01. Software architecture Intro

Software Architectures

Master in Informatics Engineering

Cláudio Teixeira (claudio@ua.pt)

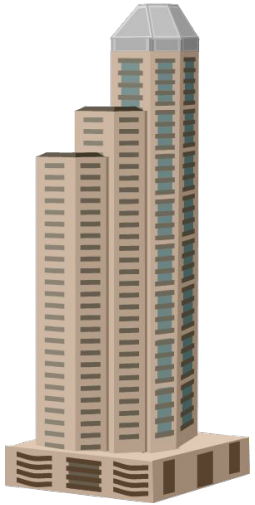
Agenda

- Meaning of software architecture
- Software architecture in organizations
- Organization & IT - Meet TOGAF & COBIT
- Group practical assignments

1.1 - What is software architecture?

- there are multiple definitions to it.
- Let’s agree on some views of it

What is software architecture?



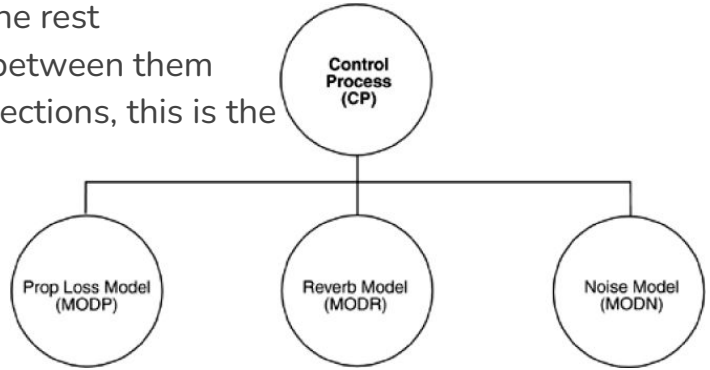
- there are multiple definitions to it.
- Let’s agree on some views of it

As the size and complexity of software systems increases, the design problem goes beyond computing algorithms and data structures: ***conceiving, designing, and specifying the overall structure of the system emerges as a new kind of problem****… This is design at the level of software architecture. (Garlan, 1992)*

The software architecture of a program or computing system is the structure or structures of the system, encompassing ***software components****, the* ***externally visible properties*** *of these components, and* ***the relationships between them****. (Bass et all, 2003)*

What does the diagram tells us?

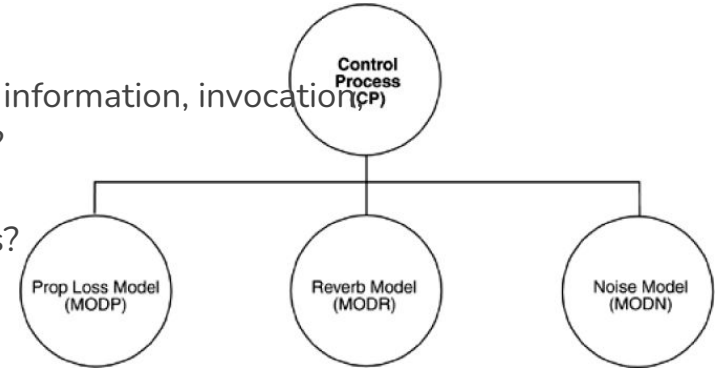
- There are 4 elements: CP, MODP, MODR, MODN
- 3 of the elements must be more similar to each other than to the CP
- All elements have a type of link/relationship with the rest
- In short: we have 4 components with connections between them
- As an architecture is a set of components and connections, this is the blueprint of an architecture.
- Is it?



Top view of the architecture of a control system for an underwater acoustic simulation

What does the diagram tells us?

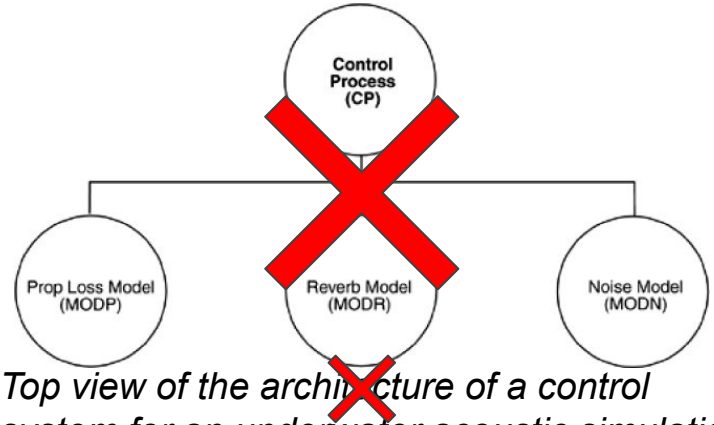
- Are they distributed functions, processes, programs, or something else?
- What are their responsibilities?
- What do they do?
- What is their role in the system?
- What do the links mean?
- Do the links mean communication, control, sending information, invocation, synchronization, sharing of sensitive information, ...?
- What are the communication mechanisms?
- What information flows through these mechanisms?



Top view of the architecture of a control system for an underwater acoustic simulation

What does the diagram tells us?

- What is the nature of the elements?
- Is the elements separation significant?
- Do they run on different processors?
- Do they run at different times?
- Do they represent ways to divide the task flow?
- What does the layout mean?
- Why is CP on a separate level?
    - Will it invoke the remaining elements and the rest will not be able to invoke it?
    - Will it contain the remaining elements in its implementation?
    - Aesthetic issue?
    - Was there not enough space horizontally to place all the elements on the same line?



Top view of the architecture of a control system for an underwater acoustic simulation

So … what is it then?

