Foundational Terminology and Concepts

Module 1.2

DevOps

- DevOps is a cultural movement that changes how individuals think about their work, values the diversity of work done, supports intentional processes that accelerate the rate by which businesses realize value, and measures the effect of social and technical change.
- It is a way of thinking and a way of working that enables individuals and organizations to develop and maintain sustainable work practices.
- It is a cultural framework for sharing stories and developing empathy, enabling people and teams to practice their crafts in effective and lasting ways.
- DevOps is about finding ways to adapt and innovate social structure, culture, and technology together in order to work more effectively.

DevOps

Defining DevOps

"DevOps is about **humans**. DevOps is a set of practices and patterns that turn **human capital** into high-performance **organizational capital**."

– *John Willis*

DevOps

Another DevOps Definition

"DevOps is the emerging professional movement that advocates a **collaborative** working **relationship** between Development and IT Operations, resulting in the fast flow of planned work (i.e., high deploy rates), while simultaneously increasing the **reliability**, **stability**, **resilience** and **security** of the production environment."

– *Gene Kim*

Changing Metrics: Shifting from Outputs to Outcomes

Outputs

Organization actually delivered

Example:

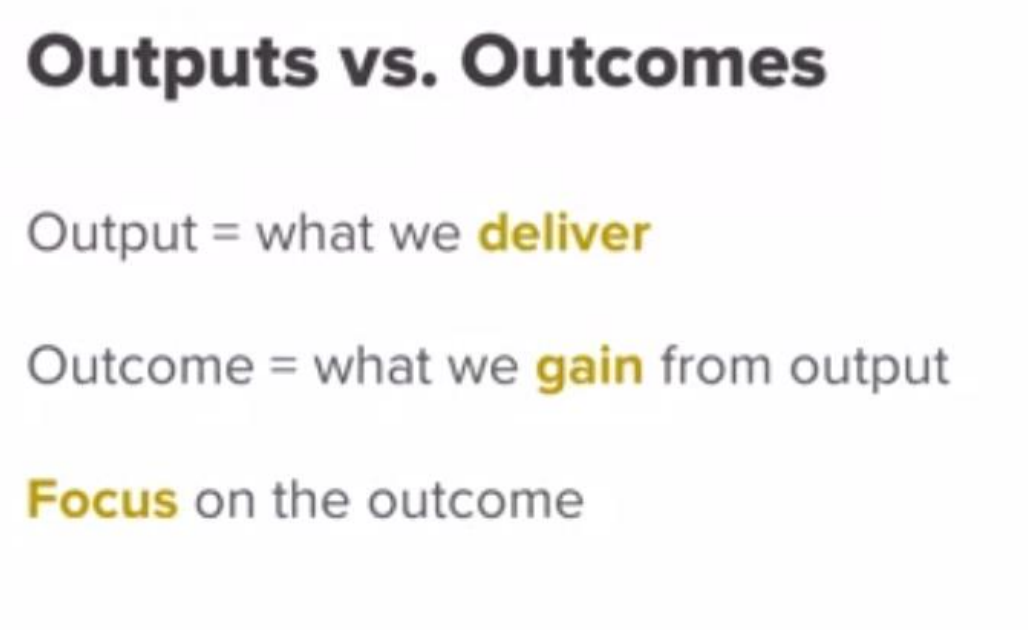
- May be new key feature of application delivered

