Learn. To Change.

REFRESH & EXAM PREPARATION FOR THE BCS PRACTITIONER CERTIFICATE IN ENTERPRISE AND SOLUTION ARCHITECTURE

House Keeping

Course times

Breaks / meals

Rooms

Messages

Security

Safety

Course Administration

Course times

Reactions

Chat box

Sound
(mute / unmute)

MS Whiteboard

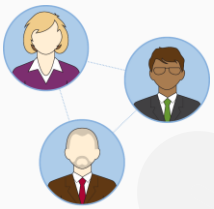
Documents / manuals

Mimeo

About You

Your:

- Name
- Company
- Job responsibility / role

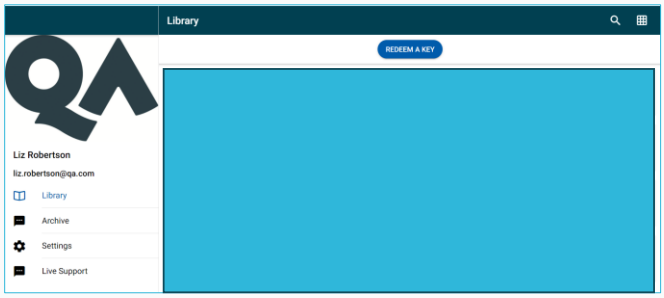


Your experience in:

- IT in general
- Architecture

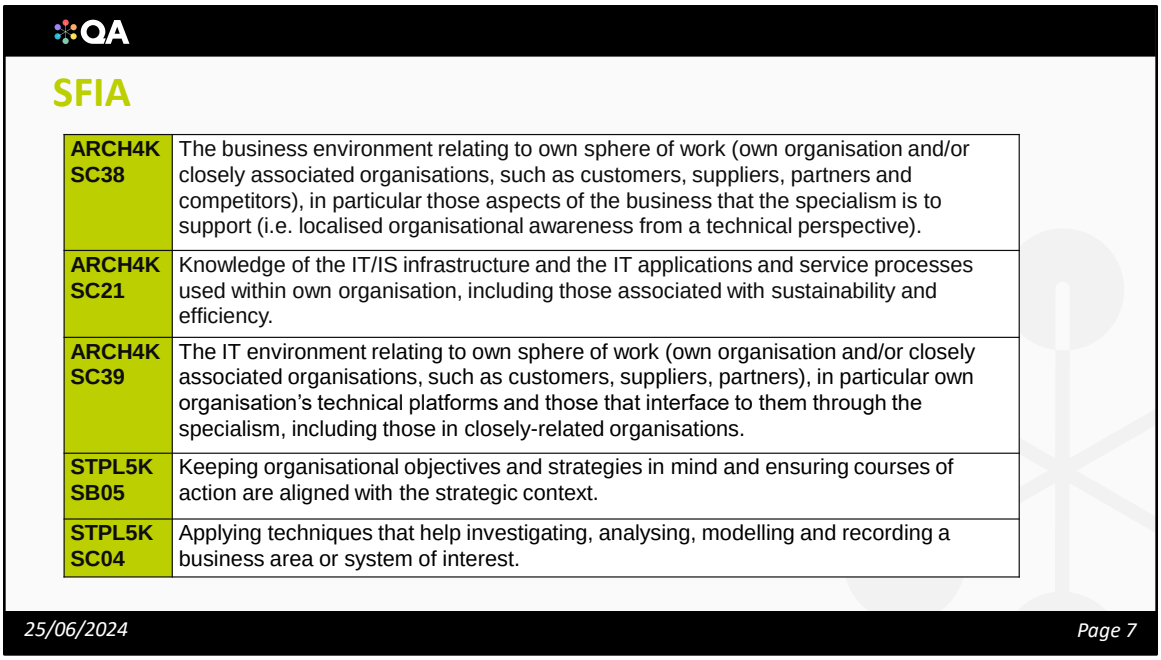
What do you want to achieve from this course?

The Course Materials



Create an account: https://mimeo.digital
Redeem your key: XXXXX

BCS PRACTITIONER CERTIFICATE IN ENTERPRISE AND SOLUTION ARCHITECTURE



This Certificate is aligned with the SFIA activites listed above.

ARCH Solution architecture

Developing and communicating a multi-dimensional solution architecture to deliver agreed business outcomes.

