

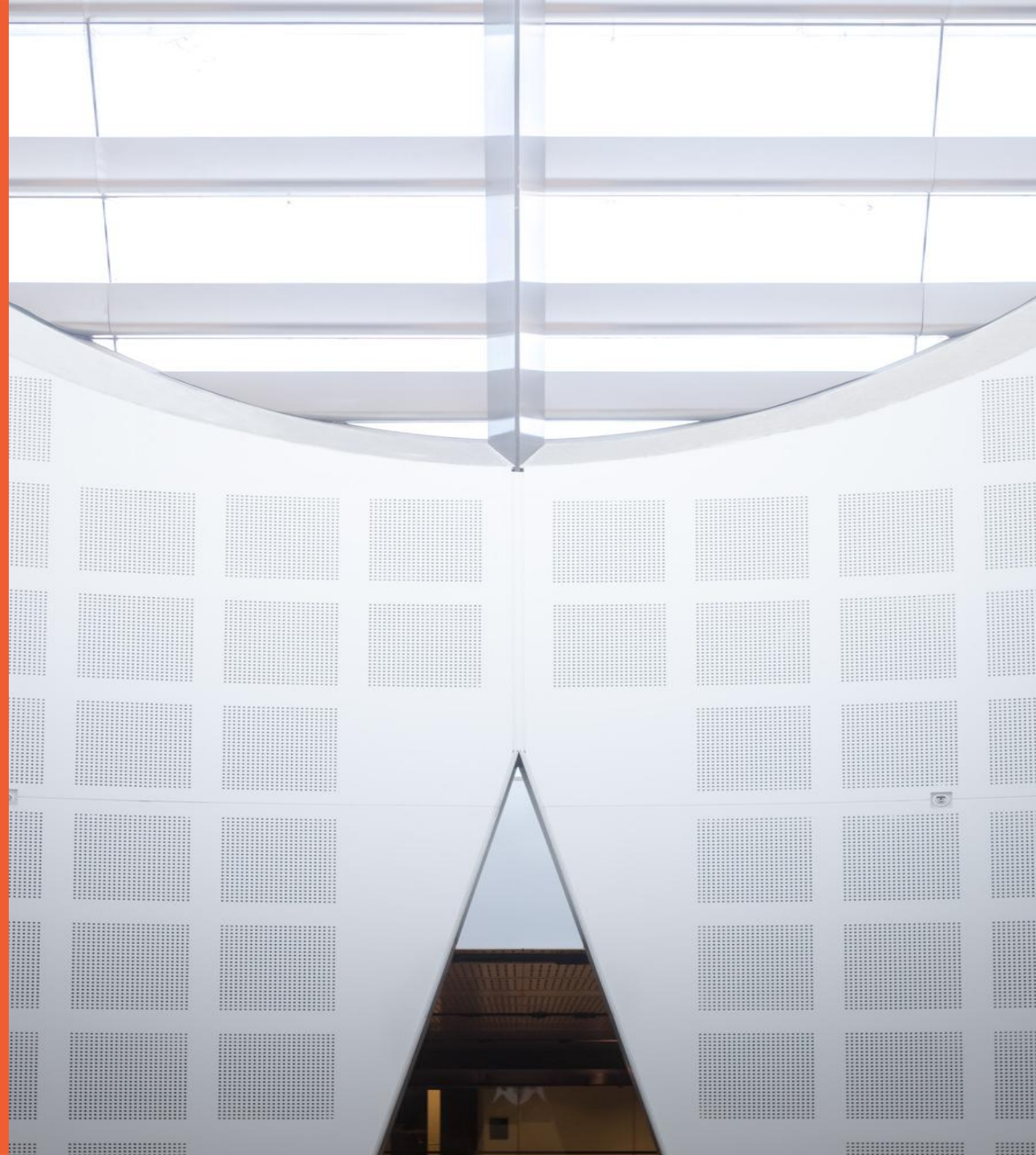
# INFO5990: Professional Practice in IT

## Week 3: IT Lifecycles and their processes

Dr Mafruha Mowrin Hossain  
School of Computer Science



THE UNIVERSITY OF  
SYDNEY



## Week 3 : IT Lifecycle and its processes

- **Topics**

- A whole of lifecycle view of IT in organisations
- An example of the Ongoing Lifecycle of a Family Project to Acquire and Operate a Smart Home System
- Integrating new components into the family's interdependent systems
- An example of an organisation's processes for integrating a new IT system

- **Learning outcomes (able to..)**

- *Identify and access descriptions of processes for managing the IT lifecycle*

# Project characteristic

## What is a project?

- A set of tasks that must be completed within a defined timeline to accomplish a specific set of goals.
- Completed by a group of people known as the project team, which is **led by a project manager** who oversees the **planning, scheduling, tracking**, and successful completion of projects.
- Can be for in-house development or outsourced development
- **Example:**  
Create a mobile app that allows customers to browse products, place orders, and schedule pickups.

## Project characteristics:

- Accomplishment with shared resources
- Cross-functional teamwork may be required
- Uncertainty and potential change during execution
- Go through the same project lifecycle
- Changes to the way the business operates
- Specific deadlines, time, and resource constraints.

# The IT lifecycle

## What is the result of a project being completed?

- The continuous delivery of the mobile app's functions to users, ensuring seamless product browsing, order placement, and pickup scheduling.
- managed by techniques such as IT service management (ITSM) & DevOps
  - Ensuring the app remains reliable, scalable, and continuously updated to improve user experience.
  - Implementing continuous improvement processes, such as real-time monitoring, automated testing, and customer feedback integration.

**Example:** The ongoing operation and maintenance of the mobile shopping app, including regular feature updates, security patches, and system optimizations.

# What does the term 'to integrate' mean?

- The meaning of **INTEGRATE** is to form, coordinate, or blend into a functioning or unified whole
- <https://www.merriam-webster.com/dictionary/integrate>

# Why is integration important?

**Integration** is essential for businesses to connect their disparate systems and processes, enable seamless data exchange, and drive innovation.

**Example:** Imagine you create a mobile app that allows customers to browse products, place orders, and schedule pickups. Integration ensures that when a customer places an order, the inventory updates automatically, the payment is processed securely, and the delivery service is notified—all working together seamlessly.

- **Improve operational efficiency**
  - **Gain a single view of the data**
  - **Enable innovation**
- **Challenges:** Integrating a new system into this complex environment is a task that demands meticulous planning and execution, as it is not a simple matter.

# Building a new system to integrate into the organisation

## Before project initiation - strategic planning for:

- value,
- costs,
- risks etc

## During the project:

- system requirements,
- Integration requirements
- design,
- build or acquire,
- customise
- Test
- Training

## Integration into the live environment

# Running an integrated system

## 1. Operation in the live environment:

- manage incidents and problems,
- report status and projections for continuity and risk,
- manage service levels
- Integrate upgrades
- Ensure integration with future new systems

## 2. At all stages: continuous feedback and improvement to align with organisational goals



# Example: The Ongoing Lifecycle of a Family Project to Acquire and Operate a Smart Home System (for Personal Use)

## Project Background

- a) A family of 2 adults (decision-makers/joint CEOs) and 2 young children (users/consumers).
- b) Living in Australia and looking for a more automated, energy-efficient, and secure home environment.

## Project Requirements

- a) Improve home security, convenience, and energy efficiency through smart home automation.