*Architecture is concerned with the selection of architectural elements, their **interaction**, the **constraints** and **rules** in these elements and their interactions... Design is related to the **modularization** and **detailed interfaces of architectural elements**, their algorithms and procedures, the types of data necessary to support the architecture and to satisfy its requirements. (Perry and Wolf, 1992)*

*The architecture… **is specifically silent… on implementation details** (algorithms and data structures). Architectural drawing uses a richer collection of abstractions than is achieved through Object Oriented Design (Monroe et al, 1997)*

Then let there be standards!
ISO/IEC/IEEE 42010:2022

- Software Architecture: *Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*
  - Emphasis is on capturing not only the static structure but also the **dynamic aspects** and **evolutionary considerations** of the system
- Interpretation:
  - Software architecture is a fundamental part of a software system
  - A software system is situated in an environment, and its software architectures takes into consideration the environment in which it must operate
  - An architecture description document the architecture and communicates to stakeholders how the architecture meets the system's needs
  - Architecture views are created from the architecture description, and each view covers one or more architecture concerns of the stakeholders

Why should we care about software architecture? Let’s just code this!

- It is the software’s blueprint, its foundation
- It stands as schematics based on decisions. That could probably not be possible without the schematics in the first place.
- It’s iterative until the final drawings
- Every software has an architecture, so we might as well document it and produce it well!
- Architecture addresses and enables software to fulfill all requirements (functional, non-functional and operational)
    - No architecture, no requirements fulfillment (or a hard time on it, at least)!

Software is only as good as its architecture

- Software quality attributes are impacted by the architectural choices (deliberately or inadvertently)
  - such as... maintainability, interoperability, security and performance (we'll get there during the semestre)
- Looking at the architecture, one can infer which qualities are being favoured and which are deemed as important
  - This saves time and money! If only by implementing it you can determine the qualities... then possible rework will be costly!

Direct effects of a good software architecture

- Ease the communication with stakeholders
  - can form the basis of discussions with the team
  - Its abstract nature means that it *should* be fairly understandable by everyone
- Helps on software changes
  - calculating its impact
  - reduces complexity of software, ergo changes will be simpler
- Provides patterns and reusable models
- Improves estimations for task completion
- Good intro material for newcomers

What about the software architect?

- Provides technical leadership into the project, management, customers and developers
- Responsible for the software architecture, its design and the architecture documentation
- Needs to understand the business domain
- Helps on analysing and gathering requirements
- Technical topics:
  - software architecture patterns (pros and cons)
  - how to handle cross cutting concerns
  - know how to integrate legacy applications
  - be able to design software that adapts to changes and evolves over time
- He is just a team player! can be influenced by the teams' performance and can impact the teams' performance.

1.2 - Software architecture in organizations

Software is built to satisfy an organisation’s goals, right? so...

the organisation’s goals, objectives, stakeholders, project management and processes impact the software architecture and their work.

One software architect to rule them all?

- Enterprise Architect:
  - Focuses on the overall structure and strategy of an organization's IT systems.
  - Aligns IT solutions with business goals and ensures interoperability across different departments.
- Solution Architect:
  - Works on the design and implementation of specific solutions or projects.
  - Develops detailed specifications for how software components should be built and integrated.
- Application Architect:
  - Concentrates on the architecture of a specific application or suite of applications.
  - Defines how individual applications interact within the larger system.
- Data Architect:
  - Specializes in designing and managing an organization's data architecture.
  - Focuses on data models, databases, data flow, and data security.
- Cloud Architect:
  - Specializes in designing and implementing cloud-based solutions.
  - Works with cloud technologies and services to optimize scalability, performance, and cost.
- Security Architect:
  - Focuses on the security aspects of software and system architecture.
  - Designs and implements security measures to protect against cyber threats.

One software architect to rule them all?

- Integration Architect:
  - Ensures that different software systems can work seamlessly together.
  - Designs solutions for integrating diverse applications and systems.
- Mobile Architect:
  - Specializes in designing software architectures for mobile applications.
  - Considers constraints and opportunities specific to mobile platforms.
- UI/UX Architect:
  - Focuses on the user interface (UI) and user experience (UX) aspects of software.
  - Designs the overall look, feel, and usability of software applications.
- DevOps Architect:
  - Integrates development and operations processes to streamline the software delivery pipeline.
  - Focuses on automation, collaboration, and continuous improvement.
- Big Data Architect:
  - Specializes in designing architectures for handling and analyzing large volumes of data.
  - Works with technologies like Hadoop, Spark, and distributed databases.
- IoT (Internet of Things) Architect:
  - Designs architectures for systems that involve interconnected devices.
  - Addresses challenges related to data exchange, security, and scalability in IoT ecosystems.

Software alignment

- how to align software development (and its predecessor, software architecture) with the goals and strategies in the organisation?
- through communication and understanding of goals:
    - architectural decisions and implications must be explainable to the board, if needed!

1.3 - Business alignment and the role of the software architect

Brief intro on IT governance
Aligning IT strategies with business goals

IT Initiatives support and drive business objectives, leading to enhanced competitiveness, operational efficiency and innovation. Alignment enables:

- **Improved Decision-Making:** Alignment helps ensure that IT investments and projects are directly linked to business priorities, facilitating better decision-making at both strategic and operational levels.
- **Increased Agility:** By aligning IT with business strategies, organizations can respond more swiftly and effectively to market changes, customer needs, and emerging technologies.
- **Enhanced Innovation:** A close partnership between IT and business units encourages the exploration of new technologies to support business initiatives, fostering innovation and creating new opportunities for growth.
- **Optimized Costs and Investments:** Alignment helps in prioritizing IT projects and investments according to their potential impact on business outcomes, ensuring resources are allocated efficiently and effectively.
- **Risk Management:** Aligning IT and business strategies helps identify and mitigate risks more effectively, especially those related to IT investments and cybersecurity.

IT Governance and Architectural Frameworks' Role in Organizational Success

Architectural frameworks provide structure, processes and mechanisms to ensure success in multiple ways:

- **Standardization and Best Practices:** Frameworks like TOGAF and COBIT offer standardized methodologies and best practices for IT governance and architecture management. This standardization helps organizations reduce complexity, improve interoperability, and enhance efficiency.
- **Strategic Alignment:** These frameworks provide tools and processes for aligning IT projects and operations with business strategies, ensuring that IT efforts are focused on delivering value and achieving strategic objectives.
- **Risk Management and Compliance:** Governance frameworks ensure that IT systems and processes are compliant with legal and regulatory requirements, helping manage risks associated with information security, data protection, and technology investments.
- **Performance Measurement:** IT governance and architectural frameworks include mechanisms for measuring and monitoring IT performance against business objectives. This enables continuous improvement and ensures IT delivers tangible business value.
- **Resource Optimization:** By providing a structured approach to managing IT resources, these frameworks help organizations optimize their IT investments, ensuring resources are allocated to initiatives that offer the highest business value.

