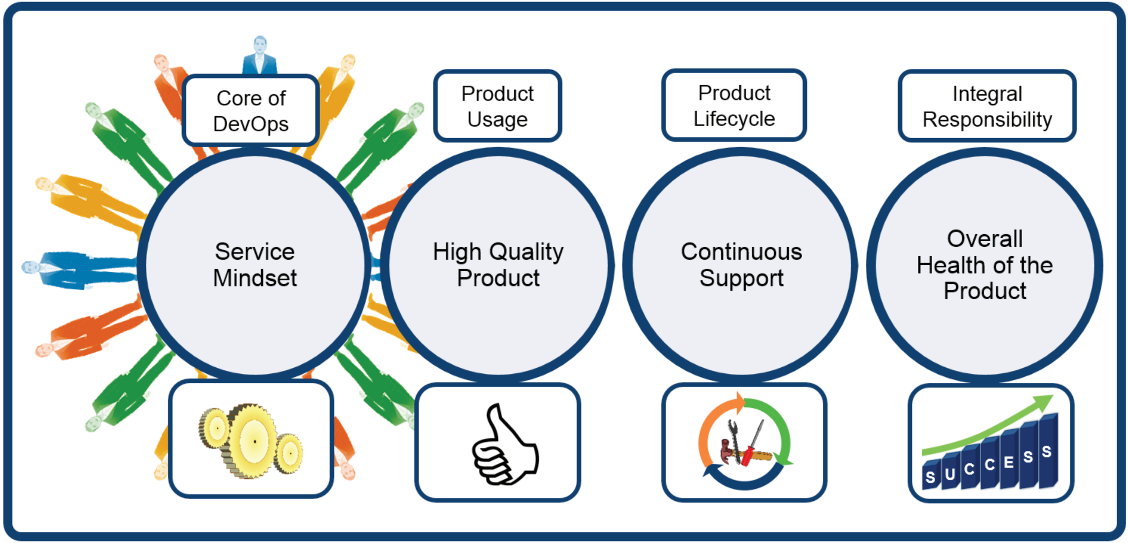
4 Communicate the vision

3 Develop a clear shared vision

2 Create a guiding coalition

1 Establish a sense of urgency

DevOps Culture and its Emphasis on Service



Core of DevOps

Service Mindset

Product Usage

High Quality Product

Product Lifecycle

Continuous Support

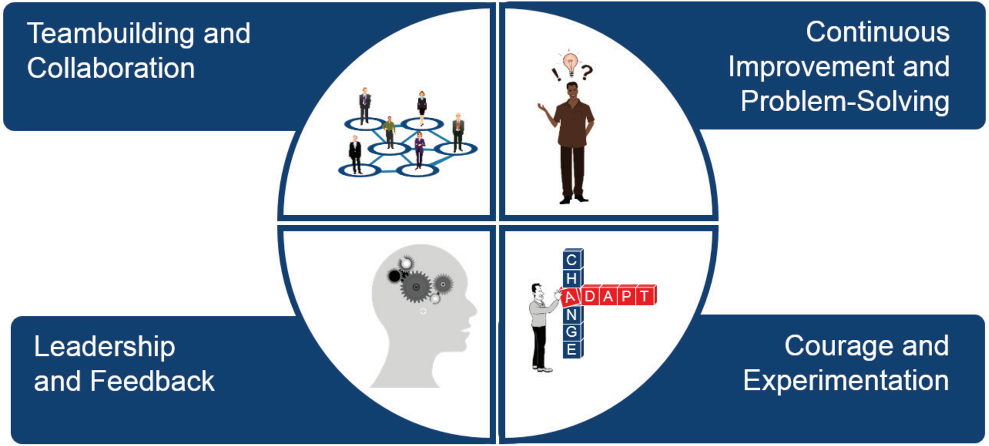
Integral Responsibility

Overall Health of the Product

A service mindset helps deliver high-quality products and services.