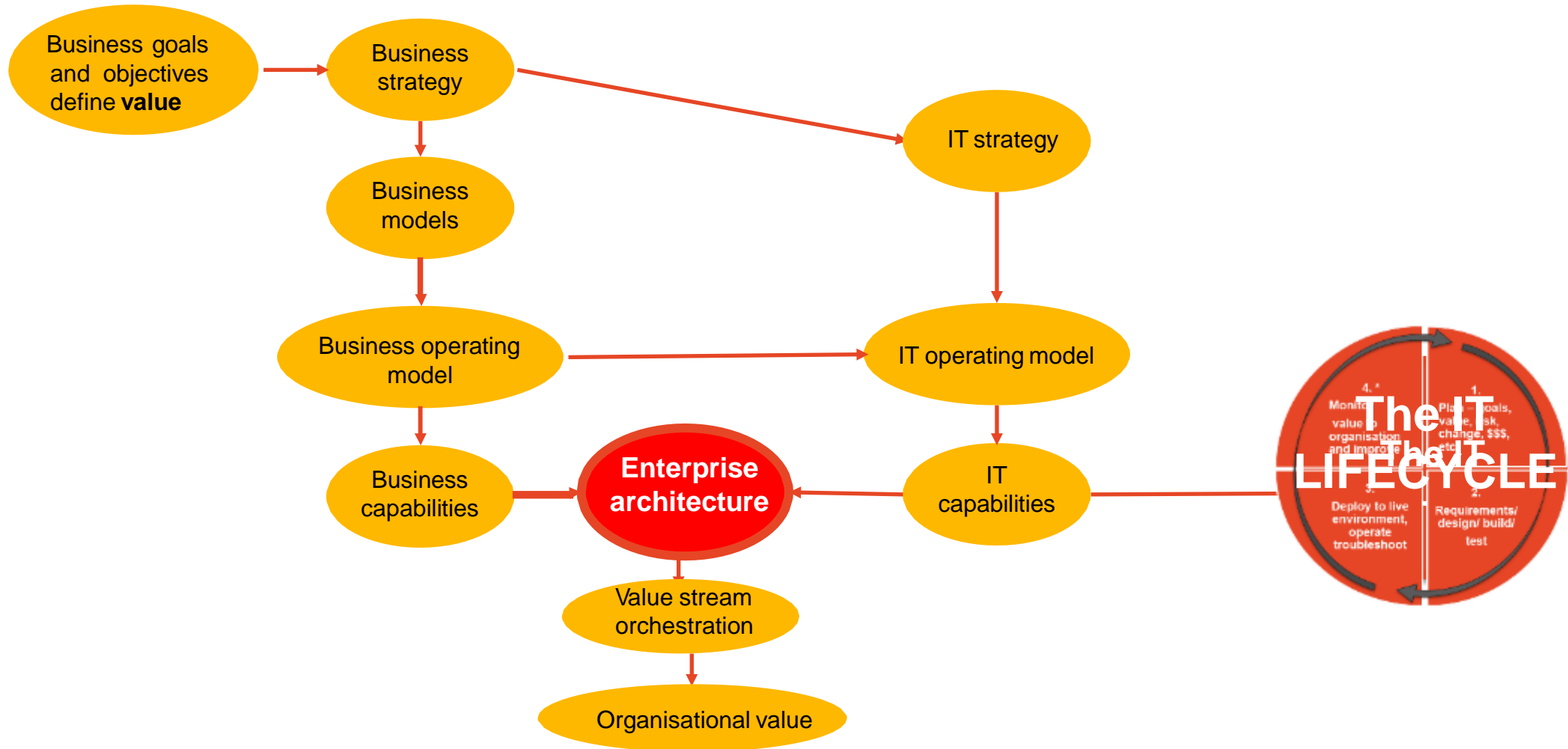
## Project tasks

- a) Research & Selection : Compare different smart home systems (Google Home, Amazon Alexa, Apple HomeKit). Select most cost-effective
- b) Purchase : Buy the most cost-effective and compatible devices.
- c) Installation : Install smart lights, security cameras, smart locks, and thermostats.

## Project completion & Ongoing Maintenance

- a) Ensure everything is working properly.
- b) Regularly check and update access permissions for family members.

# The IT lifecycle



# The IT lifecycle: Overview

## **Definition:**

*The IT lifecycle is the series of stages which an IT capability goes through from its inception to its retirement from use.*

## **Key Aspects of the IT Lifecycle:**

1. From linear to continuous IT lifecycles
2. Enterprise architecture and its role in the continuous IT lifecycle
3. IT Service management (ITSM) frameworks
4. The agile approach
5. DevOps

# **1. From Linear to Continuous IT Lifecycles**

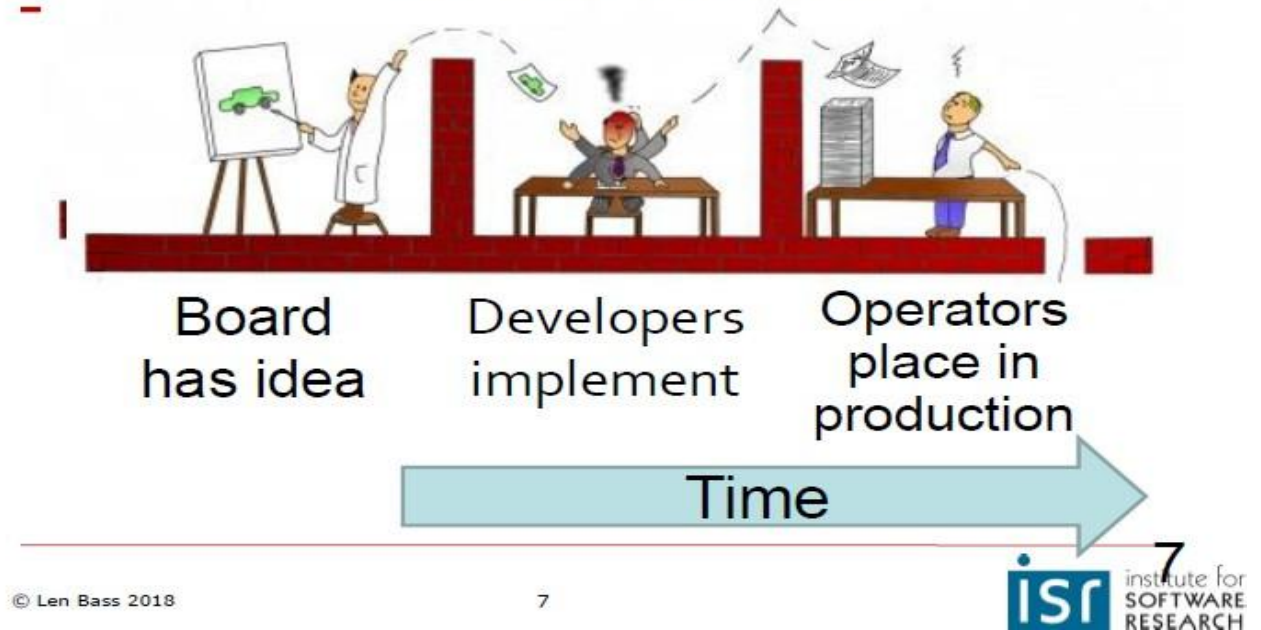
---

# Business and IT silos

- In traditional development, the development team and operations team worked in silos.
- Development teams focused on functionality, features, and non-functional requirements.
- The focus of operations teams were cost, reliability, security, risk, and manageability

Carnegie Mellon University

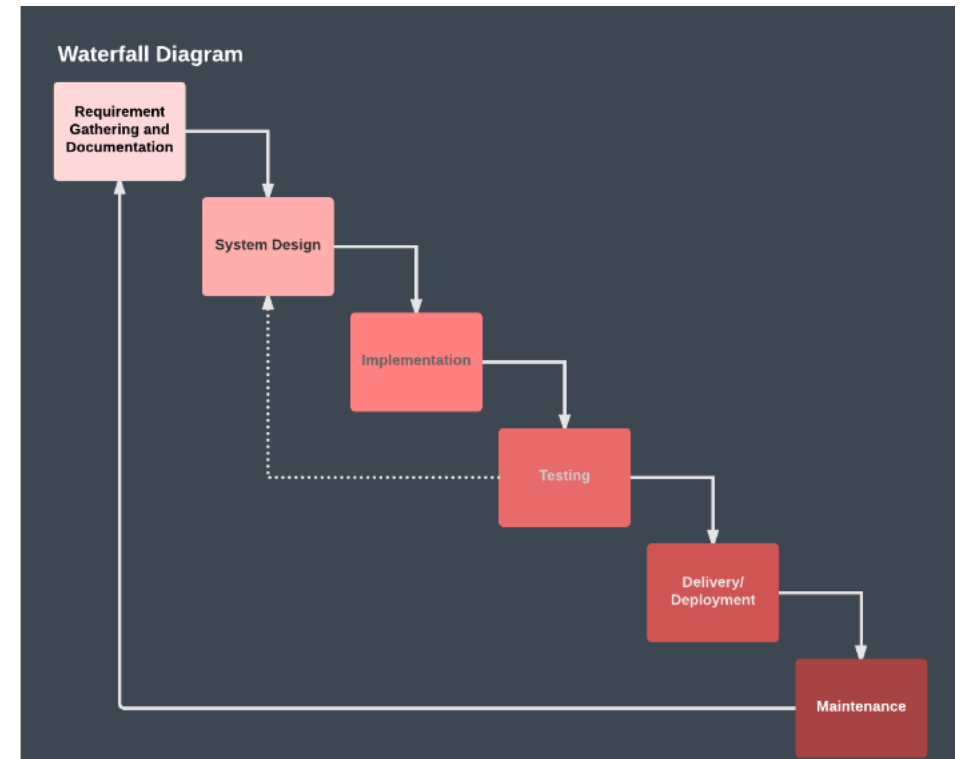
## Traditional over the wall development



Source: [Role of Enterprise Architecture in DevOps Adoption - DZone](#)

# A traditional, linear approach to the lifecycle for an IT capability, e.g. a software app.

- Traditionally, the approach to managing the lifecycle of a software application was a set of linear, sequential, stages that managed the development of the application as a project – i.e. as a temporary, time bound, endeavour undertaken to create a unique result.
- The waterfall method was typically used for this project management and was an earlier foundation of the PMBOK (Project Management body of Knowledge), and SDLC (Software Development Life Cycle) methodologies.
- Both of these methodologies have been expanded since then.