Outcomes

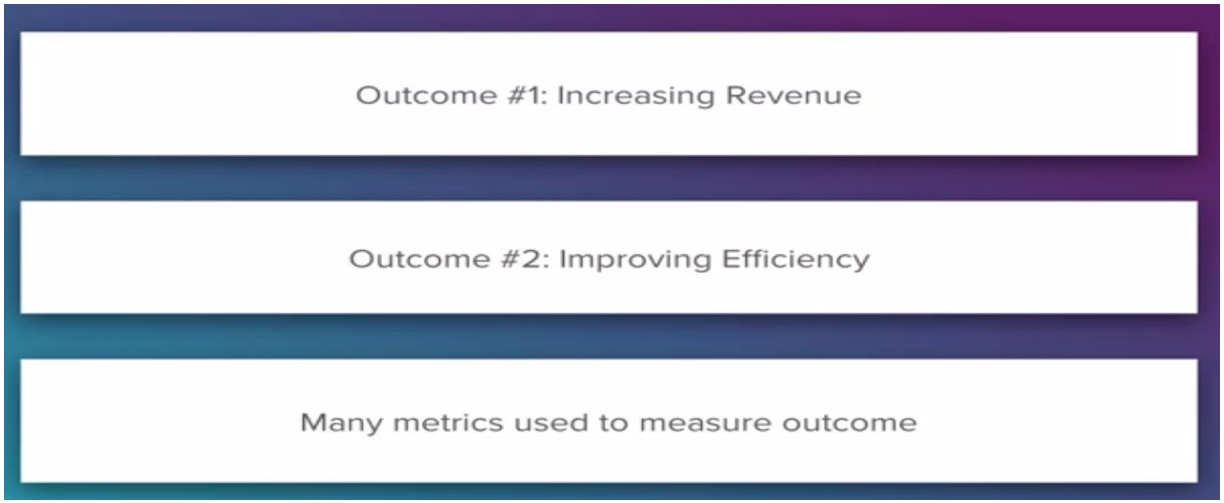
Organization gains from the outputs

Example:

- More Revenue generated from new release delivered
- Improved customer retention



Reducing costs by improving efficiency



Changing Metrics: Shifting from Outputs to Outcomes

Deployment Pain: Example

Release to production was extremely high deployment pain = **not sustainable**

Team focused on strategies to alleviate production deployment

Focused on **automation** and breaking releases to smaller batches

Change Failure Rate

Change sent to production failed

High-performing organizations have **low failure rate** = 0% - 15%

Focus on deploying **smaller changes** more frequently

Make "roll back" easy and **leverage feature flags**

Lead Time

Length of time it takes identified change to be integrated and in customer's hands

High-performing organizations = lead time is **less than an hour**

Mean Time to Restore Service (MTTR)

Length of time it takes to identify issue, fix the problem, and restore service (average)

High-performing organizations MTTR = **less than an hour**

Changing Metrics: Shifting from Outputs to Outcomes

Employee Net Promoter Score (eNPS)

"How likely you would be to refer someone to your team?"

"How likely would you be to refer someone to the overall organization?"