Culture Change: Essential for Successful DevOps

A team cannot become a DevOps team just by using a set of tools. They need to ensure that tools support DevOps requirements and workflows. A culture that supports the ideals of the DevOps movement is crucial. However, some of the cultural elements that can help you develop an effective and successful DevOps culture are:



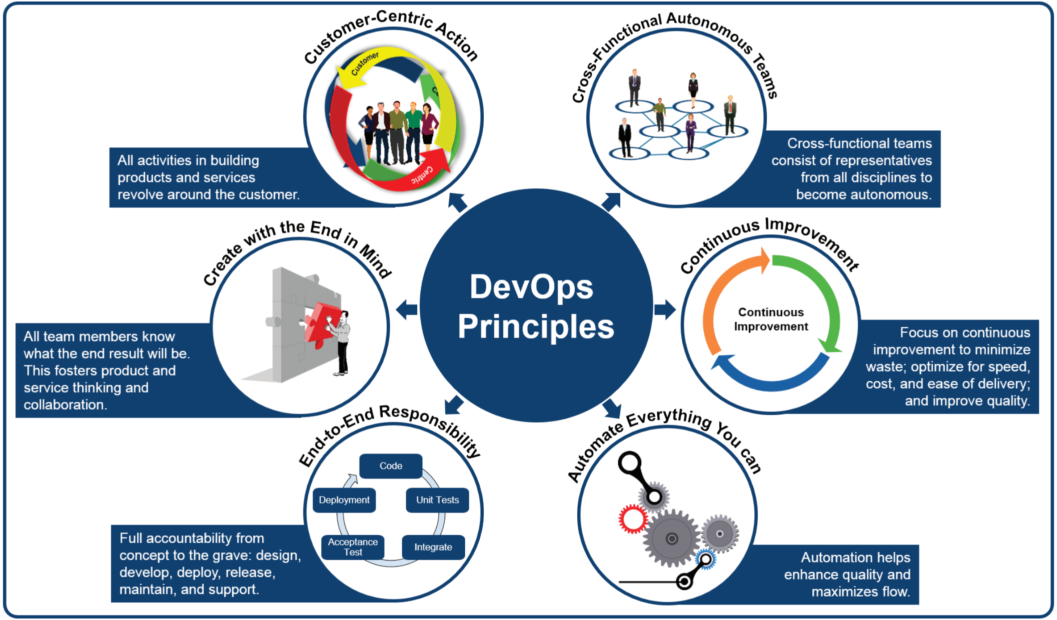
Teambuilding and Collaboration

Continuous Improvement and Problem-Solving

Leadership and Feedback

Courage and Experimentation

DevOps Principles



DevOps Principles

Customer-Centric Action

All activities in building products and services revolve around the customer.

Cross-Functional Autonomous Teams

Cross-functional teams consist of representatives from all disciplines to become autonomous.

Create with the End in Mind

All team members know what the end result will be. This fosters product and service thinking and collaboration.

Continuous Improvement

Focus on continuous improvement to minimize waste; optimize for speed, cost, and ease of delivery; and improve quality.

End-to-End Responsibility

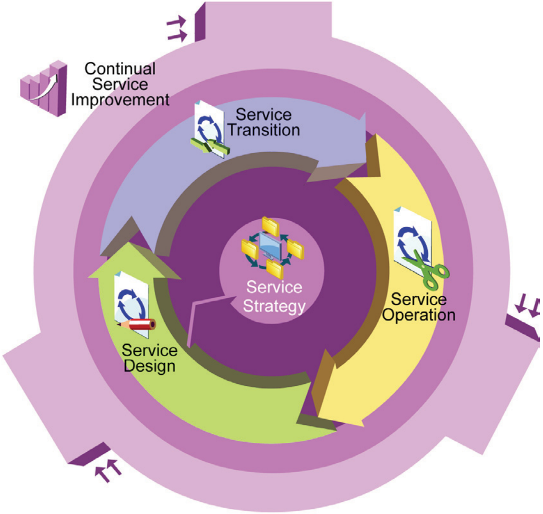
Full accountability from concept to the grave: design, develop, deploy, release, maintain, and support.