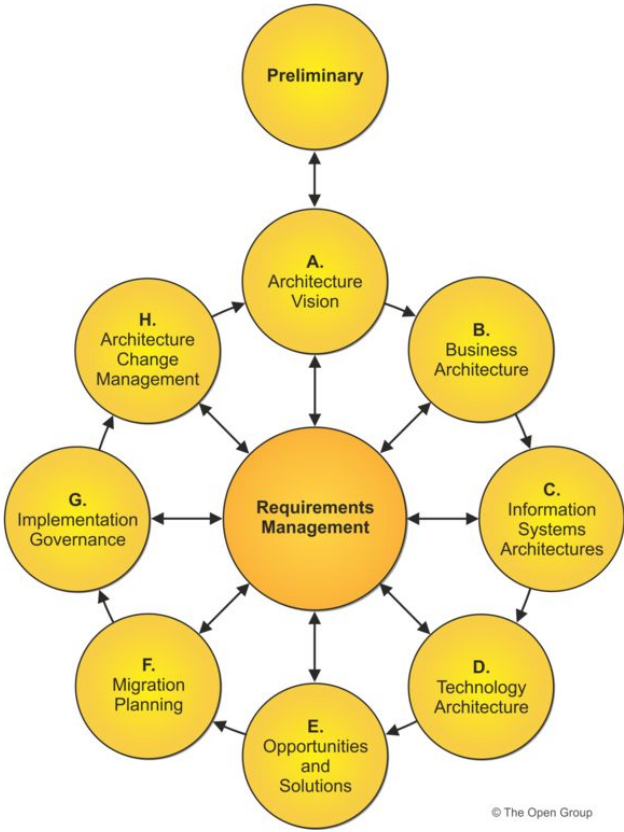
1.3 - TOGAF & COBIT

- TOGAF - The Open Group Architecture Framework
- COBIT - Control Objectives for Information and Related Technologies

- High level frameworks for organizations

TOGAF

- The Open Group Architecture Framework, is a leading framework for enterprise architecture that provides a comprehensive approach to designing, planning, implementing, and governing an **enterprise** information technology **architecture**.
- Aims to align IT goals with the overall business goals, ensuring that the enterprise architecture supports the organization's objectives effectively and efficiently.

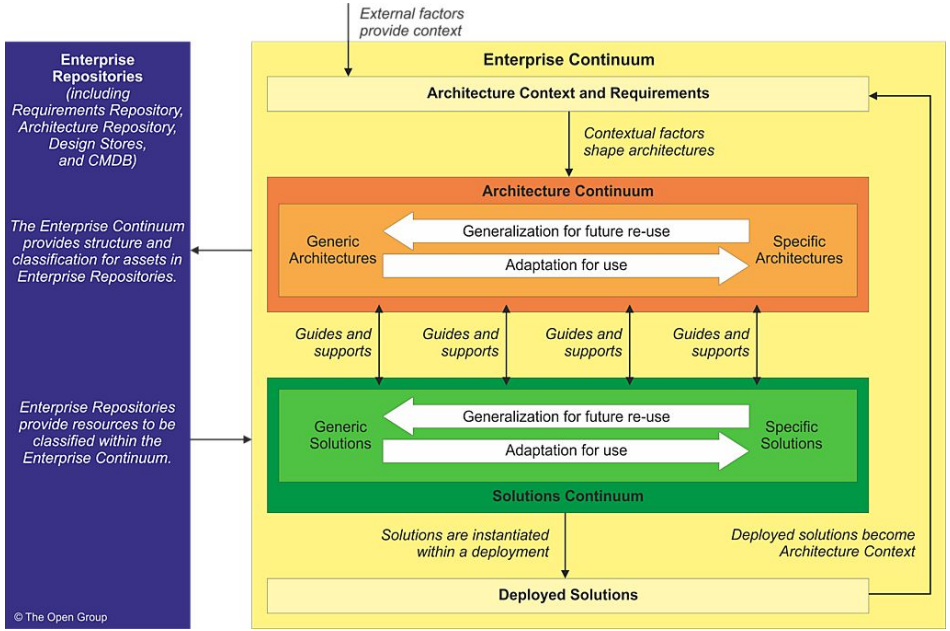


TOGAF Architecture Development Method
https://pubs.opengroup.org/architecture/togaf9-doc/arch/

TOGAF
Enterprise continuum

The Enterprise Continuum provides a framework and context to support relevant architecture assets in executing the ADM.

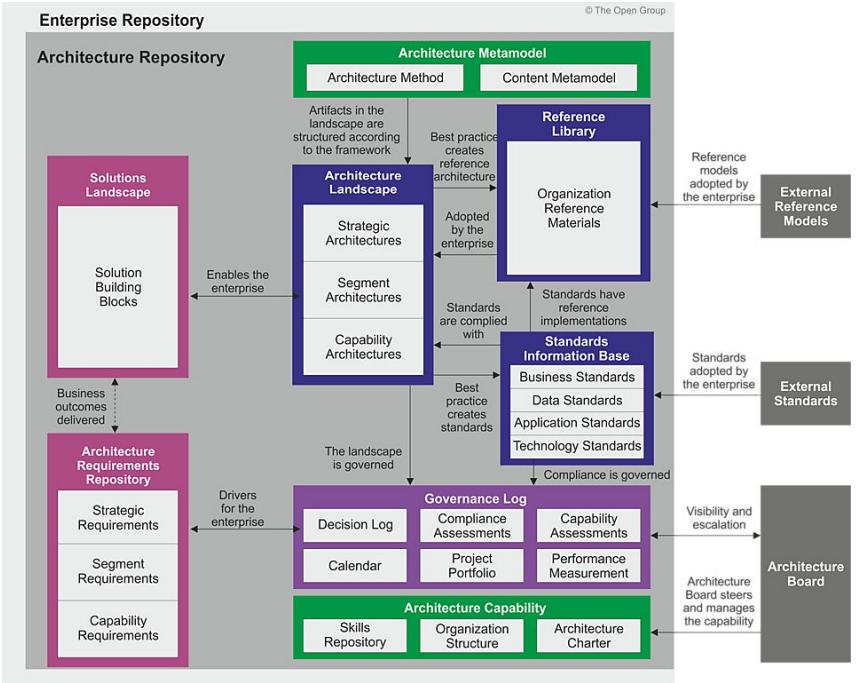
The practical implementation of the Enterprise Continuum will typically take the form of an Architecture Repository.