Culture is extremely important within a team and organization

Unplanned Work

Key indicator of organizational performance

Frequent disruption keeps from delivering project on time

Important to see how much is going on in organization

Deployment Frequency

High-performing organizations deploy **on demand** and multiple deploys per day

Directly correlated with deployment pain

Deployment Frequency: Example

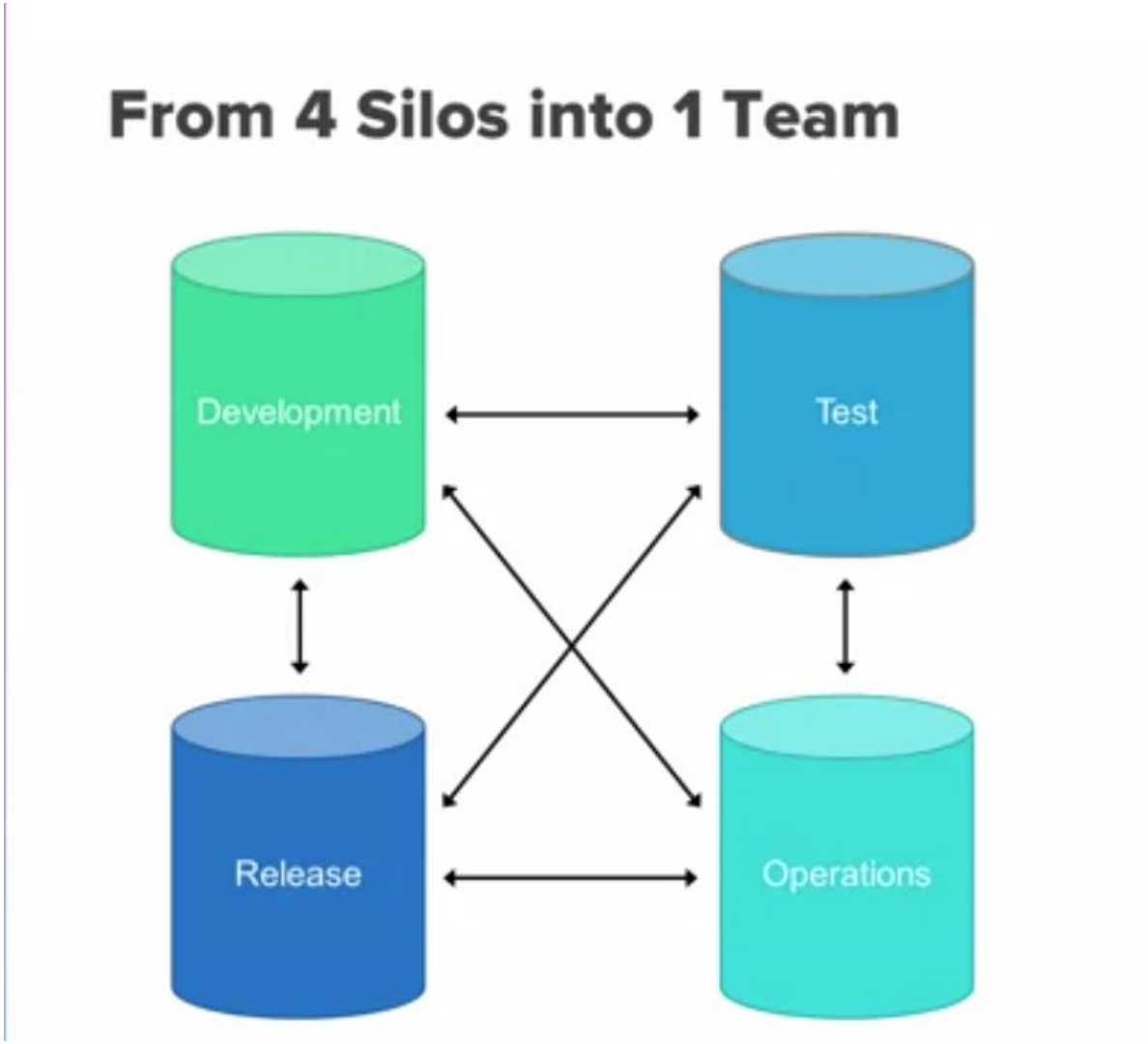
Migrating to the cloud

Low-risk and great learning experience

Validated changes before taking them live

Test, **learn**, and **respond** to market needs

DevOps


From 4 Silos into 1 Team
Development
Test
Release
Operations

Question?

- Which one of the following statements about DevOps is incorrect?
  - DevOps is suitable for brownfield software products and services
  - DevOps is suitable for greenfield software products and services
  - DevOps is only suitable for start-up companies
  - Some of the most exemplary DevOps initiatives started in companies with giant and mature IT organizations

Folk Model

Folk models are **mental models that are not necessarily accurate in the real world**, thus leading to erroneous decision making, but are shared among similar members of a culture.

- A cultural model
  - A cognitive schema that is intersubjective
  - shared by a social group.
  - A cognitive schema typically consists of a small number of conceptual objects and their relations to each other.

Folk Model

- Intersubjective Sharing
  - Everybody in the group knows the schema
  - And everybody knows that everyone
  - knows the schema
  - And everybody knows that everyone
  - knows that everyone knows the schema.

Folk Model

- Using schemas to understand the world
  - **Instantiation** of the schema means to connect specific things and events to the conceptual elements (slots or roles) in the schema.
  - The general “buying” schema has these roles
    - Purchaser, seller, merchandise, price, sale, money
    - and events linking these.
    - An instantiation of the “buying” schema assigns
    - particular people and events to the parts of the
    - schema.
    - E.G., John bought a hat at the bookstore.

DevOps as Folk Model

DevOps has become a folk model

- A term used with different intent that leads to miscommunication and misunderstanding.
- Abstraction for more concrete ideas and often substituted, being easier to understand than the concept ultimately being discussed.
- An example of this is the term situational awareness, which is often used as a stand-in for more specific ideas like perception and short-term memory.
- Folk models are not necessarily bad. They become problematic when different groups use the same term with different intent.

DevOps as Folk Model

The model of DevOps is not often taught, but instead learned and pieced together from observations and pieces of information, with the blanks filled in by common sense.

The result is that the model may lead to wrong conclusions and bad decisions, and at best, inefficiency in putting the concept into use from the point of view of the actual model.

Based on the observations from different projects, this is a common phenomenon with DevOps and makes related communication much more difficult.

Usually people use it as a term to refer to CI/CD, automation, testing etc., and in this sense, it has become a buzzword.

However, behind the buzzword is an actual and very useful phenomenon is Culture.

The Old View and the New View

- In an environment where humans are blamed and punished for errors, a culture of fear can build up walls that prevent clear communication and transparency.
- Contrast this with a blameless environment, where issues are addressed cooperatively and viewed as learning opportunities for individuals and the organization.
- The first environment views “human error as the cause of trouble.” This “old view” is described as a mindset in which the focus is on elimination of human error.
- The second environment views “human error as a symptom of trouble deeper in the system.” This “new view” is a mindset that sees human errors as structural rather than personal.

The Old View and the New View

Understanding and embracing the “new view” is key to understanding the devops movement. This view encourages us to share stories, as everything is a learning opportunity.