Solution architecture — English (sfia-online.org)

STPL Enterprise and business architecture

Aligning an organisation's technology strategy with its business mission, strategy, and processes and documenting this using architectural models.

Enterprise and business architecture — English (sfia-online.org)

QA

Syllabus Key Topics

Syllabus Area	Syllabus Weighting
1. Architecture Domains and Frameworks	15
2. Architecture and the Organisation	15
3. Professionalism in Architecture	20
4. Corporate Governance	10
5. Architectural Process, Tools, and Vision	25
6. Quality Assurance	15

Explore the key topics within the syllabus and consider the weightings – this is an indicator of the percentage of questions/available marks in each area of the exam.

QA

Syllabus

1. Architecture Domains and Frameworks

1.1 Discuss the role of architecture in an organisation.
1.2 Discuss the different architecture domains and the relationships between them.
1.3 Discuss the key architecture frameworks.
1.4 Discuss how industry standards are relevant to architecture.

2. Architecture and the Organisation

2.1 Explain the role of architecture to other business areas.
2.2 Discuss use of the business change lifecycle as an input for architecture.
2.3 Discuss the business needs and the intended solution.

3. Professionalism in Architecture

3.1 Discuss compliance with professional standards.
3.2 Discuss types of stakeholders.
3.3 Explain specific stakeholder needs.
3.4 Discuss how to influence others.
3.5 Apply a practical customer focus.
3.6 Discuss the roles and dynamics of a successful team.
3.7 Discuss social, ethical and economic concerns to the role of an architect.

Explore the learning objectives included within the syllabus.

QA

Syllabus

4. Corporate Governance

4.1 Explain corporate governance and the impact it has on the role of an architect.
4.2 Discuss key architecture governance concepts, roles, and artefacts.
4.3 Explain the importance of risk management to the role of an architect.

5. Architectural Process, Tools, and Vision

5.1 Discuss the practical application of the solution architecture framework.
5.2 Discuss three varieties of the Software Development Lifecycle (SDLC).
5.3 Discuss the design of a target solution architecture.
5.4 Discuss the use of common architecture models.
5.5 Discuss the use of common information systems modelling techniques.
5.6 Explain different types of business requirements.
5.7 Explain a method of preparing a gap analysis.
5.8 Discuss a business case for presentation.

6. Quality Assurance

6.1 Discuss the use of a range of quality assurance tools or techniques.
6.2 Discuss the principle of Separation of Concerns (SoC).
6.3 Explain change management tools and processes.
6.4 Discuss opportunities for continuous improvement.

Explore the learning objectives included within the syllabus.

The Course Schedule

DAY 1 09:00 - 15:30

0.0 Introduction and Definitions

1.1 Refresh on Architecture Domains and Frameworks
1.2 Practice Exercises

2.1 Refresh on Architecture and the Organisation
2.2 Practice Exercises

The Course Schedule

DAY 2 09:00 - 16:30

3.1 Refresh on Professionalism in Architecture
3.2 Practice Exercises

4.1 Refresh on Corporate Governance
4.2 Practice Exercises

5.1 Refresh on Architectural, Processes, Tools and Vision
5.2 Practice Exercises

The Course Schedule

DAY 3 09:00 - 15:30

6.1 Refresh On Quality Assurance
6.2 Practice Exercises

7.1 General Practice Exercise
7.2 Questions and Wrap Up

The Approach for the Training and the Assessment 1

The BCS have designed this qualification to enable you demonstrate your skill and capabilities in applying concepts of enterprise and solution architecture in practice.

It is not a standard training course. It is an opportunity for you to review the underlying frameworks, models and approaches that are generally recommended and how you apply these.

It is about how you act as an architect in day-to-day work and provides you with an opportunity to consider how you can best construct present and communicate architectures to different stakeholders.

The subsequent BCS assessment will test this ability to understand and analyse architecture change scenarios and apply knowledge gained in this and other courses and your personal work.

The Approach for the Training and the Assessment 2

This course does not provide :

All the knowledge needed. You are expected to already know and be able to apply most of it.

The course does provide:

A refresher of some of the key knowledge and approaches you will need.