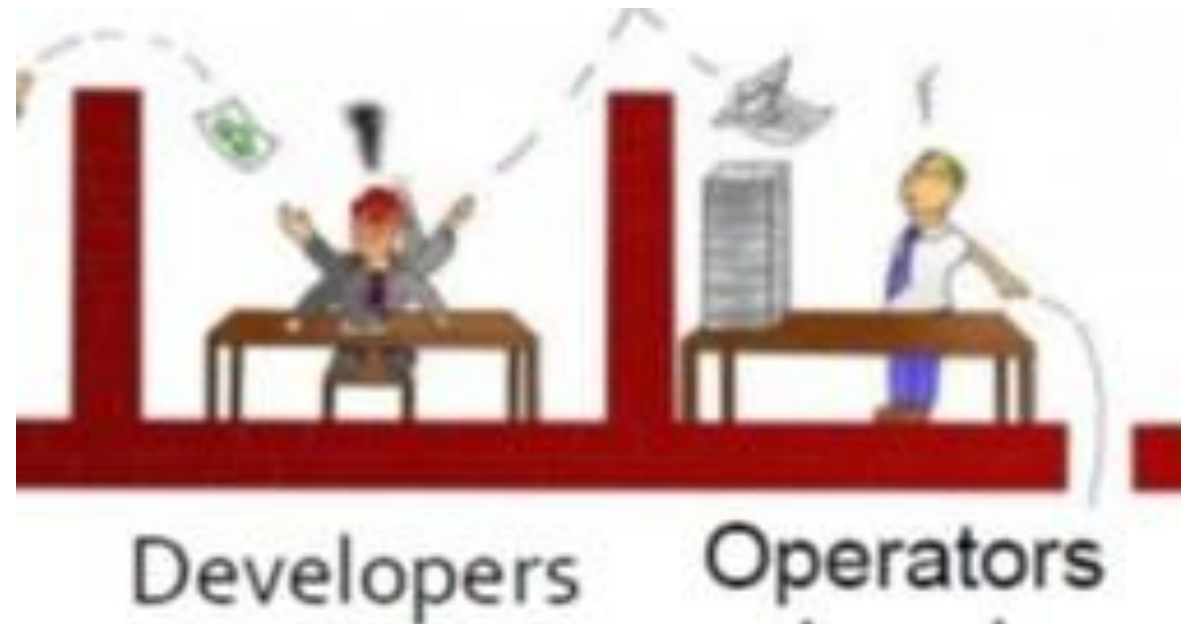


<https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology>

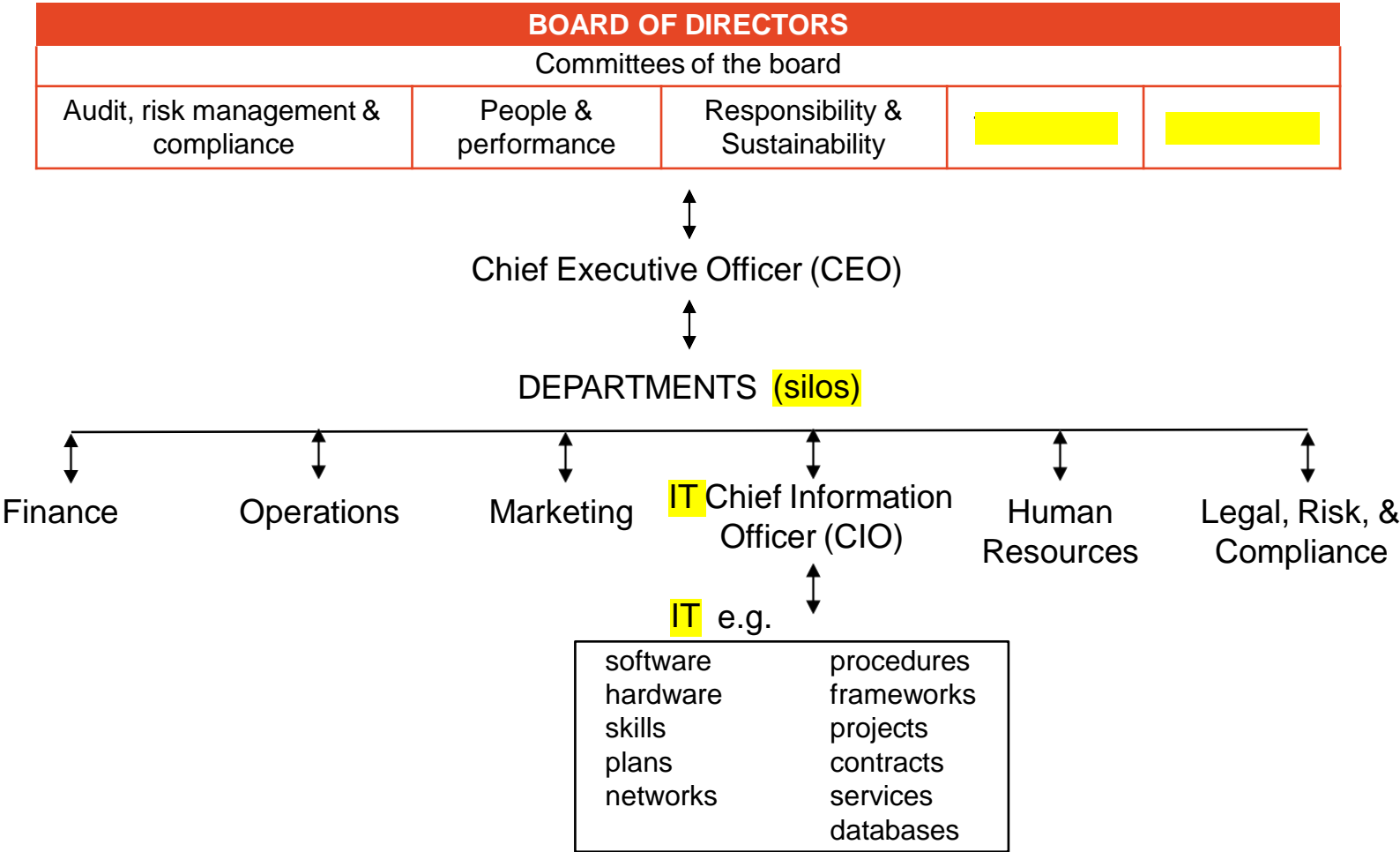
## .... and at the end of the development cycle -

---

- When the project to develop the software application was completed, the result was then moved into the domain of operations to be run in the live environment.
- Errors and changes would be passed back to the development domain.



# The linear approach: business and IT silos

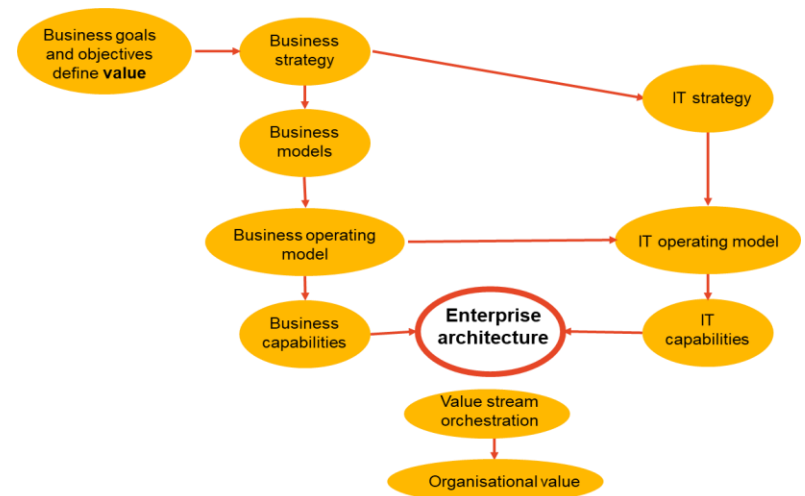




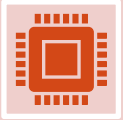
# IT continuous lifecycle approach

---

- While the waterfall method is still appropriate for many situations, various factors have enabled other methods to also emerge, with the consequence that:
  - IT is increasingly inextricably intertwined with all aspects of an organisation rather than provided as a hands-off service to the business.
  - The IT lifecycle is iterative and continuous over time.