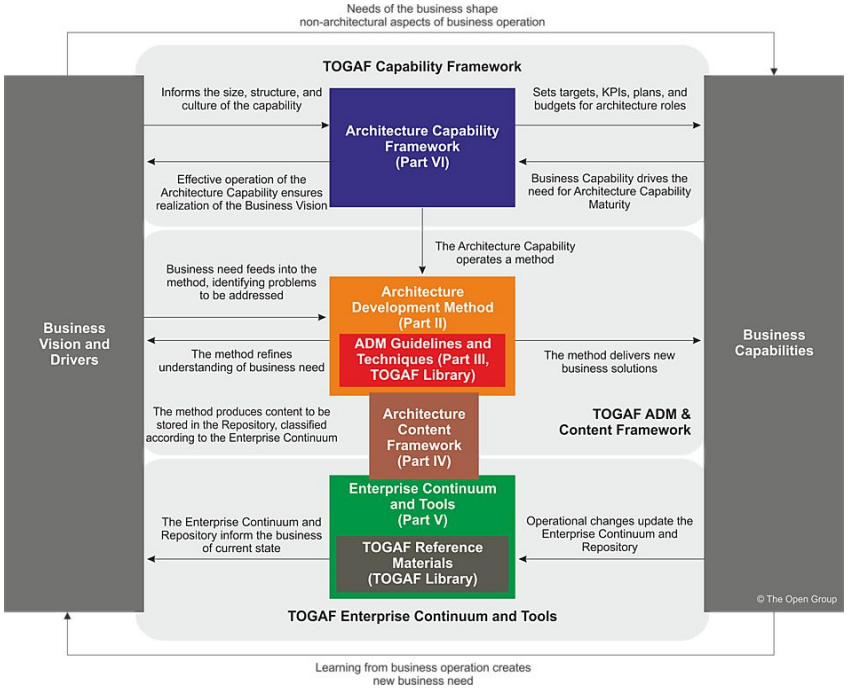
TOGAF
Architecture Repository

Operating a mature Architecture Capability within a large enterprise creates a huge volume of architectural output.

Effective management and leverage of these architectural work products require a formal taxonomy for different types of architectural asset alongside dedicated processes and tools for architectural content storage.



TOGAF


Needs of the business shape
non-architectural aspects of business operation
TOGAF Capability Framework
Informs the size, structure, and culture of the capability
Architecture Capability Framework (Part VI)
Sets targets, KPIs, plans, and budgets for architecture roles
Effective operation of the Architecture Capability ensures realization of the Business Vision
Business Capability drives the need for Architecture Capability Maturity
The Architecture Capability operates a method
Business need feeds into the method, identifying problems to be addressed
Architecture Development Method (Part II)
ADM Guidelines and Techniques (Part III, TOGAF Library)
The method refines understanding of business need
The method delivers new business solutions
Business Vision and Drivers
Business Capabilities
The method produces content to be stored in the Repository, classified according to the Enterprise Continuum
Architecture Content Framework (Part IV)
TOGAF ADM & Content Framework
Enterprise Continuum and Tools (Part V)
TOGAF Reference Materials (TOGAF Library)
The Enterprise Continuum and Repository inform the business of current state
Operational changes update the Enterprise Continuum and Repository
© The Open Group
TOGAF Enterprise Continuum and Tools
Learning from business operation creates new business need

TOGAF Capability Mapping

20 minutes
5 min class discussion

https://docs.google.com/document/d/1vfipWSxlbjbU56XERC-0vO_UZngoFOlXYxEkm7yRIIQ/edit?usp=sharing

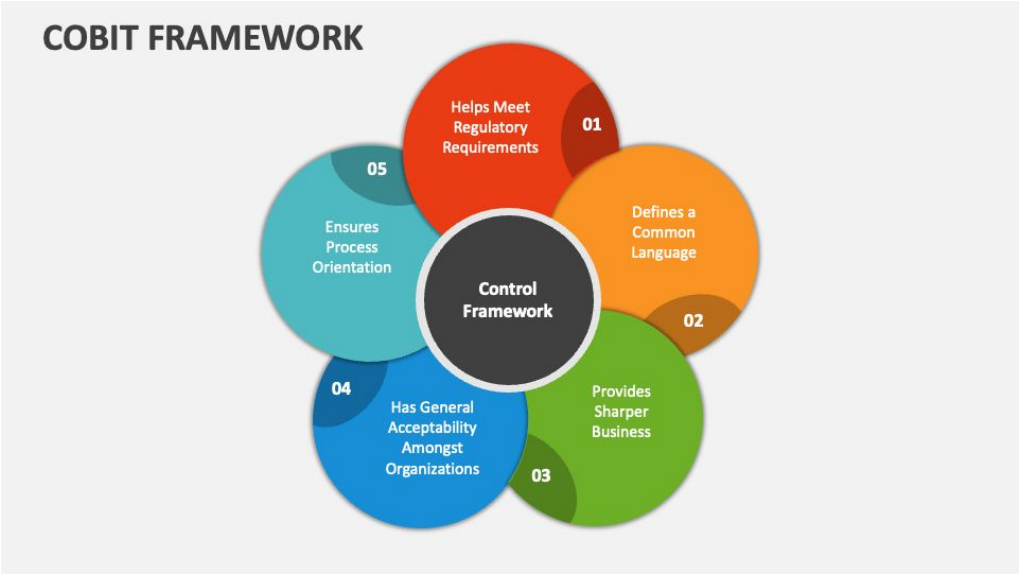
COBIT

COBIT (Control Objectives for Information and Related Technologies) is a globally recognized framework for IT governance and management. Developed by ISACA, COBIT ensures information and technology (I&T) supports enterprise goals, optimizes risks, and provides value.