Automate Everything You can

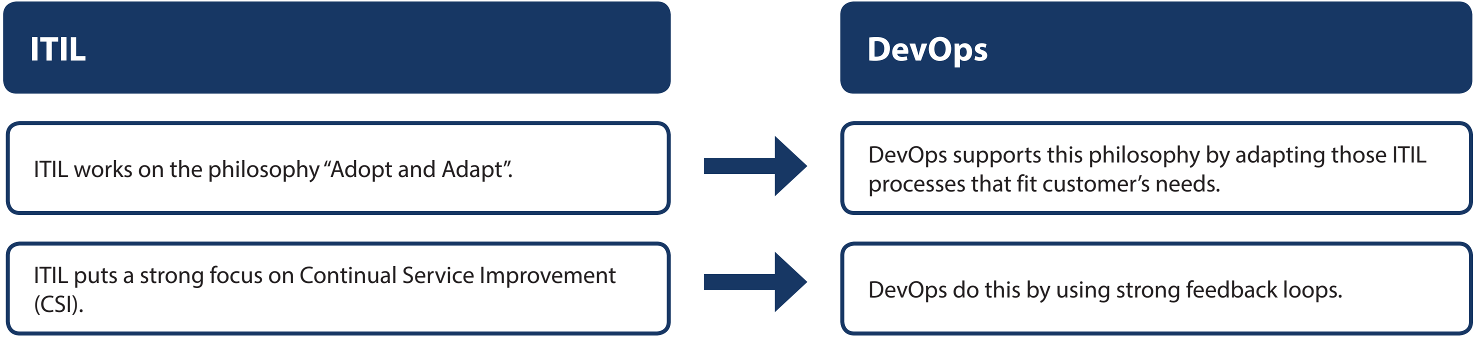
Automation helps enhance quality and maximizes flow.

ITIL

The IT Infrastructure Library (ITIL) framework provides guidance on the processes and functions that help implement quality IT services that include IT infrastructure and business applications.



ITIL and DevOps



ITIL

DevOps

ITIL works on the philosophy “Adopt and Adapt”.

DevOps supports this philosophy by adapting those ITIL processes that fit customer’s needs.

ITIL puts a strong focus on Continual Service Improvement (CSI).

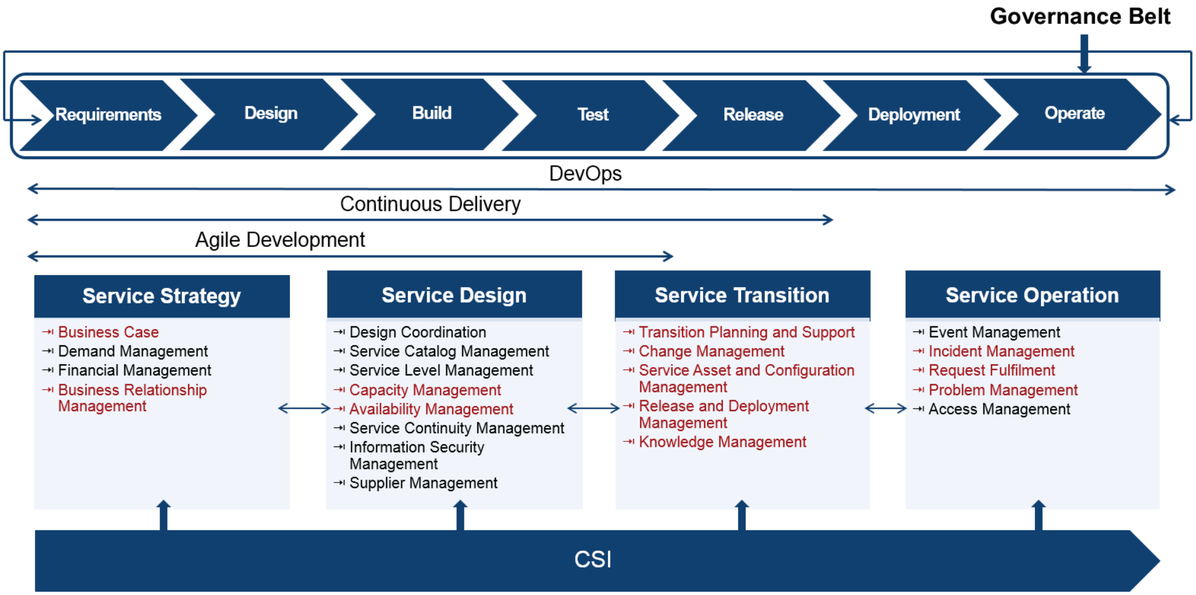
DevOps do this by using strong feedback loops.