Shared stories:

- lead to increased transparency and trust within a team;
- instruct our co-workers in how to avoid a costly mistake without having to directly experience it; and
- increase the time spent on solving new problems, allowing for more innovation.

When these stories are shared throughout the industry, we impact the industry as a whole, creating new opportunities, knowledge, and shared understanding.

Question?

In a DevOps organization which one of the following elements does not directly contribute to your value stream?

- DevOps team
- Stakeholders of downstream work centers
- Errors, incidents and fixes
- Clients

The DevOps Compact

The heart of devops starts with people working not only as groups but as teams with a desire for mutual understanding.

This can be described as a compact that teams will work together, communicate their intentions and the issues that they run into, and dynamically adjust in order to work toward their shared organizational goals.

The principles of this compact that make it work include:

- Shared, clearly defined goals
- Ongoing communication
- Dynamic adjustment and repairs of understanding.

Example of the The DevOps Compact

- Two employees work on separate teams at Sparkle Corp.
- The General is a senior developer with a number of different experiences in her background and has worked at Sparkle Corp for two years. George is an operations engineer with some experience and is relatively new to Sparkle Corp.
- Their two teams support the global community of people that depend on the Sparkle Corp website for their creative endeavors. Their shared goal is to implement a new feature that will increase the value to end users, hopefully without impacting the site.
- As the one with more experience at the company, the General will be really clear with George about the expectations, values, and processes in place at Sparkle Corp. In turn, George will be really clear with the General about when he needs help or doesn't understand part of the process.
- Both the General and George will check in with each other's work before proceeding to next steps—an example of the trust-but-verify model, as described with the climbing process.

The General and George have a shared understanding of their goals:

- Implementing a new feature that increases the value to Sparkle Corp customers
- Maintaining safety and trust in their communication with each other

The DevOps Compact

With the shared understanding that everyone is still a part of the compact, actions turn into repairs.

- We repair our misunderstandings about who would be working on a particular feature or when something would get done.
- We repair bugs that affect our understanding of how the software is supposed to behave.
- We repair processes and their documentation when things don’t go the way we expect in production.

Question?

- Goal of DevOps is
  - Establish an environment to release more reliable applications faster
  - Esbatalish an environment where the release of applications is valued more than its quality
  - Establish an environment where application development performs all the operation tasks
  - Establish an environment where change management does not control application releases

Question?

Which DevOps principle focuses on product and service thinking?

A) Customer-centric action

B) Continuous Improvement

C) Create with the end in mind

D) Automate everything you can

1.3 Methodologies

Software Development Methodologies

The process of splitting up development work, usually into distinct phases, is known as a software development methodology.

These different phases of work may include:

- Specification of deliverables or Artifacts
- Development and verification of the code with respect to the specification
- Deployment of the code to its final customers or production environment

Software Development Methodologies

Waterfall

The waterfall methodology or model is a project management process with an emphasis on a sequential progression from one stage of the process to the next. The original stages were requirements specification, design, implementation, integration, testing, installation, and maintenance, and progress was visualized as flowing from one stage to another.

Agile

Agile is the name given to a group of software development methodologies that are designed to be more lightweight and flexible than previous methods such as waterfall.

The Agile Manifesto, written in 2001 outlines its main principles as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- individuals and interactions over processes and tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following a plan

Agile methodologies include processes such as Scrum and other methods that place a heavy emphasis on collaboration, flexibility, and the end result of working software.

Software Development Methodologies

Is DevOps Just Agile?

DevOps shares many characteristics with the Agile movement, especially with the focus on individuals, interactions, and collaboration.

You might wonder if DevOps is just “rebranded” Agile.

While DevOps has certainly grown around Agile methodology, it is a separate cultural movement steeped in the history of the computing industry with a broad reach that includes more than just developers.

DevOps adopts and extends Agile principles and applies them to the entire organization, not only the development process.