- Framework that provides a set of best practices for IT governance and risk management
- Supports organizations to ensure their business goals are met through effective IT governance
- It encompasses:
  - Control Objectives:
    - Control objectives structure.
    - Relationship with business processes.
  - Alignment with Business Objectives:
    - How COBIT contributes to achieving organizational objectives.
    - Approach to risk management.

COBIT

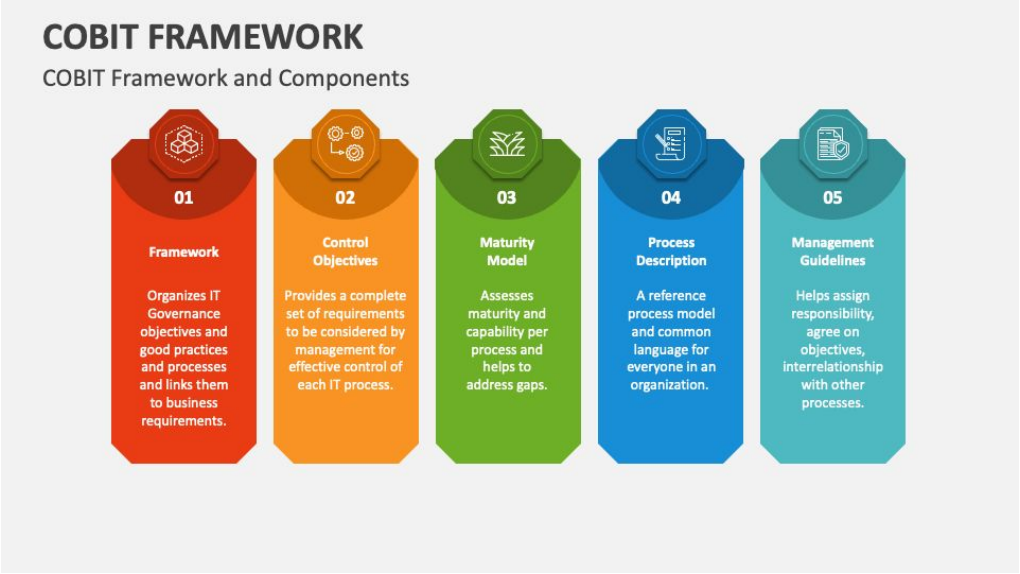
COBIT provides the principles, practices, analytical tools, and models to achieve this alignment and governance.



COBIT

COBIT's key components include Control Objectives, Maturity Models, Processes and Management Guidelines.

Main focus: balancing benefits realization, risk optimization, and resource optimization.

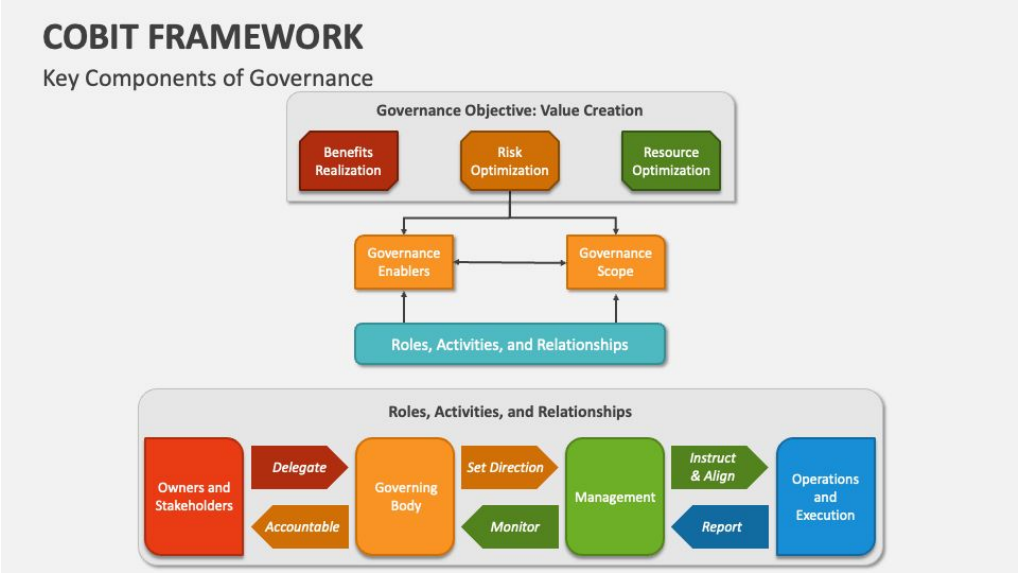


COBIT

COBIT facilitates the alignment of IT with business strategies by providing a framework that translates business objectives into IT goals and metrics.

It helps stakeholders understand their responsibilities toward governance and ensures IT-related decisions are made in alignment with the organization's strategic objectives.