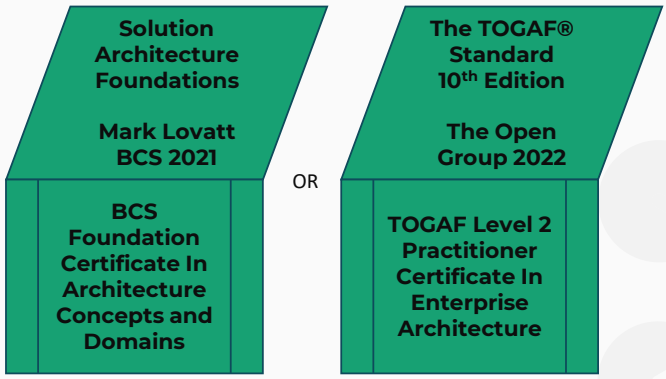
Information and exercises for you to prepare for the discussive BCS assessment of some 50 minutes to demonstrate this knowledge and its application

You will be doing significant personal preparation work during the course and need additional personal study and preparation for the discursive assessment which you need to personally book before or after this course.

Changes to the BCS Syllabus for Version 5.0

- Provides up-to-date content
- Provides a vendor-neutral view of IT architecture
- Provides an understanding of resources available for development (e.g. bodies of knowledge, national institutes, etc.)
- Works in conjunction with other vendor-specific qualifications and IT architecture frameworks
- Places a higher emphasis on professionalism and soft skills
- Introduces new sections on corporate governance and risk management
- Focuses on providing a broader organisational perspective
- Introduces specific sections on industry standards
- Provides a more focussed view of architectural processes, tools, and vision

The Course Source Materials and Pre-requisites



The Course Goal and Objectives

To prepare you to present your understanding of enterprise and solution architecture as it is applied in practice in your work for the BCS certification assessment.

This includes:

- The roles and relationships between architecture levels and domains.
- The required skills and professional behaviours of an architect.
- Tools, frameworks and artefacts used in architecture.
- Implementing and upholding governance and quality assurance.

The BCS Assessment

50 MINUTES – SCENARIO BASED QUESTIONS

You will be presented with 6 scenarios and questions to:

- Assess content in relation to a variety of different scenarios.
- Recognise multiple approaches to different situations.
- Include criteria to assess your ability to communicate.

You will have about 8 minutes to provide answers to each of the questions. There may be supplementary questions depending on the progress of your responses.

There may or may not be specifically right or wrong answers but there will be better or worse answers.

You should apply the content of the learning, the approaches and models, and frameworks that you have studied to the situation and issues identified in the assessment scenarios.

The Course Slides

- The course slides are a mix of those:
- Supplied by the BCS that provide the shape and extent of the materials required.
- Supplied by QA that provide more information on the underlying content that is required and any relevant extension material.

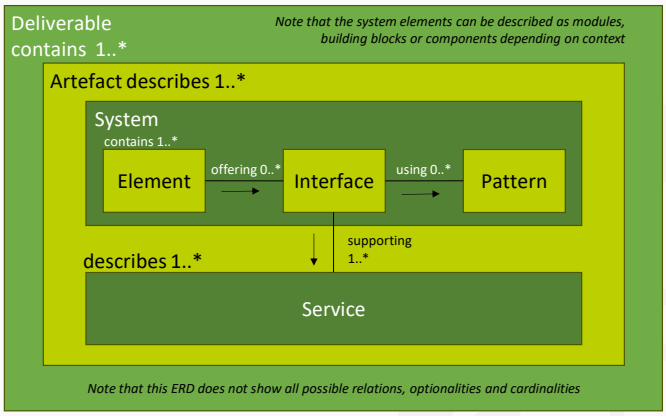
Defining Terms

- **Architecture description**
- **Enterprise** (Solution / Segment-Domain / Capability-Solution / Service)
- **Solution**
- **Architecture**
- **Architect**
- **Architecting**
- **Architecture Framework**

Architecture Description

Work products are used to express an architecture.