# Factors in this shift



Increased technological capacity and speed (e.g. in hardware), and new capabilities (e.g. virtualisation, and AI) → increase not just speed of operation but speed of development as well



The increased sophistication and automation of Enterprise Architecture as a reliable repository of information about components, relationships among them, and principles guiding their development and use



The development of IT service management (ITSM) processes that align all aspects of development, operation, and use of IT with a business - from strategy to continuous improvement (e.g. ITIL)

## Factors in this shift (cont.)



**Agile approaches** - iterative, small scale, incremental, collaborative, and cross-functional teams (that cut across silos)



**DevOps** - the integration of development and testing, continuous releases, and cross-functional teams



**Value stream approaches** – customer-focused orchestration of business and IT capabilities to create a value stream using cross-functional teams



Increased focus on aligning IT strategy with business strategy, IT governance, and compliance

# Waterfall vs Agile vs DevOps

## Waterfall



## Agile



## DevOps



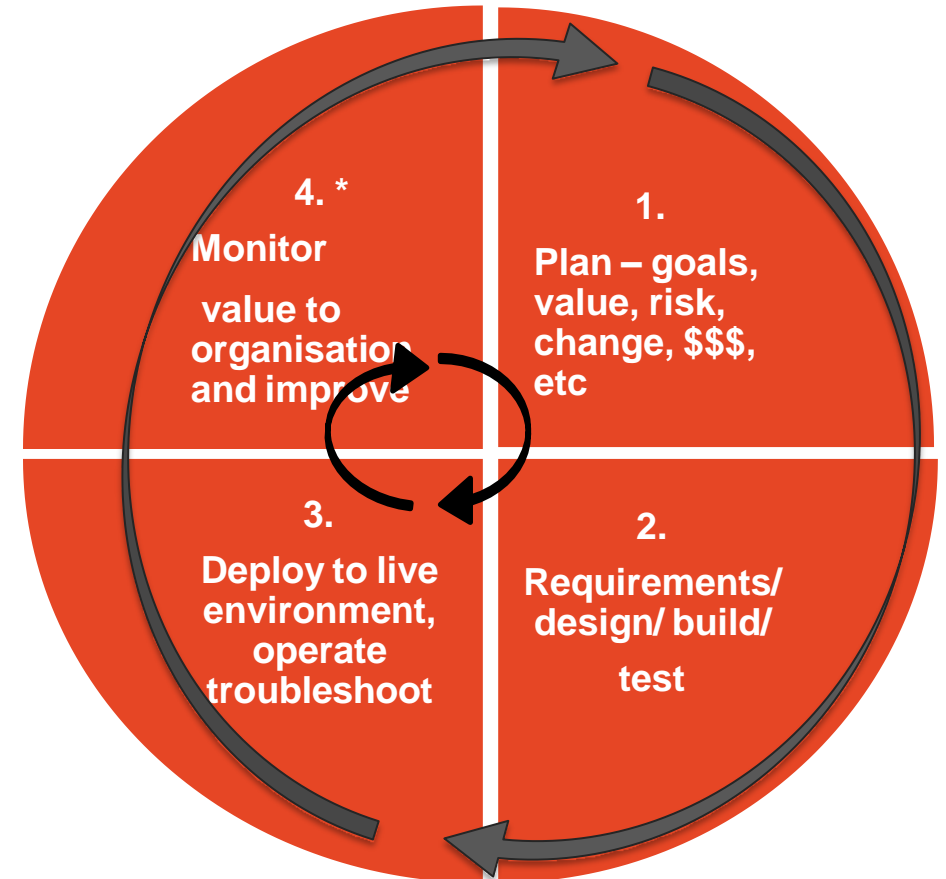
# Current approaches to the IT lifecycle

---

- Current approaches to the lifecycle of an IT capability such as a **software application** treat it as a continuous cycle of processes that:
  - Are initiated by e.g. a customer need,
  - Or, are initiated by an organisational need to:
    - Drive growth,
    - Deliver cost savings,
    - Reduce risk
  - Are aligned with the organisation's goals and strategies
  - Integrate both business and IT capabilities
  - Are responsive to rapid change
  - Deliver measurable business value

# Elements of a continuous IT lifecycle

1. Planning the development (aligned with business strategy)
2. Establishing requirements, designing, building (or acquiring), testing
3. Deploying to the live environment (integrating with other e.g. other applications, middleware, operating systems), maintaining service levels (e.g. troubleshooting) and measuring performance
4. Monitoring the value to the organisation. Depending on the results of the monitoring a new planning process would begin for refinements and improvements, making the cycle continuous (that cycle time can vary from very short to long)
5. In this way, the continuous IT lifecycle embodies all the elements of Continuous Improvement (which we will discuss in a later week).



# Some frameworks and approaches for managing the continuous IT lifecycle

---

- This idea of a continuous IT lifecycle is reflected in frameworks and approaches such as **IT service management (ITSM), agile and DevOps**.
- The critical enablers of the success of these frameworks and approaches are the technological advances mentioned earlier- orders of magnitude increases in technology capacity (for example in hardware), new software capabilities (for example Virtualisation), and AI
- These are the basis for the complex automated tools for example, workflow management, continuous integration, continuous delivery, integrated security, enterprise architecture, and many others that make the continuous lifecycle possible.

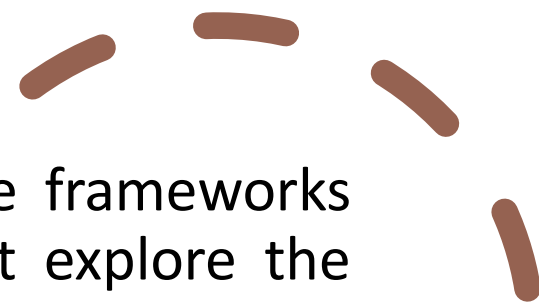
# Difference Between Linear Approach and Continuous Lifecycle Approach

Aspect	Linear Approach	Continuous Lifecycle Approach
Definition	A <b>step-by-step, sequential</b> method where each phase is completed before moving to the next.	A <b>flexible, iterative</b> approach where development, testing, and deployment happen continuously.
Process Flow	<b>One-way, structured</b> progression (e.g., <b>Waterfall model</b> ).	<b>Cyclical and iterative</b> , allowing feedback and improvements (e.g., <b>Agile, DevOps</b> ).
Flexibility	<b>Rigid</b> – Changes are difficult to implement once a phase is completed.	<b>Highly adaptable</b> – Continuous improvements and iterations are possible.
Risk Management	High risk – Issues are often detected <b>late in the process</b> (e.g., during testing).	Lower risk – Continuous testing helps identify and fix issues <b>early</b> .
Examples	Traditional software development, large-scale enterprise systems, regulatory projects.	Agile software development, DevOps, cloud-based applications.





# The continuous IT lifecycle and enterprise architecture

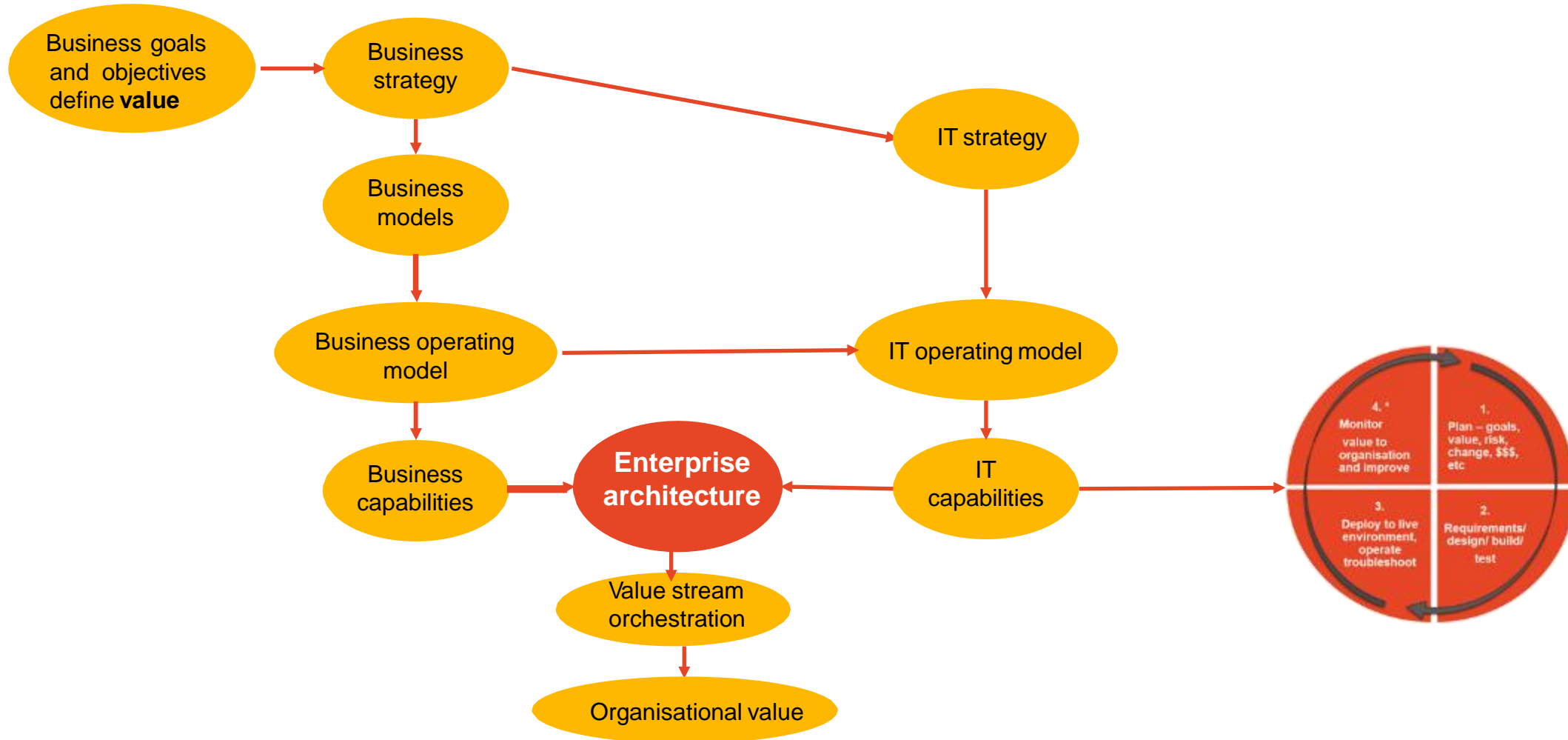


Next, before discussing these frameworks and approaches, we will first explore the nature of enterprise architecture—an important enabler of the continuous lifecycle. Then, we will discuss the frameworks in more detail.