ITIL and DevOps: Similarities and Differences

ITIL	DevOps
It is a framework that aims at aligning IT with business.	A philosophy that aims at IT and business teams working together.
It follows a phased approach, such as SDLC.	It follows an iterative approach.
It focuses on process implementation and improvement.	It focuses on automating everything that is possible.
It appreciates defining roles to maintain good practices across organizations.	No defined roles, organizations can define the roles as per their needs.
People involvement depends on the phase, the service is in.	All teams participate throughout the service/product lifecycle.

Best of Both Worlds: DevOps and ITSM

Many of the DevOps concepts, methods, and tools can be applied to the IT Service Lifecycle. In the same way, many of the ITIL processes can be used by the DevOps teams.



Governance Belt

Requirements → Design → Build → Test → Release → Deployment → Operate

DevOps

Continuous Delivery

Agile Development

Service Strategy

- Business Case
- Demand Management
- Financial Management
- Business Relationship Management

Service Design

- Design Coordination
- Service Catalog Management
- Service Level Management
- Capacity Management
- Availability Management
- Service Continuity Management
- Information Security Management
- Supplier Management

Service Transition

- Transition Planning and Support
- Change Management
- Service Asset and Configuration Management
- Release and Deployment Management
- Knowledge Management

Service Operation

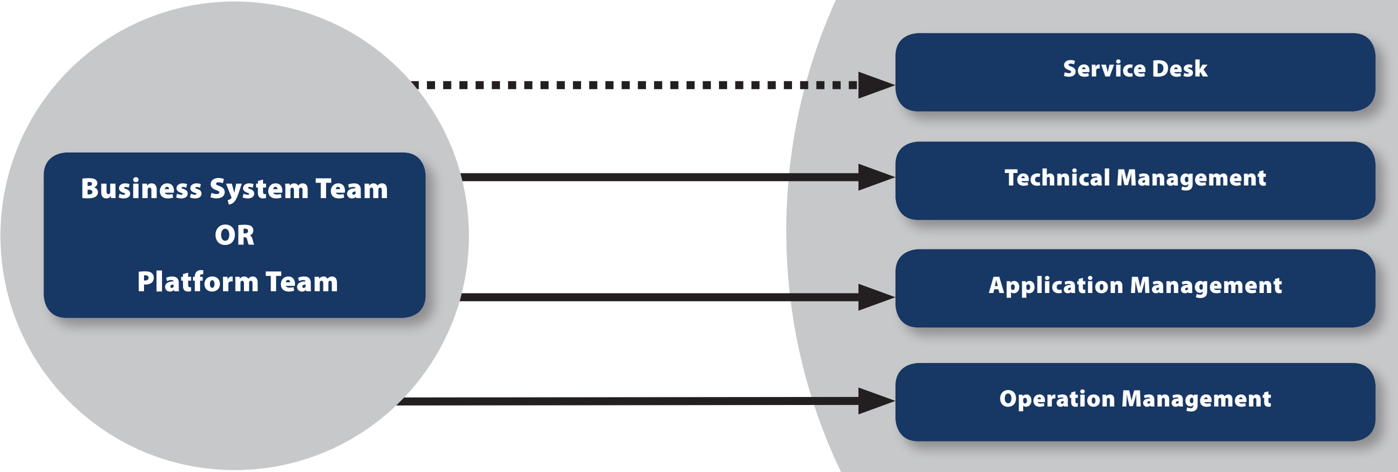
- Event Management
- Incident Management
- Request Fulfilment
- Problem Management
- Access Management

CSI

🟥 → ITSM processes used in DevOps

Integration of ITIL Functions into Business System Team / Platform Team

The roles described in ITIL still exist, although the titles might change, or the work is distributed among members. The biggest change is how teams are organized. Who sits together?



Service Strategy

It is up to the ITSM organization how they deliver and support the changing needs of the customer.