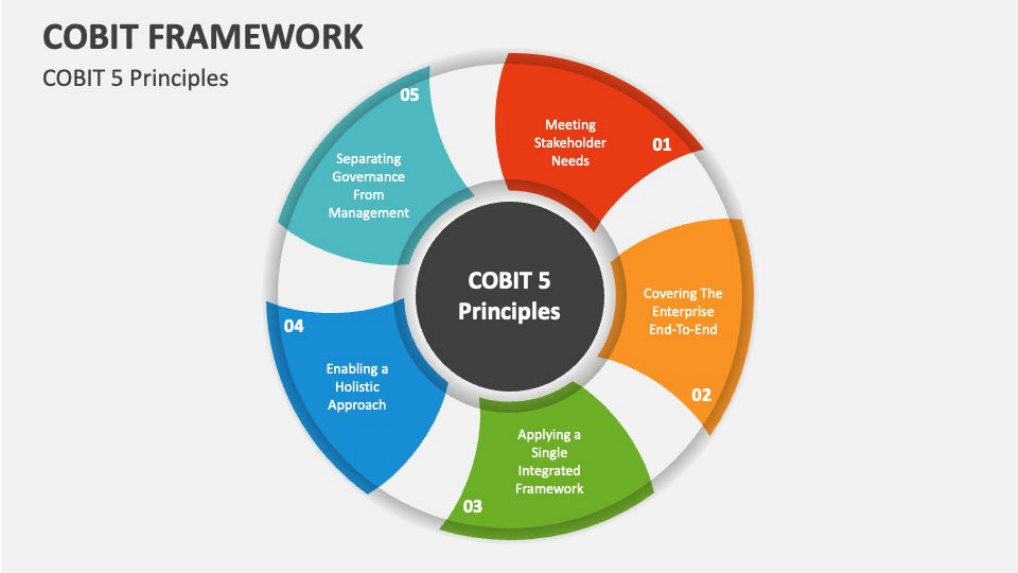
It keeps IT steered by the Owners and Stakeholders.



COBIT

COBIT principles enhance an organization's ability to govern and manage IT resources effectively.

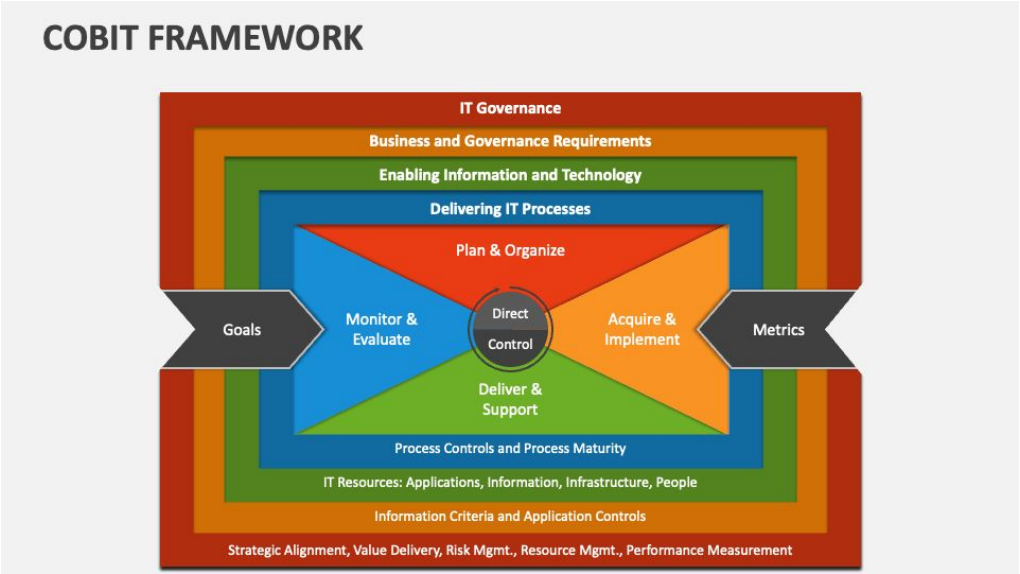
By aligning IT strategies with business objectives, ensuring comprehensive coverage across the enterprise, integrating with other frameworks, adopting a holistic approach, and clearly distinguishing between governance and management, organizations can achieve strategic goals, manage risks effectively, and optimize resource utilization.



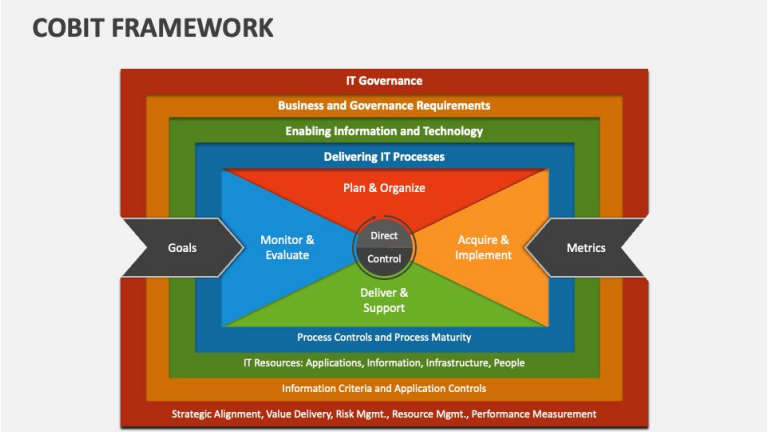
COBIT

Implementing COBIT involves assessing the current state of IT governance, identifying gaps, and applying COBIT principles to develop a tailored governance framework that meets the organization's unique needs.

This includes establishing clear policies, roles, and responsibilities, and setting measurable objectives.



COBIT
Domains



1. **Governance and Framework:**

Focuses on the overall governance structure, ensuring that IT governance is aligned with business goals and delivers value while balancing risk and resource use. It sets the stage for the entire governance framework, defining and establishing the components necessary for effective governance, including policies, processes, organizational structures, and ensuring compliance with relevant laws and regulations.

2. **Strategy and Planning (plan & organize):**

Strategy Management: Involves defining the direction, planning, and tactics necessary to achieve business objectives through IT. It covers the development of IT strategy in alignment with business strategy, ensuring that IT delivers the intended benefits.