## **2. Enterprise Architecture and its Role in the Continuous IT Lifecycle**

---

# Enterprise architecture



# Enterprise architecture

---

- **Enterprise Architecture (EA)** is a strategic framework that defines the structure, processes, and technology of an organization to align IT and business goals.
- **Why is EA Important?**
  - Helps organizations manage complexity and adapt to change.
  - Provides a holistic view of business processes, IT systems, and data.
  - Ensures alignment between business strategy and IT infrastructure.
  - Supports efficiency, scalability, and innovation in a rapidly evolving digital landscape.
- **Key Focus Areas of EA:**
  - Business Architecture – Defines business processes, goals, and organizational structure.
  - Application Architecture – Manages software systems and their interactions.
  - Data Architecture – Structures data assets for consistency and accessibility.
  - Technology Architecture – Specifies hardware, networks, and IT infrastructure.

# Enterprise architecture frameworks

---

- Enterprise Architecture (EA) frameworks (TOGAF, Zachman Framework, FEAF, MODAF) provide structured methodologies, principles, and best practices for designing and managing an organization's IT and business alignment.
- One of the **most commonly used** is **TOGAF** – The Open Group Architecture Framework.
- <https://www.opengroup.org/>

# Enterprise architecture - TOGAF

TOGAF is based on four interrelated areas of specialization called architecture domains:

- **Business architecture**
  - Defines the business strategy, governance, organization, and key business processes of the organization.
- **Data architecture**
  - Describes the structure of an organization's logical and physical data assets and the associated data management resources

# Enterprise architecture – TOGAF (cont.)

- Applications architecture
  - Provides a blueprint for the individual systems to be deployed, the interactions between the application systems, and their relationships to the core business processes of the organization with the frameworks for services to be exposed as business functions for integration
- Technical architecture, or technology architecture,
  - Describes the hardware, software, and network infrastructure needed to support the deployment of core, mission-critical applications
  - [TOGAF Wikipedia page](#)

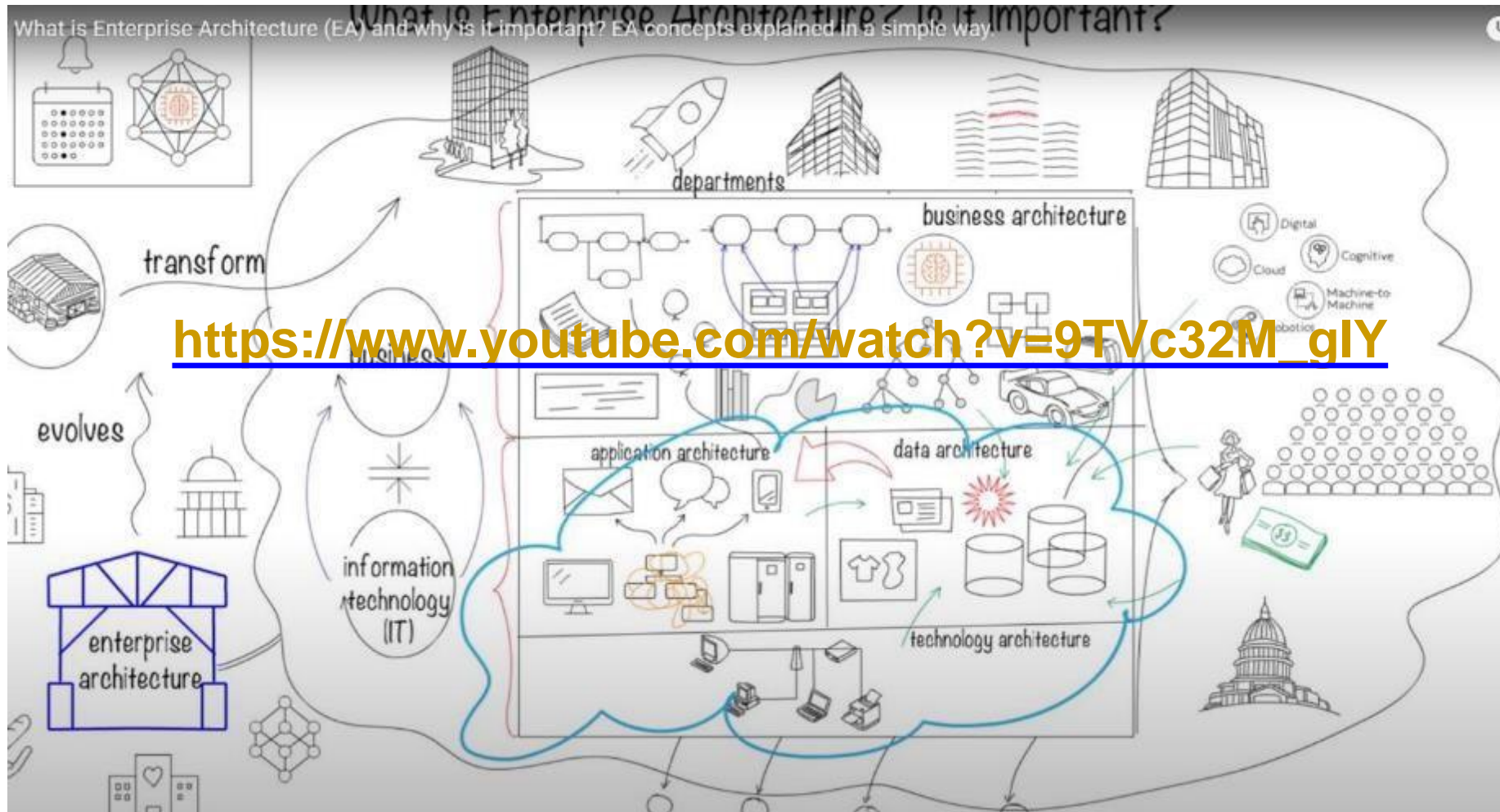
# Some benefits of enterprise architecture

---

- a) Improved decision-making;
- b) Improved adaptability to changing demands or market conditions;
- c) Elimination of inefficient and redundant processes;
- d) Optimisation of the use of organisational resources and capabilities;
- e) Support organisation changes for redesigns and reorganization;
- f) Facilitates collaboration across the organisation
- g) Facilitates the use of frameworks, methods, and approaches (e.g. project management, IT service management) as a single source of information



# This video explains enterprise architecture concepts in a simple way



### **3. IT Service Management (ITSM) Frameworks**

---

# A brief glimpse into ITSM history

---

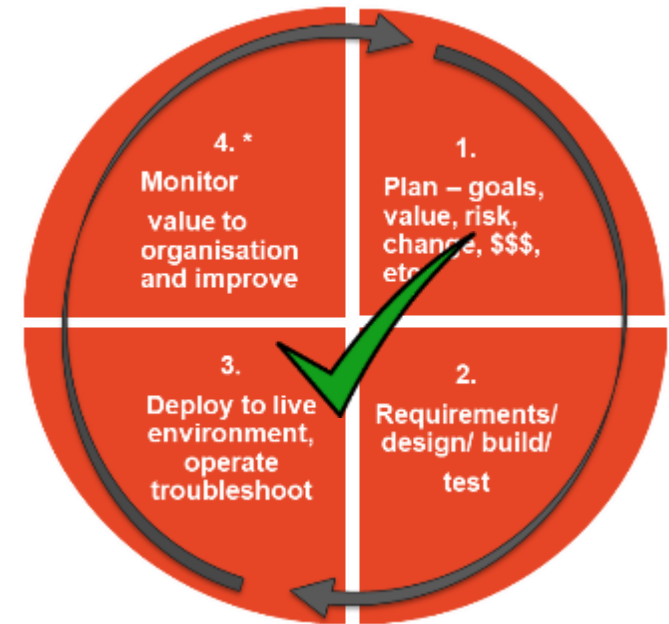
- In earlier days of IT, there was little or no formalised feedback loop or integration between development and operations domains.
- The origins of today's IT service management frameworks lie in work of the IT operations section of a large government department based on their experiences of completed IT projects in operation. The absence of standardised process for managing operational events, evaluation, and feedback, led to poor quality and costly IT services.
- This led to the formalisation of the concept of an integrated and continuous cycle of processes for the planning, producing, evaluating, and improving of IT services -breaking down the barriers between development and operations.

.