An enterprise, segment/domain or capability/solution architecture can be documented by some set of views of work products describing interacting building blocks, in artefacts, aggregated into deliverables:



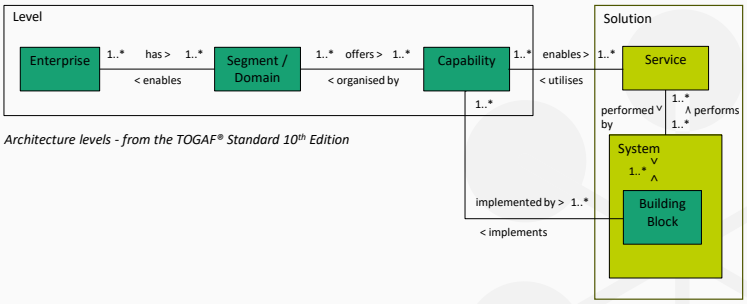
Enterprise

A usually managed grouping of resources and capability organised to implement a mission and achieve shared goals and objectives and their associated outcomes (for one, part of, or more than one, organisation).

Solution

A solution is a targeted set of capabilities implementing one or more building blocks in a specific context.

A solution can be relevant to the scope of an enterprise, a specific segment / domain, a specific service and / or stand on its own as set of capabilities.



Architecture levels - from the TOGAF® Standard 10th Edition

Architecture

The fundamental concepts or properties of a system in its environment; and embodied in its elements, relationships, and in the principles of its design and evolution.

From: ISO /IEC/IEE 42010:2011

The purpose of the architecture definition process is to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views.

From: ISO/IEC IEEE 15288:2015.

It is essentially the inherent elements, structure and behaviour of a system or set of systems which may be present because of design and / or evolution in line with t set of goals, objectives, requirements and constraints.

Architect

A person performing the role of an architect.

- Organises deliverables, artefacts, and building blocks/components into various groupings.
- Evolves resources and building blocks/components to deliver targeted systems/sub-systems and services.

at the relevant level of composition and detail enterprise – segment / domain – capability / solution – system

Architecting

The activity of conceiving, defining, expressing, documenting, communicating, assuring, certifying, maintaining and improving an architecture.

- Organising deliverables, artefacts, and building blocks/components into various groupings
- Evolving resources and building blocks/components to deliver targeted systems / sub-systems and services at the relevant level of detail and composition.

at the relevant level of composition and detail enterprise – segment / domain – capability / solution – system

Architecture Framework

A foundational structure, or set of structures, for developing, evolving and maintaining a broad range of different architectures.

It should include a method for describing both a baseline and target state of the enterprise, in terms of a set of building blocks, for showing how the building blocks fit together, and planning the evolution from baseline to target states.

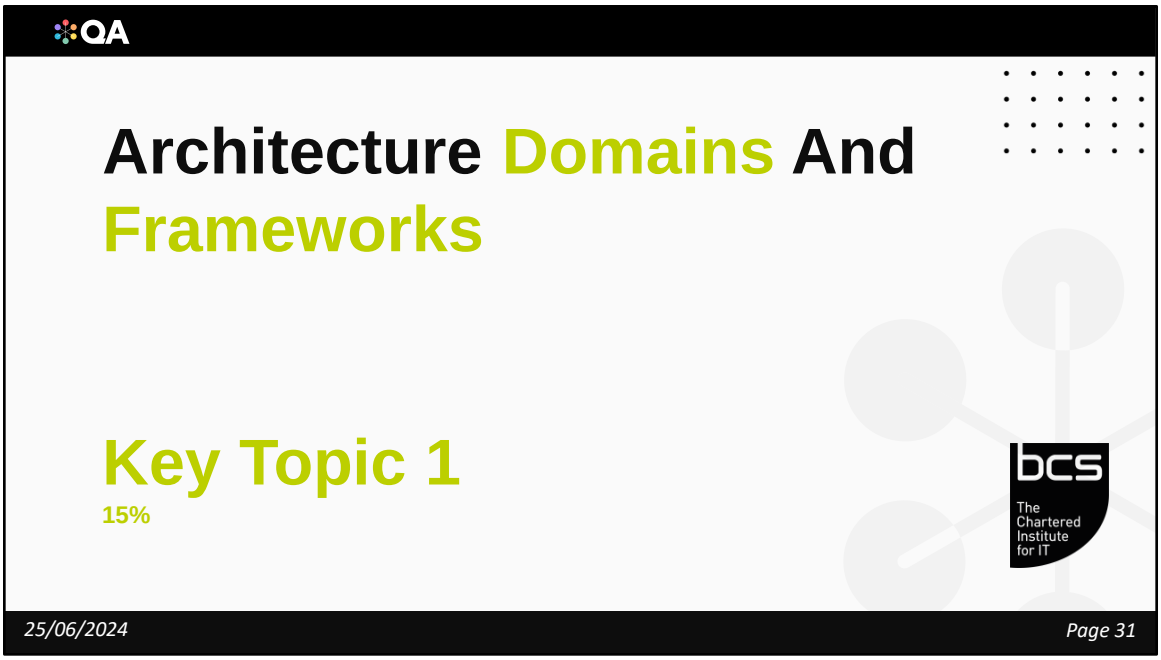
A framework is typically tailored to meet the specific needs of an organisation. Tailoring of the framework should establish a set of tools and a common vocabulary for applying the framework.