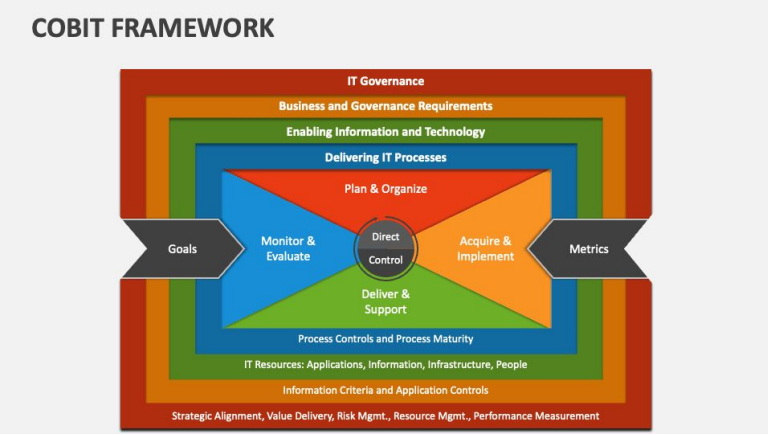
Enterprise Architecture Management: Deals with the structuring of IT infrastructure and operations in a way that supports the organization's strategy. This includes the definition of IT architecture and ensuring it aligns with business processes.

3. **Build and Acquire (Acquire & implement)**

Program and Project Management: Focuses on the management of IT projects and programs, ensuring they are delivered on time, within budget, and according to specifications while delivering the expected benefits to the business.

IT Solution Delivery and Support: Covers the development, procurement, and implementation of IT solutions that meet business requirements. It ensures that IT services are delivered effectively and support business operations.

COBIT
Domains



4. **Deliver and Support**

Service Delivery Management: Ensures that IT services are delivered in accordance with agreed-upon levels of service. It covers the management of IT service delivery processes to meet business needs reliably and efficiently.

Risk Management: Involves identifying, assessing, and managing IT-related risks to ensure they are within acceptable levels. It ensures that the organization's risk exposure is managed and that business continuity is maintained.

Resource Optimization: Focuses on optimizing the use of IT resources, including infrastructure, applications, information, and people. It ensures that IT resources are used efficiently and effectively to support business objectives.

5. **Monitor and Evaluate**

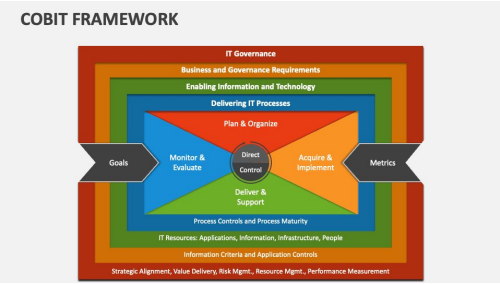
Performance and Conformance Monitoring and Reporting: Deals with the monitoring and reporting of IT performance against agreed-upon metrics and compliance with policies, standards, laws, and regulations. It ensures that IT is delivering value to the business and meeting its obligations.

Internal Control and Compliance: Focuses on the establishment and maintenance of internal controls to ensure reliability and integrity of information, compliance with policies and regulations, and effective and efficient operations.

Governance of IT Investments: Ensures that IT investments are aligned with business goals and deliver the expected return on investment. It involves the governance of IT investment decisions and the monitoring of IT investment performance.

COBIT

COBIT stands as a critical framework for aligning IT operations with business objectives, ensuring effective governance, and fostering a culture of continuous improvement and compliance.



Risk analysis and management

Risk Analysis and Management watches for issues concerning data integrity and maintaining operational security and stability. This process involves:

- **Identifying potential threats** that could impact IT systems, from cyber-attacks and data breaches to system failures and natural disasters.
- **Assessing the risks** by evaluating the likelihood of these threats occurring and their potential impact on the organization. This assessment helps in understanding the severity and extent of each risk.
- **Prioritizing risks** based on their severity and impact, ensuring that resources are allocated effectively to address the most critical vulnerabilities first.

Risk Management in COBIT and TOGAF

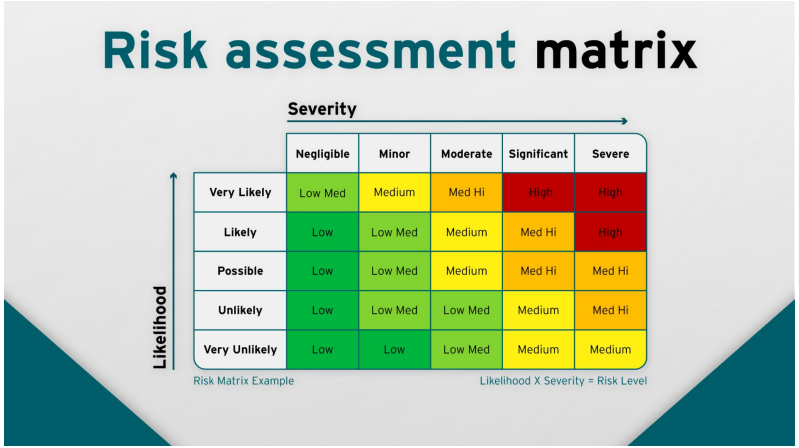
COBIT's Risk Management: Emphasizes aligning IT risk management practices with **business objectives**, ensuring an integrated approach to managing IT-related risks. COBIT provides a framework for establishing **controls and measures to mitigate identified risks**, monitoring and continually improving risk management practices.

TOGAF's Risk Management: Focuses on identifying and managing risks throughout the enterprise architecture development process, using the ADM. **TOGAF encourages proactive risk identification and mitigation strategies** to be embedded within the architecture planning and implementation phases.

Risk appetite

Risk appetite refers to the amount and type of risk an organization is willing to accept in pursuit of its objectives. Establishing clear risk appetite strategies is crucial for guiding decision-making and risk management practices. Key components of developing these strategies include:

- **Defining Risk Appetite**: the organization's willingness to take on risk, considering both quantitative and qualitative factors. It involves setting clear thresholds for different types of risks, such as operational, financial, and reputational risks.
- **Aligning with Business Objectives**: Risk appetite aligns with the organization's strategic goals and objectives. This alignment helps in prioritizing resources and efforts towards managing risks that could impede achieving these goals.
- **Developing Risk Tolerance Levels**: While risk appetite defines the overall willingness to accept risk, risk tolerance specifies acceptable variations in outcomes. Establishing tolerance levels for key performance indicators helps in monitoring and controlling risks.