# What is ITSM?

---

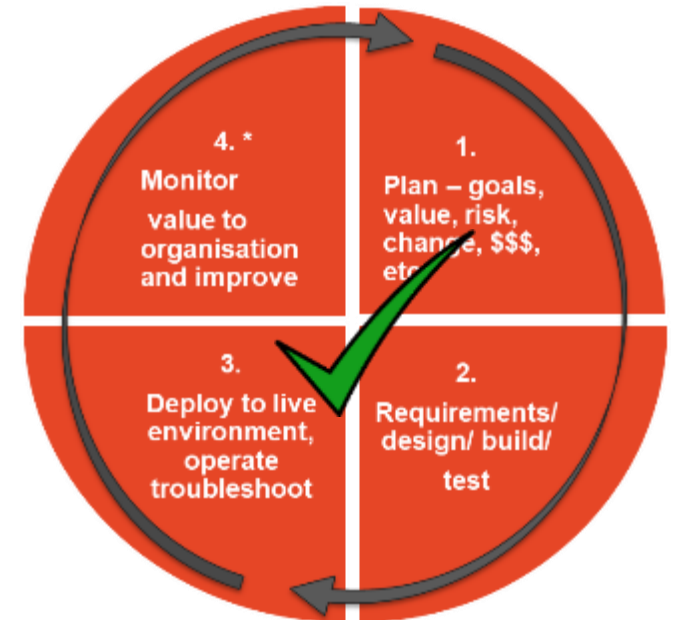
- **ITSM is a strategic approach** to IT management that delivers value to customers/users continuously with increased efficiency and effectiveness.
- The core concept of ITSM is that:
  - An organisation's IT-based outcomes should be delivered continuously
  - through a set of policies, processes, tools, and skills for both development and operations
  - to create value aligned with organisational goals and strategy
  - rather than be the result of a development project handed over to a separate domain of operations to be run
- The most commonly used framework for ITSM currently is [ITIL](#)



# Key Benefits of ITSM

---

- Improved service delivery and customer satisfaction
- Enhanced efficiency through structured processes
- Reduced IT operational costs and risks
- Continuous monitoring and service improvement
- Better alignment of IT services with business goals

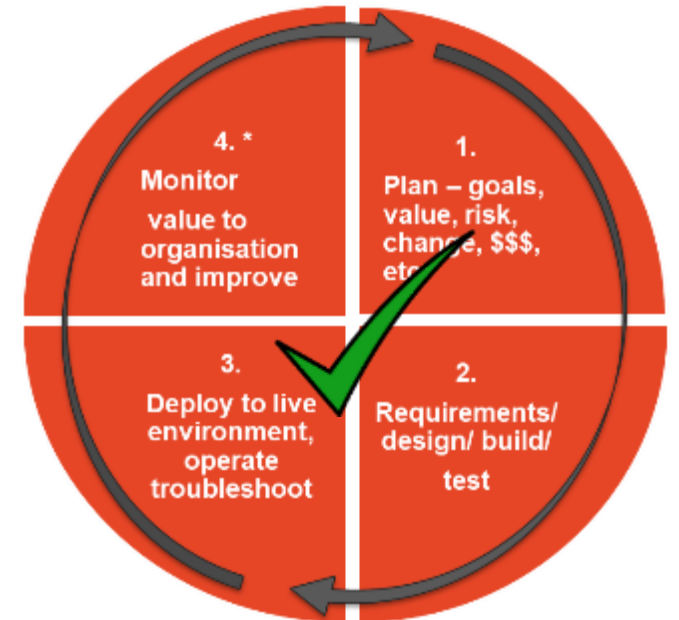


# Understanding ITIL in ITSM

---

ITIL (Information Technology Infrastructure Library) is the most widely used ITSM framework

- It provides best practices for IT service management
- Focuses on aligning IT services with business needs
- Five core components:
  - Service Strategy
  - Service Design
  - Service Transition
  - Service Operation
  - Continual Service Improvement



## 4. The agile approach

---

# What is Agile?

- Agile is an approach to software development that seeks the **continuous delivery of small pieces of working software** in rapid iterations to improve flexibility, adaptability, and customer satisfaction
- Usually, it consists of small, cross-functional teams regularly meeting in person throughout the lifecycle



<https://www.biggerplate.com/mindmaps/OKG4aO2i/agile-principles>



## Agile (cont.)

- Agile values are based on four key principles that should be prioritized over others in software development:
    - Individuals and interactions over processes and tools
    - Working software over comprehensive documentation
    - Customer collaboration over contract negotiation
    - Responding to change over following a plan
- <https://www.redhat.com/en/topics/devops/what-is-agile-methodology>



<https://www.biggerplate.com/mindmaps/OKG4aO2i/agile-principles>

# Agile: example 1

---

- In one European bank undergoing a full-scale evolution, agile has become the default way for people to work, with colleagues from multiple functions including IT, sitting side by side.
- Results are measured by **value streams**—the sources of the value being generated—and journeys, flowing from the customer need back to the performance of the bank.
- Prioritization and resourcing take the form of active daily and weekly conversations about the next most important thing to work on.

<https://www.mckinsey.com/~media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/introducing%20the%20next-generation%20operating%20model/introducing-the-next-gen-operating-model.ashx>

## agile: example 2

---

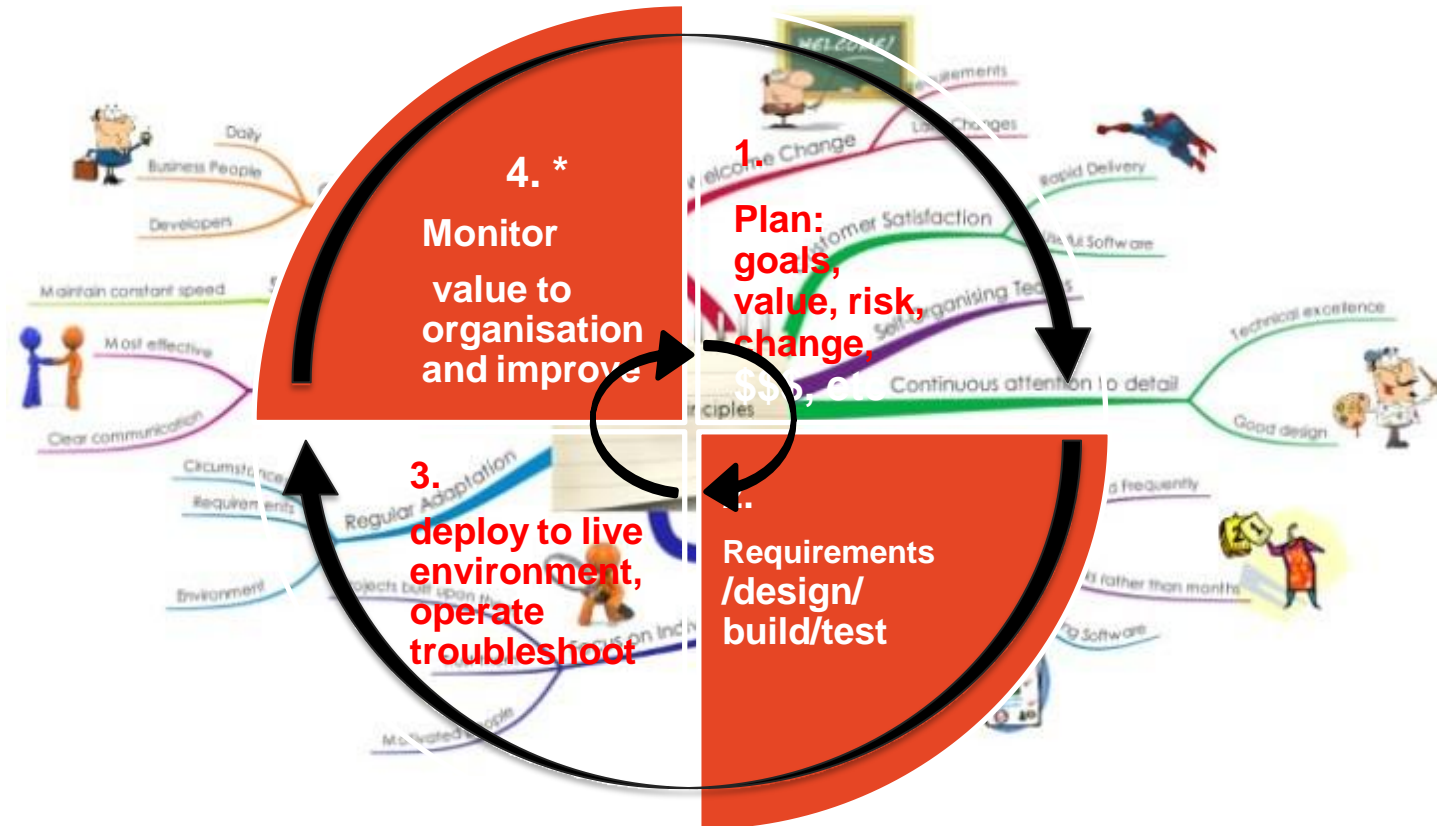
- At another European financial institution, an agile program had been mobilized for three months, and individual teams were working hard, but no real progress had been made in building the planned offering.
- Why? The company had formed a cross-functional team that included all the key units,
- But its members still reported to functional heads and were housed in six different locations.
- Recognizing the initiative was stuck, the company appointed a single executive as end-to-end leader and held five full-day in-person meetings, allowing many people to meet their peers face-to face for the first time.