Software Development Methodologies

Scrum

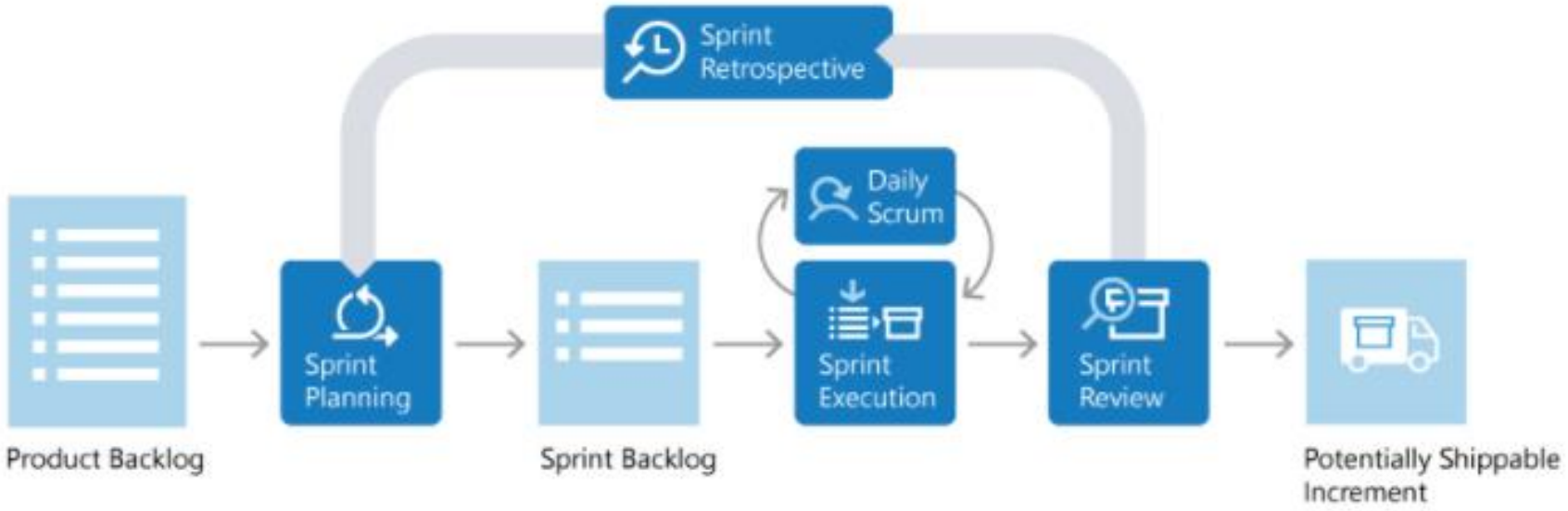
Scrum is a software development methodology that focuses on maximizing a development team's ability to quickly respond to changes in both project and customer requirements. It uses predefined development cycles called sprints, usually between one week and one month long, beginning with a sprint planning meeting to define goals and ending with a sprint review and sprint retrospective to discuss progress and any issues that arose during that sprint.

One key feature of Scrum is the daily Scrum or daily standup, a daily meeting where team members (rather rapidly) each answer three questions:

- What did I do yesterday that helped the team meet its sprint goals?
- What am I planning to do today to help the team meet those goals?
- What, if anything, do I see that is blocking either me or the team from reaching their goals?

Software Development Methodologies

Scrum



References: https://docs.microsoft.com/en-us/devops/plan/what-is-scrum

Software Development Methodologies

Scrum

These meetings, which take place in the morning in order to help people align with what they are planning to do that day and help each other with any blocking issues, are often facilitated by the Scrum master.

The Scrum master is an important role that also includes responsibilities such as helping the team self-organize and coordinate work efforts, helping remove blockers so the team will continue making progress, and involving project owners and stakeholders so there is a shared understanding of what "done" means and what progress is being made.

The principles of Scrum are often seen applied less formally in many software development practices today.

Question?

Which DevOps principle focuses on product and service thinking?

A) Customer-centric action

B) Continuous Improvement

C) Create with the end in mind

D) Automate everything you can

Operations Methodologies

Similar to how software development methodologies split up software development work into different phases or otherwise try to bring more order to those processes, IT or operations work can be split up or organized as well.

ITIL: Information Technology Infrastructure Library

- A set of practices defined for managing IT services.
- It is published as a series of five volumes that describe its processes, procedures, tasks, and checklists, and is used to demonstrate compliance as well as measure improvement toward that end.
- The five core sections in the most recent (2011) version being service strategy, service design, service transition, service operation, and continual service improvement.