Risk management strategies

- **Avoid:** Eliminating the risk by discontinuing the activities that generate it. This approach is suitable for high-level risks that surpass the organization's risk appetite and could significantly impact business objectives. *(not that much related to software)*
- **Mitigate:** Implementing measures to reduce the likelihood or impact of the risk. Mitigation strategies include enhancing security protocols, improving operational procedures, or adopting new technologies. This approach is commonly used for risks that are within the organization's risk appetite but still require management to ensure minimal impact.
- **Transfer:** Shifting the risk to a third party, such as through insurance or outsourcing. This strategy is particularly relevant for risks where the organization might not have direct control or where transferring the risk is more cost-effective than in-house mitigation.
- **Accept:** Recognizing that the risk is within the organization's risk appetite and deciding not to take specific actions to alter its likelihood or impact. This decision is often reserved for low-level risks or when the cost of other strategies outweighs the benefit.

Connecting IT Governance and Software Architecture

- **Strategic Alignment:** Software architecture bridges the strategic objectives outlined by IT governance with the practical implementation of technology solutions. By adhering to overarching standards and frameworks like TOGAF, software architects ensure that systems are designed in a way that supports business goals.
- **Risk Management:** Architectural frameworks facilitate risk management by providing guidelines for secure, reliable, and resilient system design. Software architecture, guided by these frameworks, addresses risks at the design level, incorporating security, scalability, and compliance into the foundational structure of IT solutions.

The Role of Software Architecture in Organizational Success

Software architecture must ensure that the designed systems are aligned with the broader business strategy and can adapt to changing needs and technologies.

> The architecture defines **how systems interact**, how **data flows** between components, and how **external and internal services are integrated**, **directly impacting** the organization's ability to innovate and compete.

Some benefits of proper alignment:

- **Innovation and Agility:** A well-defined software architecture allows for quicker adaptation to new market demands and technologies. It enables organizations to pivot more easily and introduce innovations, maintaining a competitive edge.
- **Operational Efficiency:** By defining clear protocols, interfaces, and data interchange formats, software architecture improves interoperability and reduces redundancies, leading to increased operational efficiency and reduced costs.
- **Quality and Performance:** Architectural decisions directly affect the quality attributes of a system, such as its performance, reliability, and scalability. A focus on architecture helps in building systems that meet performance expectations and can scale with the organization's growth.

Integrating Software Architecture with IT Governance and Architectural Frameworks

The integration of software architecture within the IT governance and architectural framework context ensures that technology decisions are made with a clear understanding of their strategic impact. This integration:

- **Ensures Alignment:** By aligning software architecture practices with frameworks like TOGAF and governance models like COBIT, organizations can ensure that every architectural decision supports the strategic vision and complies with governance policies.
- **Facilitates Communication:** Architectural frameworks provide a common language and set of standards that facilitate communication between business stakeholders, IT leaders, and developers, ensuring that all parties have a clear understanding of the system's architecture and its business implications.
- **Promotes Best Practices:** Adopting architectural frameworks within the governance model encourages the use of industry best practices in software design and development, leading to higher quality systems that are more aligned with business needs.

Risk Matrix Analysis

20 minutes
5 min class discussion

https://docs.google.com/document/d/1Lhp4QIDJjW3arF4sXZPrcHyWywxOHx1ffMXOGACQUAc/edit?usp=sharing

1.4 - The soft skills of Software architects

Beyond technical expertise, software architects must possess a robust set of soft skills to effectively lead projects and drive organizational success.

These skills include communication, leadership, negotiation, problem-solving, and adaptability.

Mastering these soft skills enables architects to **bridge the gap** between technical teams and business stakeholders, ensuring **architectural visions are aligned** with business goals and **effectively implemented**.

Mastering Communication and Leadership

Effective Communication is critical for articulating technical concepts to non-technical stakeholders and ensuring clear understanding within development teams.

Leadership involves inspiring and guiding teams towards achieving architectural goals, managing conflicts, and fostering a collaborative environment.

Strategies for Improvement: Engage in cross-departmental projects, practice public speaking, and participate in leadership workshops or mentorship programs.

Negotiation and Problem-Solving

Negotiation is key in achieving consensus among diverse stakeholder interests, balancing technical constraints with business needs.

Problem-solving involves analytical thinking to address complex architectural challenges with innovative solutions.

Strategies for Improvement: Attend negotiation training, participate in design thinking workshops, and tackle real-world problems in community hackathons. Help your classmates and fight for your ideias!

The architect’s CI/CD

Continuous Improvement … Continuous Deployment of a new YOU

Continuous learning and improvement are vital for staying ahead in the rapidly evolving field of software architecture. It may include …

- **Practicing Regularly:** Stay hands-on with coding and architectural design to maintain technical proficiency.
- **Participating in Open Source Projects:** Contribute to open source projects to gain exposure to diverse architectural patterns and practices.
- **Writing Blogs and Articles:** Share knowledge and insights on architectural best practices, emerging trends, and personal experiences.
- **Teaching Others:** Teach courses, lead workshops, or mentor juniors to refine your understanding and communication skills.
- **Networking:** Join professional networks, attend conferences, and participate in forums to exchange ideas with peers.

Bibliography

- (Garlan, 1992) – David Garlan, Mary Shaw. “An Introduction to Software Architecture”
- (Mens, T. , Demeyer, S. , 2008) – Tom Mens and Serge Demeyer, “Software Evolution”, ISBN: 978-3-540-76440-3, http://www.springerlink.com/content/978-3-540-76439-7/#section=200232&page=1
- (Bass et al, 1998) – L. Bass, P. Clements, R. Kazman (1998). Software Architecture in Practice. Reading, MA: Addison Wesley Longman, Inc.
- (Garlan et al, 1993) – D. Garlan, M. Shaw (1993). "An Introduction to Software Architecture." In V. Ambriola, G. Tortora, eds., Advances in Software Engineering and Knowledge Engineering, Vol. 2, pp. 1—39. New Jersey: World Scientific Publishing Company.