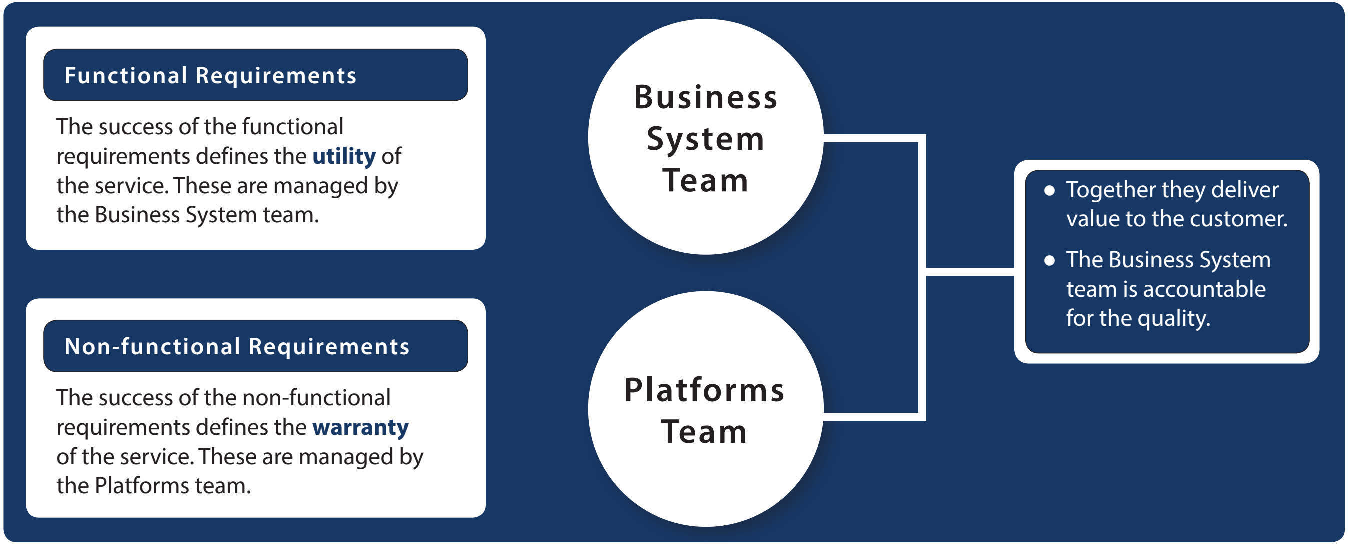
What customers do they have?

What needs do the customers have?

What are the requirements from the business?

- Number of releases per day / month / year
- Speed of change
- Number of changes per year (change slots)
- Changing requirements (during development process)
- Quality (always everywhere)

Ensuring Value Delivery



Functional Requirements

The success of the functional requirements defines the **utility** of the service. These are managed by the Business System team.

Non-functional Requirements

The success of the non-functional requirements defines the **warranty** of the service. These are managed by the Platforms team.

Business System Team

Platforms Team

- Together they deliver value to the customer.
- The Business System team is accountable for the quality.

Service Design

Operations are part of the Business System team.

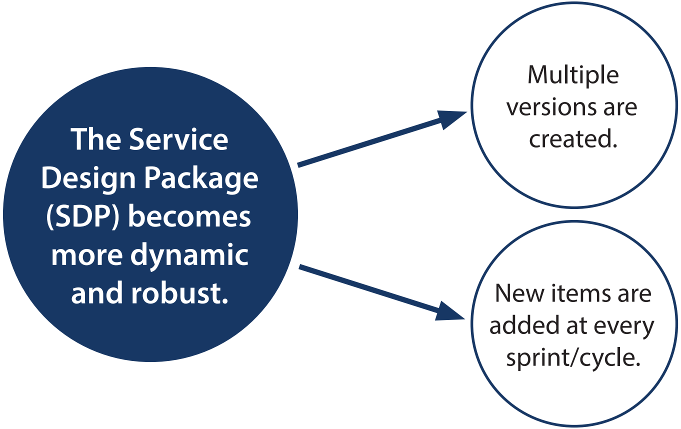
The team is responsible for designing, building, and running the service.

Both functional and non-functional requirements are considered in the design phase.

The team is responsible for the entire lifecycle of the product/service.

The pain of errors is felt by all team members.