<https://www.mckinsey.com/~media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/introducing%20the%20next-generation%20operating%20model/introducing-the-next-gen-operating-model.ashx>

## 5. DevOps

---

# “DevOps integrates the culture and practice of agile with ITSM” (Atlassian)

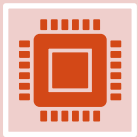


<https://www.biggerplate.com/mindmaps/OKG4aO2i/agile-principles>

# What is DevOps?



DevOps **combines business planning**, software development and operations into a single cycle of processes which overlap.



DevOps is an approach to delivering value from investment in IT capabilities based on agile principles, which break large projects into smaller deliverables and multiple deployments.



**It creates business value by:**

increasing software reliability and flexibility,  
enabling a faster time to market to quickly seize market opportunities,  
enhancing customer experiences from faster feedback cycles,  
facilitating innovation

# What is DevOps and benefit? (cont.)

- Business owners, as well as development, operations, and quality assurance people:
  - no longer operate in silos where decision making was time consuming and risked not being aligned with the goals and strategy of the organisation,
  - collaborate in cross-functional teams to deliver software using a continuous cycle of processes.
- This is made possible by automated tools and environments for development, testing, integration, deployment, monitoring, improvement and quality assurance  
*(Devops vs. Compliance: A guide to having it all: Atlassian.com)*

## DevOps Benefits:

- Speed
- Rapid delivery
- Reliability
- Improved collaboration
- Security

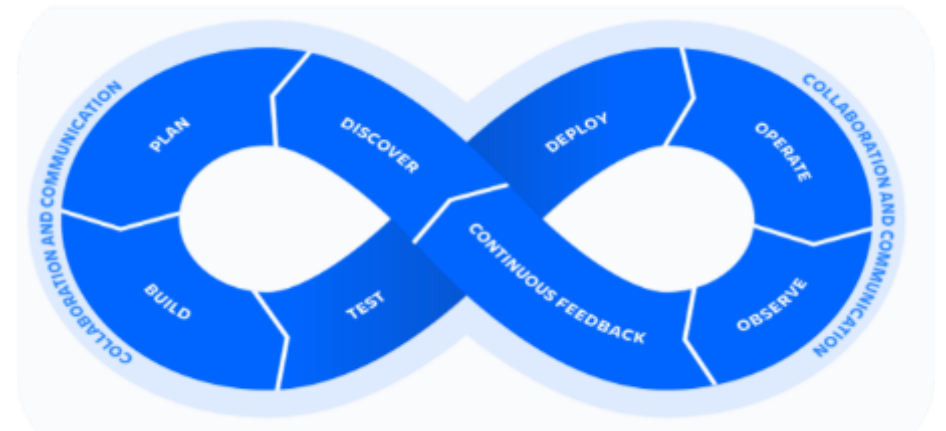
# How does it work?

- DevOps breaks **large projects into smaller deliverables** and multiple deployments
- These are easier to manage from design to deployment and operation.
- Automated tools and environments for developing, testing, integrating, deploying, monitoring and improvement are critical enablers of DevOps



# Iterative phases of the devops cycle

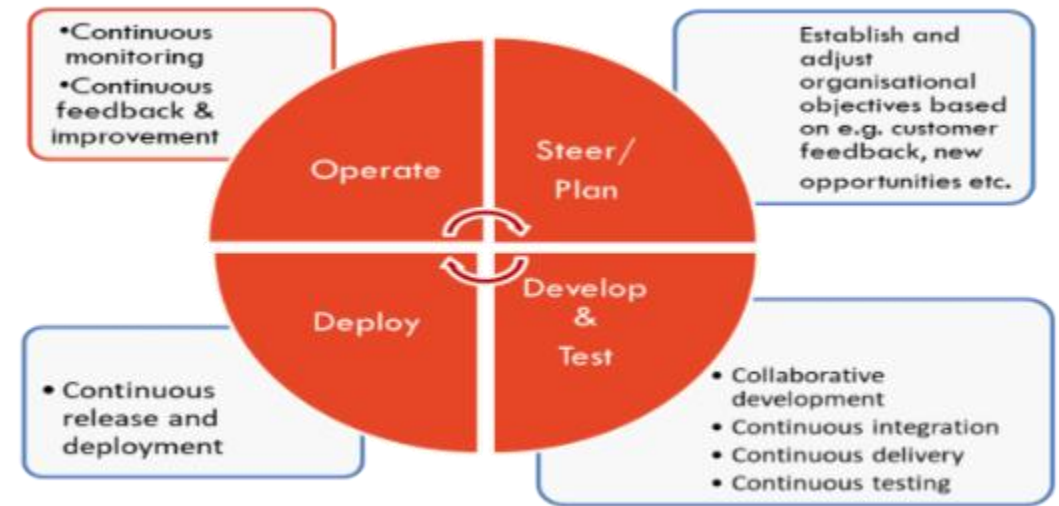
- To represent the continuous nature of DevOps, practitioners often use the infinity loop (lemniscate) to show how the phases of the DevOps lifecycle relate to each other.
- Despite appearing to flow sequentially, the loop symbolizes the need for constant collaboration and iterative improvement throughout the entire lifecycle.



<https://www.atlassian.com/devops>

# Iterative phases of the DevOps cycle (cont.)

- Breakdown of the iterative phases of the DevOps cycle

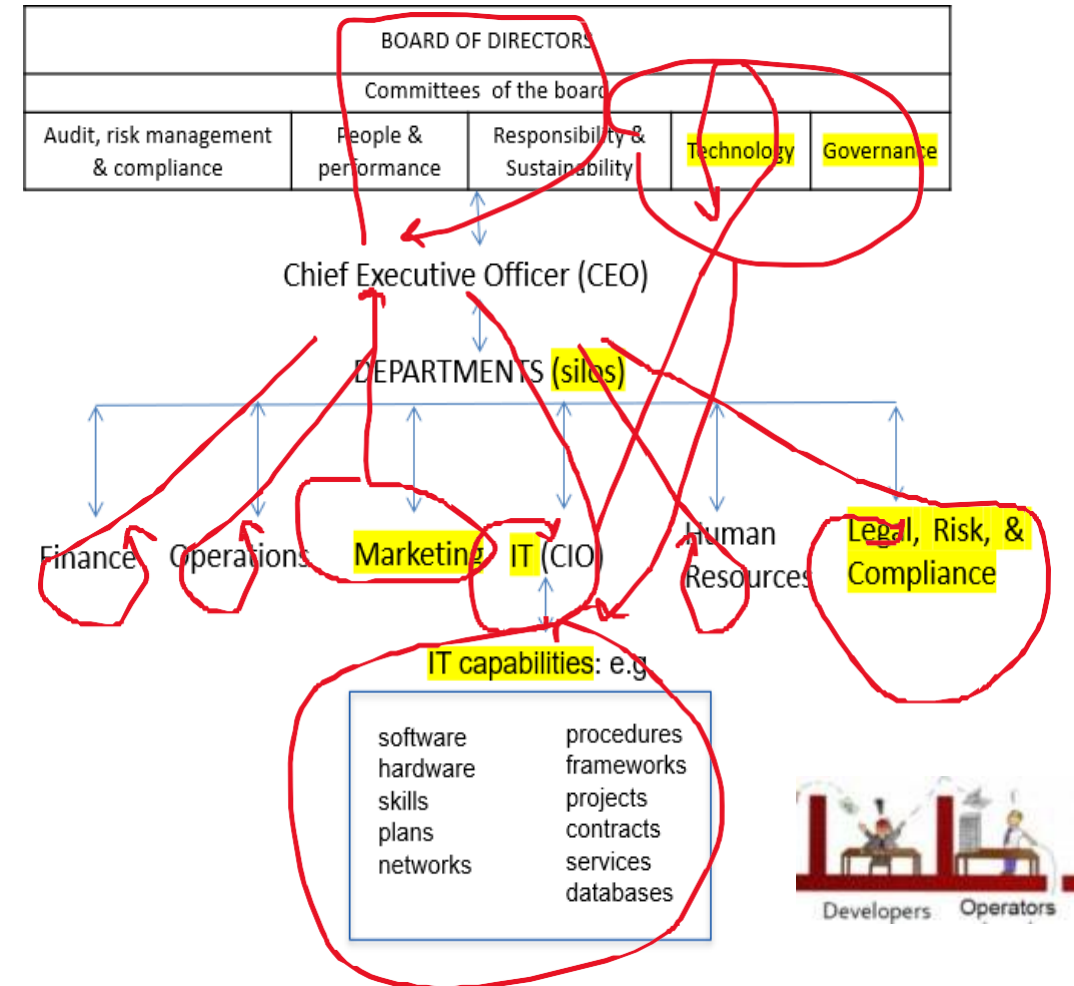


For a simple and clear explanation of DevOps watch this YouTube video

[https://www.youtube.com/watch?v=\\_l94-tJlovvg](https://www.youtube.com/watch?v=_l94-tJlovvg).( after the first 5 minutes it includes a bit of advertising)