Operations Methodologies

ITIL

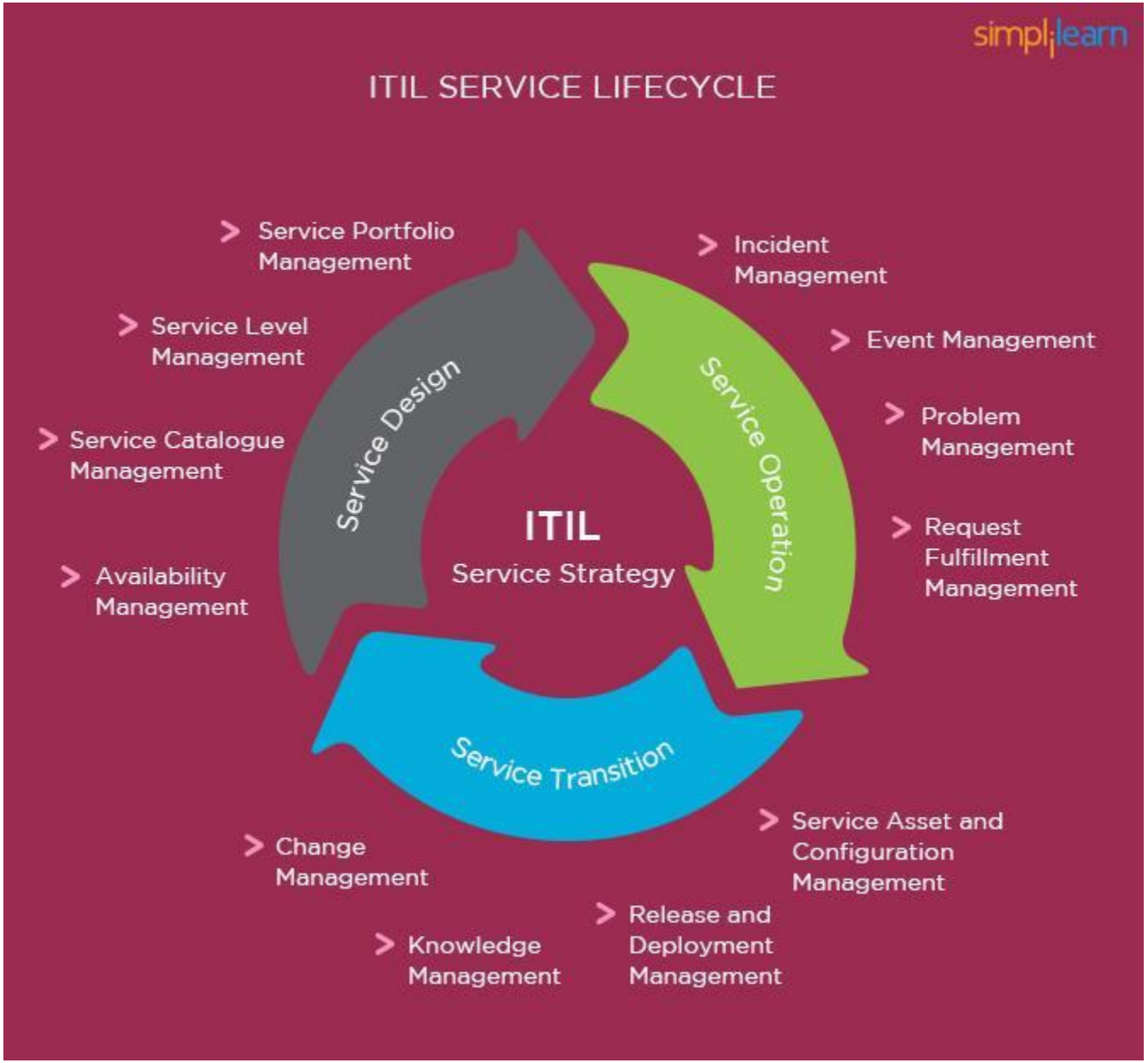
ITIL stands for Information Technology Infrastructure Library. It is a set of best practices for delivering IT services—it standardizes the selection, planning, delivery, and support of IT services to maximize efficiency and maintain predictable levels of service.

The five core sections are:

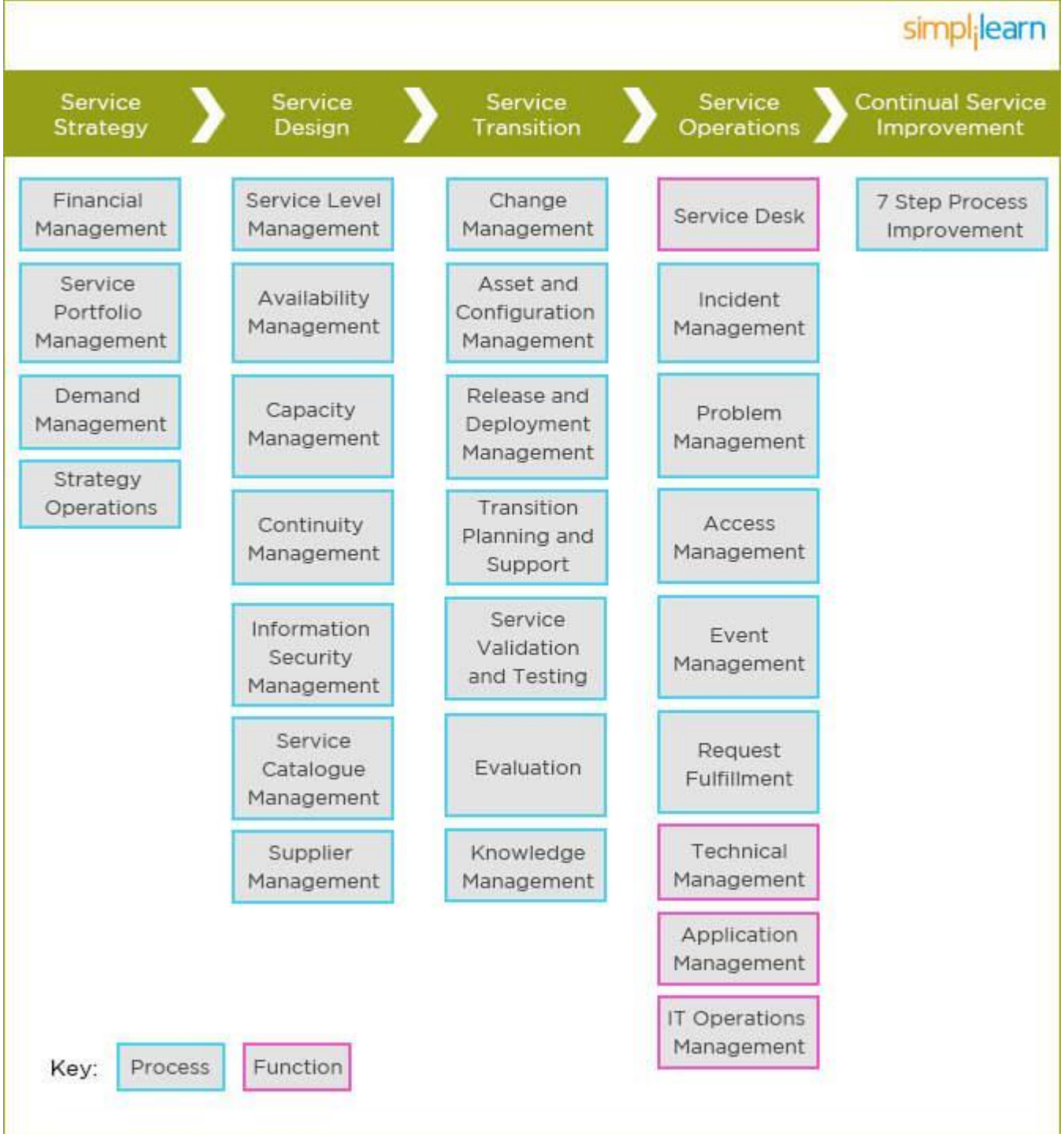
- **Service Strategy.** Describes business goals and customer requirements and how to align objectives of both entities.
- **Service Design.** Outlines practices for the production of IT policies, architectures and documentation.
- **Service Transition.** Advises on change management and release practices, and also guides admins through environmental interruptions and changes.
- **Service Operation.** Offers ways to manage IT services on a daily, monthly and yearly basis.
- **Continual Service Improvement.** Covers how to introduce improvements and policy updates within the ITIL process framework.

Operations Methodologies

ITIL



ITIL


simplilearn
Service Strategy
Service Design
Service Transition
Service Operations
Continual Service Improvement
Financial Management
Service Portfolio Management
Demand Management
Strategy Operations
Service Level Management
Availability Management
Capacity Management
Continuity Management
Information Security Management
Service Catalogue Management
Supplier Management
Change Management
Asset and Configuration Management
Release and Deployment Management
Transition Planning and Support
Service Validation and Testing
Evaluation
Knowledge Management
Service Desk
Incident Management
Problem Management
Access Management
Event Management
Request Fulfillment
Technical Management
Application Management
IT Operations Management
7 Step Process Improvement
Key:
Process
Function

Operations Methodologies

ITIL Framework

1. Service Strategy

The purpose of Service Strategy is to provide a strategy for the service lifecycle. The strategy should be in sync with business objectives. The utility and warranty of this component are designed to ensure that the service is fit for purpose and fit for use, respectively. Ensuring this is important, as these two components are what add value to the delivery of services to customers.