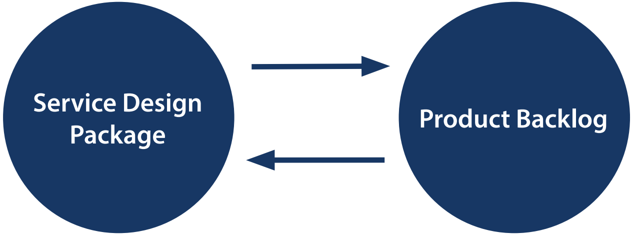
Impact on the Service Design Package



The Service Design Package (SDP) becomes more dynamic and robust.

Multiple versions are created.

New items are added at every sprint/cycle.



Service Design Package

Product Backlog

The SDP is the core documentation of the Service Design phase. Many features can be distilled from the SDP and placed on the Product Backlog. The SDP can also be completed with items from the Product Backlog. In ideal cases, the two can be merged.

Service Transition

You build it you run it.

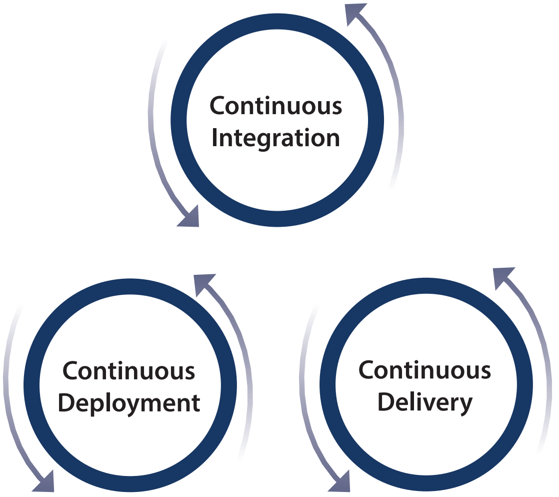
The team is responsible for the development, testing, deployment, and delivery.

No silos of competences

The team has a mix of people and competencies to make this possible.

Impact on frequency, size, type, risks of changes, and releases will be different.

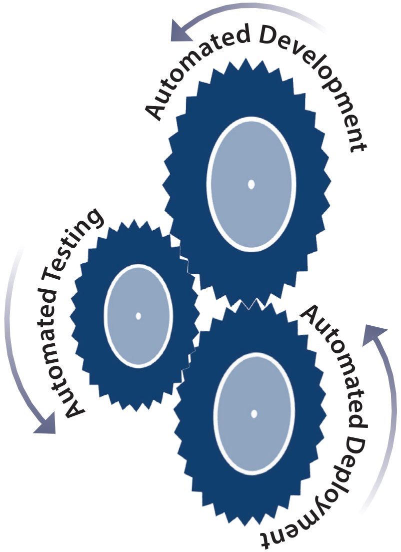
Automation of Transition



Continuous Integration

Continuous Deployment

Continuous Delivery



Automated Development

Automated Testing

Automated Deployment

Standardizing changes and releases by using models.

DevOps for Service Transition

Change Management

Traditional IT organizations are deep into bureaucratic headwaters. Changes take a long cycle of approval (Change Advisory Board (CAB)) before getting to Development teams.

DevOps busts bureaucracy by improving change approvals quickly.

DevOps integrated with ITIL helps in linking change requests from incidents and problems.

Automated backlog for developer (incident→problem→change→deployment→incident/problem closure) helps managing changes faster.

Service Operation

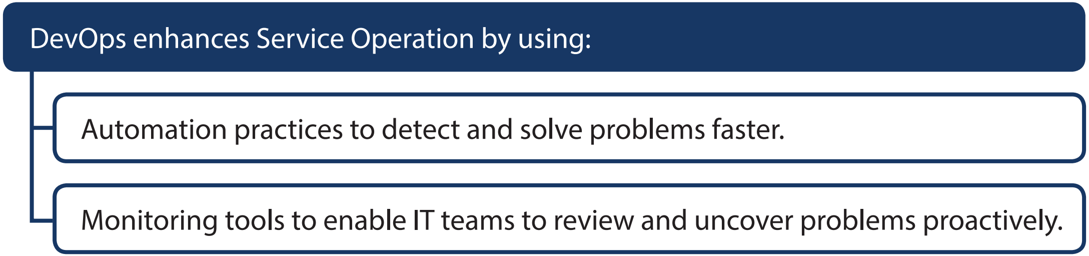
- Most operational tasks are executed by the Business System team.
- All on board in case of emergency or during peak hours.
- All required competences to run the operation should be presented to the Business System team.