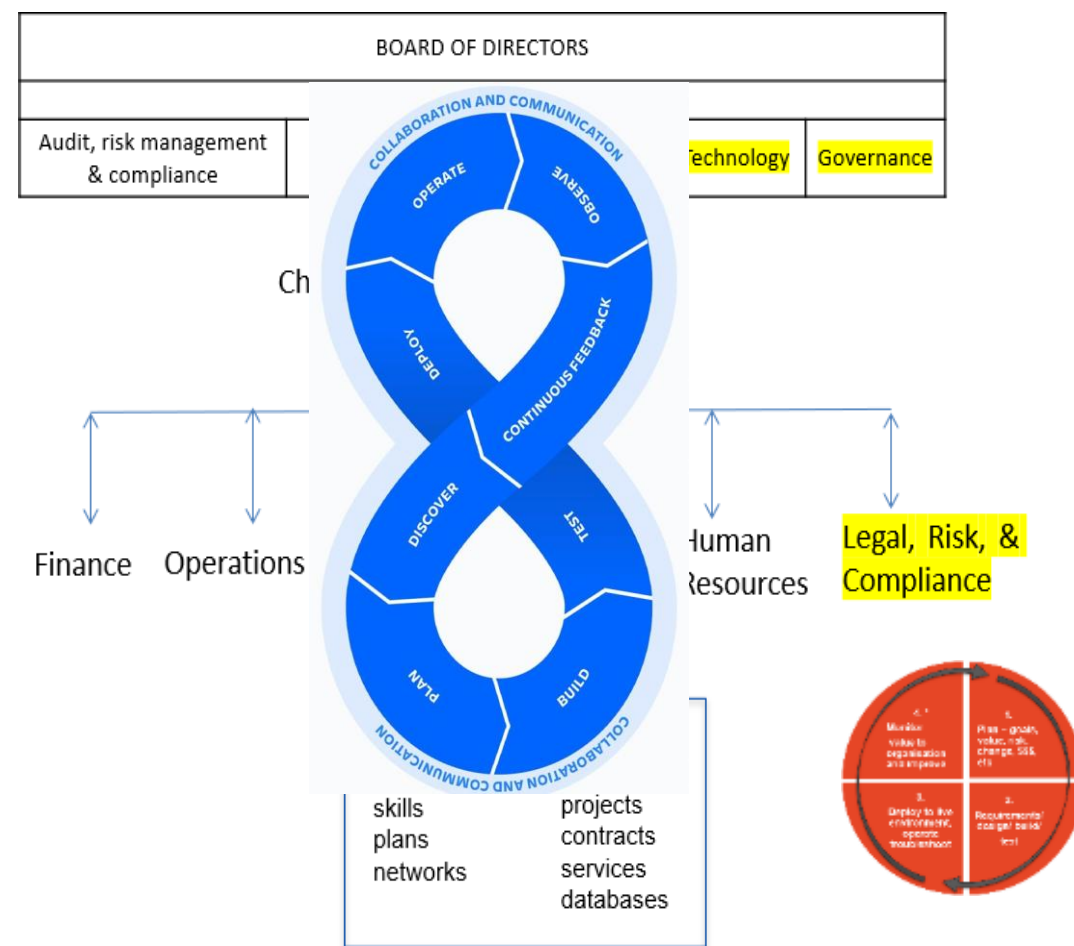
# The effect of silos on the cycle time of getting a new app to market for a supermarket (e.g. “Colesworth”)

Where's the Vegemite?

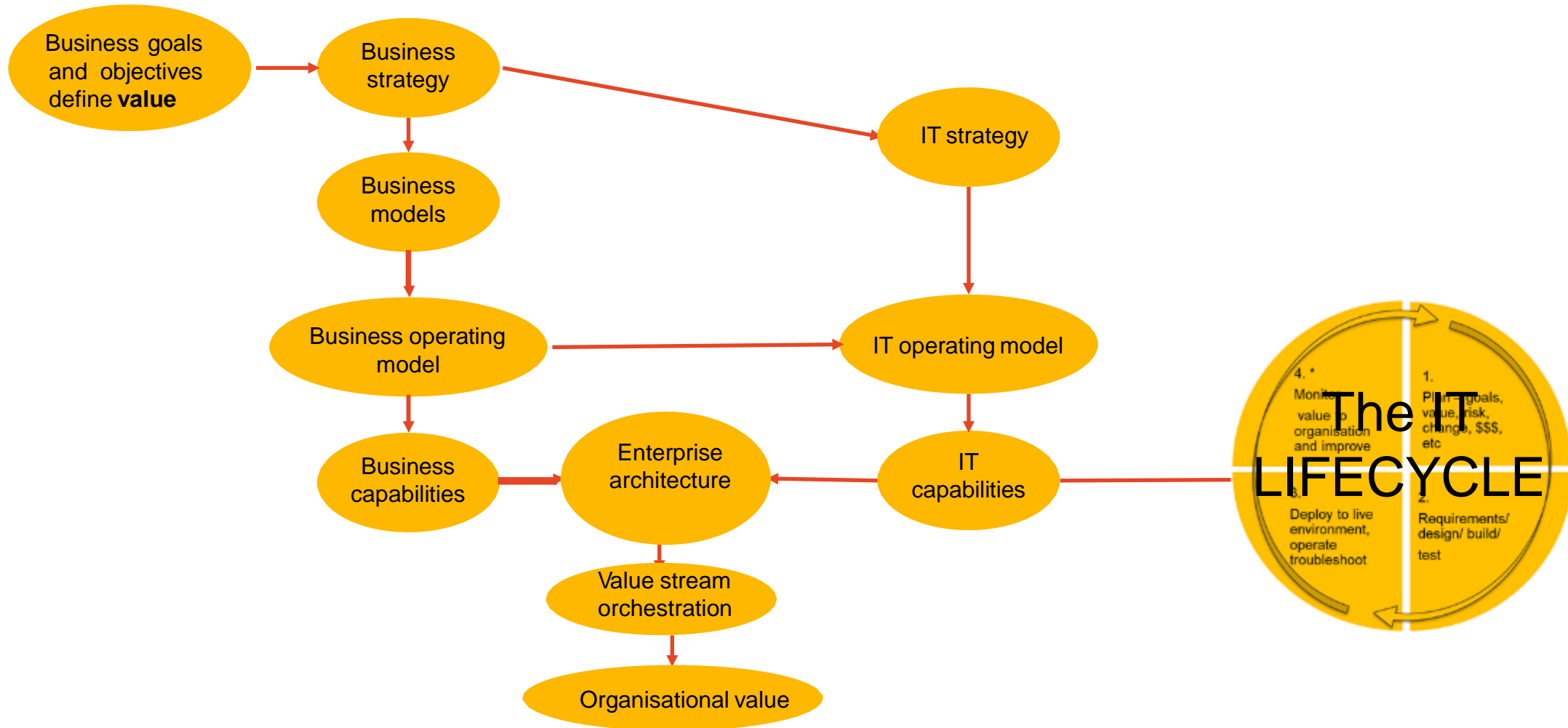


# The continuous lifecycle: breaking down the silos for faster and better outcomes

- Where's the Vegemite?



# Wrap up: Creating organisational value with IT investments. & The IT lifecycle.



# Different Concepts and Their Relation to the IT Continuous Cycle

Concept	Purpose	Key Focus	Relation to IT Continuous IT Cycle
Enterprise Architecture (EA)	Aligns <b>IT strategy with business goals</b>	Standardized <b>IT planning, governance, and integration</b>	Provides a <b>structured foundation</b> for continuous IT planning, enabling <b>scalability and long-term system evolution</b> .
IT Service Management (ITSM)	Ensures <b>efficient and reliable IT operations</b>	Service delivery, incident management, support	Supports IT operations by <b>standardizing processes and ensuring continuous service improvement</b> in the IT cycle.
Agile	Increases <b>flexibility and collaboration</b> in software development	Iterative delivery, continuous feedback	Enables <b>adaptive IT evolution</b> by integrating iterative feedback loops within the <b>development and deployment cycle</b> .
DevOps	Enables <b>continuous development, testing, and deployment</b>	Automation, CI/CD, faster software delivery	Facilitates <b>continuous integration and delivery (CI/CD)</b> , ensuring <b>rapid and reliable software updates</b> in the IT cycle.
Value Stream	Focuses on <b>delivering customer value efficiently</b>	Maps and optimizes <b>end-to-end processes</b>	Ensures <b>continuous improvement in IT processes</b> by aligning <b>business and IT workflows</b> for optimized value delivery.

End