at the relevant level of composition and detail enterprise – segment / domain – capability / solution – system

The end of the introduction – Section 0

Section 01

Architecture Domains And Frameworks



Elements to be addressed:

Architecture Domains and Frameworks

Discuss the role of architecture in an organisation.
Discuss the different architecture domains and the relationships between them.
Discuss the key architecture frameworks.
Discuss how industry standards are relevant to 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”.

ISO 42010

The course schedule

DAY 1 09:00 - 15:30

0.0 Introduction and Definitions

1.1 Refresh on Architecture Domains and Frameworks

1.2 Practice Exercises

2.1 Refresh on Architecture and the Organisation

2.2 Practice Exercises

Section 1 - Architecture Domains and Frameworks

1. **The role of architecture in an organisation.**
2. Different architecture domains and the relationships between them.
3. Key architecture frameworks.
4. How industry and professional standards are relevant to architecture.

QA

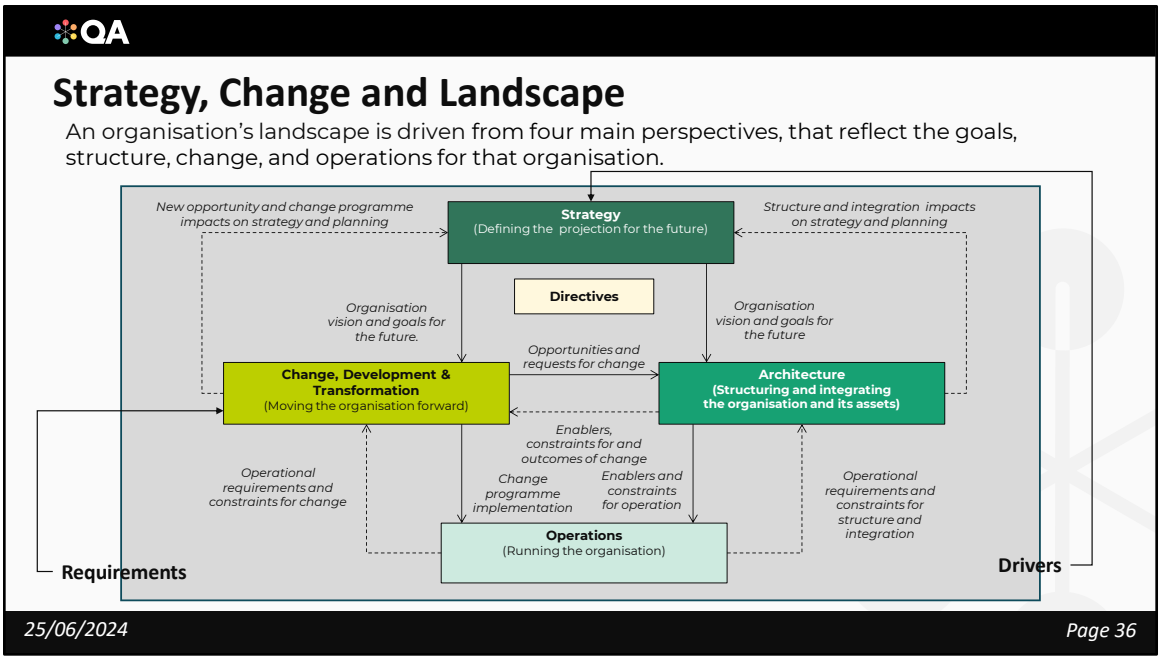
Architecture In The Organisation

- Strategic change, problem solving, legal requirements
- Purpose and scope of role
- The need for architecture
- The benefits of architecture

Consider:

What is the purpose of architecture?
How does architecture interact with the business?
What is architecture responsible for?
Does the size/type of organisation affect the architecture?
How does architecture benefit an organisation (e.g. responsive to change/agile)

You should be able to discuss what architecture is, and the reason it is required in organisations. You should be able to discuss the drivers for architecture may be both internal and external, to describe these drivers and the effect they have on specific domains, and to explain the purpose and scope of the role of an architect, in a given domain.