+ Ability to resolve incidents much quicker

+ Ability to identify causes much better

The operation needs to be prioritized, if required, to keep delivering value to the customer.

DevOps for Service Operation

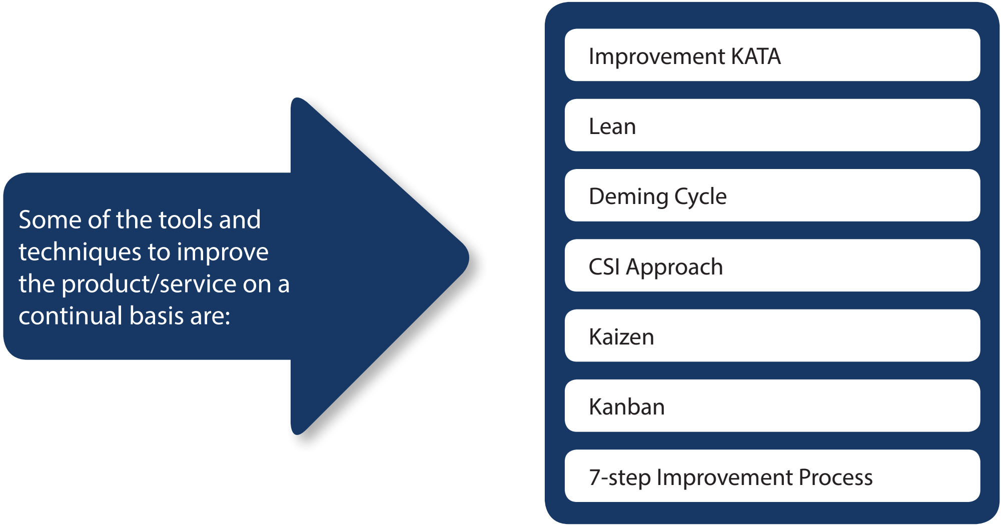


DevOps enhances Service Operation by using:

- Automation practices to detect and solve problems faster.
- Monitoring tools to enable IT teams to review and uncover problems proactively.

CSI

Merging methodologies to achieve continuous improvement in processes and culture.



Some of the tools and techniques to improve the product/service on a continual basis are:

- Improvement KATA
- Lean
- Deming Cycle
- CSI Approach
- Kaizen
- Kanban
- 7-step Improvement Process

CSI: What to improve?

Application:

DevOps focuses on feedback mechanism and customer advocacy to improve the application's usage.

Environment:

DevOps uses CSI approaches to proactively and reactively improve the environment.

Process:

- You can use DevOps to tailor Lean approaches for application delivery.
- DevOps applies Agile development practices to end-to-end application delivery, from ideation to production.

CSI: Where to start?