2. Service Design

The Service Design lifecycle phase is about the design of services and all supporting elements for introduction into the live environment. The "Four Ps of Service Design" represent areas that should be taken into consideration when designing a service. They are:

- People
  Human resources and organizational structures required to support the service
- Processes
  Service Management processes required to support the service
- Products
  Technology and other infrastructure required to support the service
- Partners
  Third parties that offer additional support required to support the service

Operations Methodologies

ITIL Framework

3. Service Transition

The objective of the Service Transition process is to build and deploy IT services, making sure that changes to services and Service Management processes are carried out in a coordinated way.

- In this phase of the lifecycle, the design is built, tested, and moved into production to enable the business customer to achieve the desired value. This phase addresses managing changes: controlling the assets and configuration items (the underlying components, such as hardware and software) associated with the new and changed systems, service validation, testing and transition planning to ensure that users, support personnel, and the production environment have been prepared for the release to production.

Operations Methodologies

ITIL Framework

4. Service Operations

- This stage focuses on meeting end-users' expectations while balancing costs and discovering any potential problems. The Service Operations process includes fulfilling user requests, resolving service failures, fixing problems, and carrying out routine operational tasks. This is the only category of the five that has functions as well as processes.

5. Continual Service Improvement (CSI)

- The objective of this stage is to use methods from quality management to learn from past successes and failures. It aims to continually improve the effectiveness and efficiency of IT processes and services in line with the concept of continual improvement adopted in ISO 2000.
- it has seven steps:
  1. Identifying improvement strategies
  2. Defining what will be measured
  3. Gathering data
  4. Processing data
  5. Analysing data
  6. Presenting and using the information drawn from the data
  7. Using the information to improve

References: https://www.simplilearn.com/itil-key-concepts-and-summary-article

Operations Methodologies

COBIT

Control Objectives for Information and Related Technology (COBIT) is an ISACA framework for governance and management of information and technology first released in 1996. A core principle of COBIT is to align business goals with IT goals.

COBIT is based on 5 principles:

- Meeting stakeholder needs;
- Covering the enterprise from end to end;
- Applying a single integrated framework;
- Enabling a holistic approach; and
- Separating governance from management.

References: http://www.free-management-ebooks.com/news/cobit/

Systems Methodologies

Some methodologies focus on thinking about systems as a whole, rather than limiting focus to more specific areas such as software development or IT operations.

Lean

Womack and Jones defined the five principles of Lean Thinking as follows:

1. Value
2. Value stream
3. Flow
4. Pull
5. Perfection

These ideas, especially the pursuit of perfection through systemic identification and elimination of waste, drove the definition of Lean as the maximization of customer value and minimization of waste.

Lean systems focus on the parts of the system that add value by eliminating waste everywhere else, whether that be overproduction of some parts, defective products that have to be rebuilt, or time spent waiting on some other part of the system.

Systems Methodologies

Lean

- Stemming from this are the concepts of Lean IT and Lean software development, which apply these same concepts to software engineering and IT operations. Waste to be eliminated in these areas can include:

  • Unnecessary software features • Communication delays • Slow application response times

  • Overbearing bureaucratic processes
- Waste in the context of Lean is the opposite of value.
- Mary Poppendieck and Thomas Poppendieck have mapped Lean manufacturing waste to software development waste as follows:

  • Partially done work • Extra features • Relearning • Unnecessary handoffs Task switching • Delays • Defects
- As with DevOps, there is no one way to do Lean software development.
- There are two main approaches to Lean:

  a focus on waste elimination through a set of tools, and

  a focus on improving the flow of work, also known as The Toyota Way.

Both approaches have the same goal, but due to the differing approaches may result in different outcomes.

Systems Methodologies

Lean

References: https://www.planettogether.com/blog/five-principles-of-lean-manufacturing

Systems Methodologies

- **Define Value** -Understand customers value in terms of their needs, what they really want, and what they are willing to pay for. There are many techniques such as interviews, surveys, and demographic information that can help you discover what exactly customers find valuable.
- **Mapping the Value Stream** - Goal is to utilize the customer's value as a reference point and locate areas that correlate with their values. Any activities and processes that do not add value to the end customer are considered wasteful.
- **Creating Flow** - After removing the waste from the value stream, the following action is to ensure that the flow of the remaining steps will run smoothly without interruption or delays.
- **Establish Pull** -hrough following the value stream and working backward through the production system, you can ensure that the products produced will be able to satisfy the needs of customers.
- **Pursue Perfection** - Waste is prevented through the achievement of the first four steps which include identifying value, mapping the value stream, creating flow, and adopting a pull system. The fifth step, pursuing perfection makes lean thinking and continuous process improvement a part of the organizational culture. All employees should attempt to strive toward perfection while delivering products based on the customer's needs.