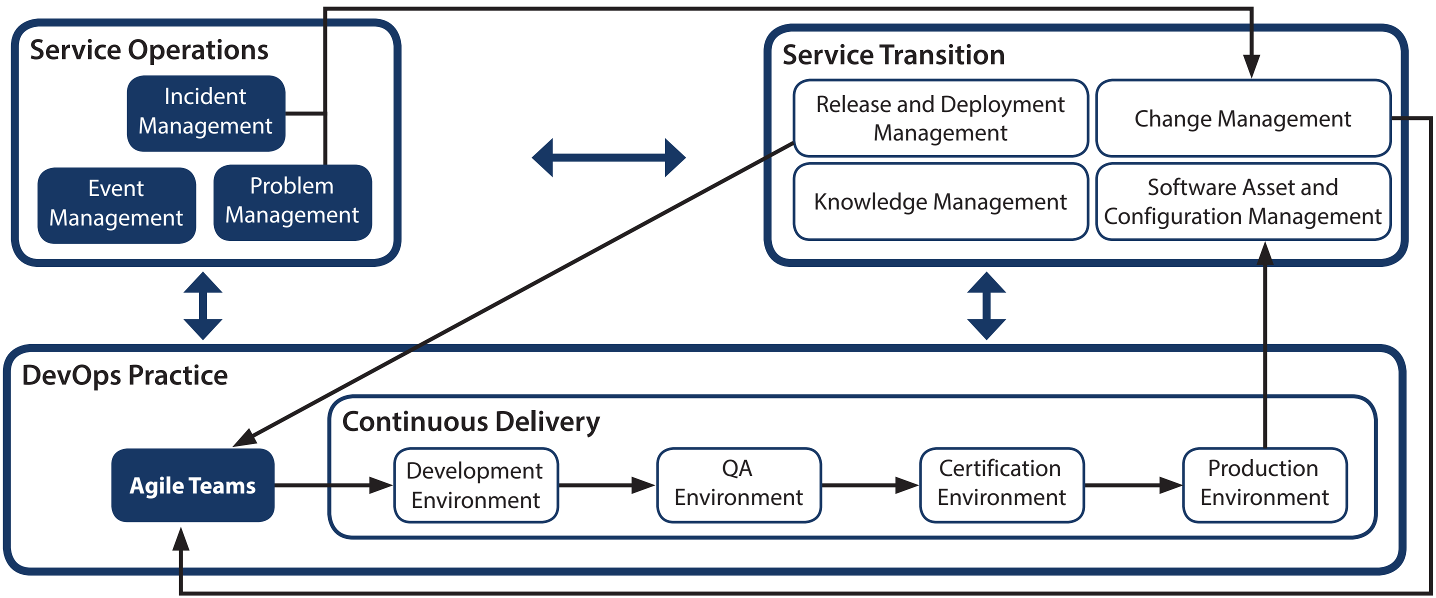
Identify inefficiencies (fat) by using Lean Value Stream Mapping

Define delivery pipeline (new project, requirements, changes to code running in production)

How efficiently this pipeline operates?

What are the waste and the bottleneck?

DevOps Streamlining Services



Key Terminology

TERM	DESCRIPTION
Agile	Agile is a time-boxed and iterative approach to software delivery. It aims to build software incrementally from the start of the project.
Autonomous	It means having the freedom to act independently.
Business System Team	Business System teams manage end users, services, and products.
Continuous Delivery	Defined by Jez Humble - “Continuous Delivery is about putting the release schedule in the hands of the business, not in the hands of IT. Implementing Continuous Delivery means making sure your software is always production ready throughout its entire lifecycle – that any build could potentially be released to users at the touch of a button using a fully automated process in a matter of seconds or minutes”.
Continuous Deployment	“Continuous Deployment is subtly different to Continuous Delivery in that release are automatically pushed into production when all tests pass. In Continuous Delivery, release is a human decision.” Dave Farley

Key Terminology (Contd.)

TERM	DESCRIPTION
Continuous Integration	Continuous Integration (CI) is the practice, in software engineering, of merging all developer working copies to a shared mainline several times a day. (Wikipedia, March 2016) “Continuous Integration usually refers to integrating, building, and testing code within the development environment.” Martin Fowler.
CSI	It focuses on learning from past successes and failures to continually improve the effectiveness and efficiency of services and processes.
Culture	It refers to the characteristics of a particular set of people, who form the distinctive, social, and physiological environment of an organization.
Lean	Lean is a term introduced by the Research team of Toyota to describe its business. The primary idea of Lean is to deliver maximum customer value with minimum waste of resources.

Key Terminology (Contd.)

TERM	DESCRIPTION
Platform Team	Platform teams manage platform products used by Business System teams. They do not take over the responsibility of the service offered by Business System teams. They offer platform (or Infrastructure) services through self-service and automation to make the Business System teams work more effectively.
Product Backlog	Scrum Term - A continuously evolving and ordered list of requirements and topics, required to make sure optimal product value is achieved. The Product backlog is the one single source of truth for modifications to the product. One list to rule them all.
Value	It is the value of a product or service in the eyes of the customer.

Interesting References

- Spotify Tribes Part I and Part II, *https://www.youtube.com/watch?v=Mpsn3WaI_4k*
- Introducing DevOps where ITIL Rules, *https://www.upguard.com/blog/introducing-devops-itil-rules-enterprise-2*
- DASA White Paper

www.ITpreneurs.com