Strategy defines mission, drivers, goals and objectives with expected change areas of business capability, value steams, information sets and potential courses of action.

Drivers are the causes of consideration for change that may come from inside or outside of the organisation. Drivers may be relatively general and high level while requirements are at a greater levels of details (SMART).

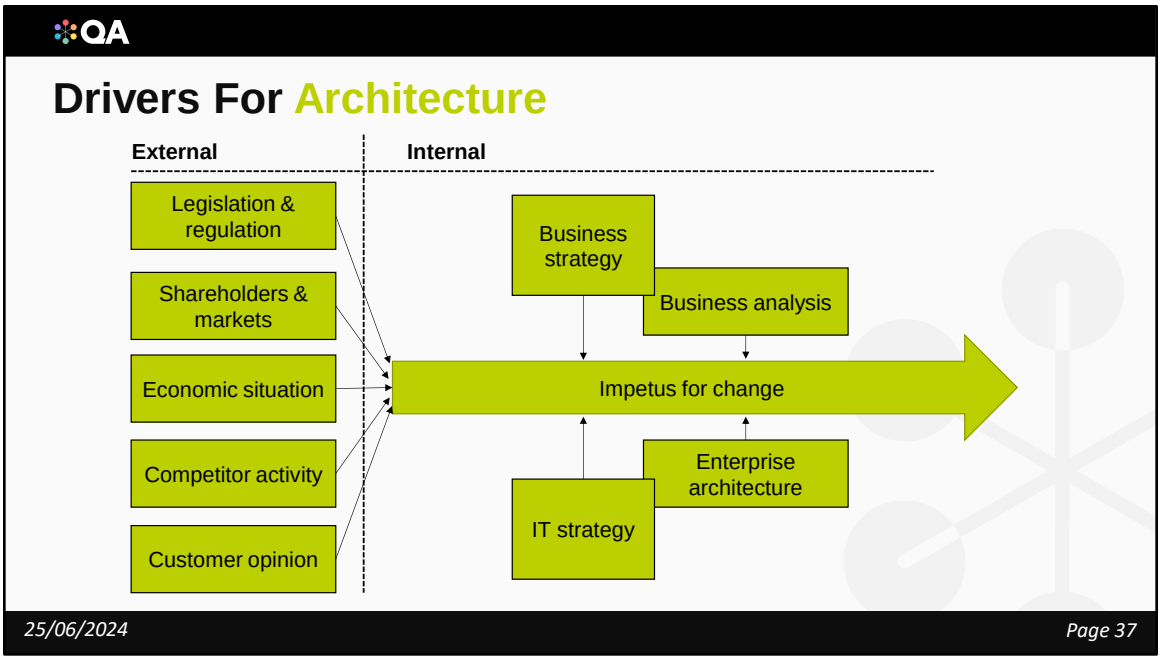
Directives (Principles, Policies, Standard, Best Practices and Business Rules) provide a bridge between strategy, the organising architecture and the solution implementation.

Architecture provides a description of the building blocks that make up the resources of the enterprise to implement its needed capabilities organised into implemented systems and services and manged by some set of domains providing control and management of those capabilities and their use.

Change organises the specific units of specification, development and implementation that will cause the operation of the organisation to be changed in line with its strategy and specific requirements.

Operations implements and manages the executing processes and entities that perform the work of the organisation delivering outcomes that satisfy the organisation's goals, objectives and requirements.

Requirements (things that are needed) may come from inside or outside of the organisation and are an input to all of these organisational elements providing the specific detailed targeted inputs and outcomes that implement the strategy. Requirements can be identified from any relevant stakeholder associated with a change. Requirements should be in line with agreed goals and objectives and directly feed specifically into change activity.



External –Consider the types of changes a market or economy could experience and the impact of not changing to comply with legislation or regulations.

Internal – Consider the types of changes that could happen in a business or IT strategy and how business analysis could create a need for architecture, including the link between business strategy, analysis and architecture.

The Purpose and Scope of Architecture

Purpose:

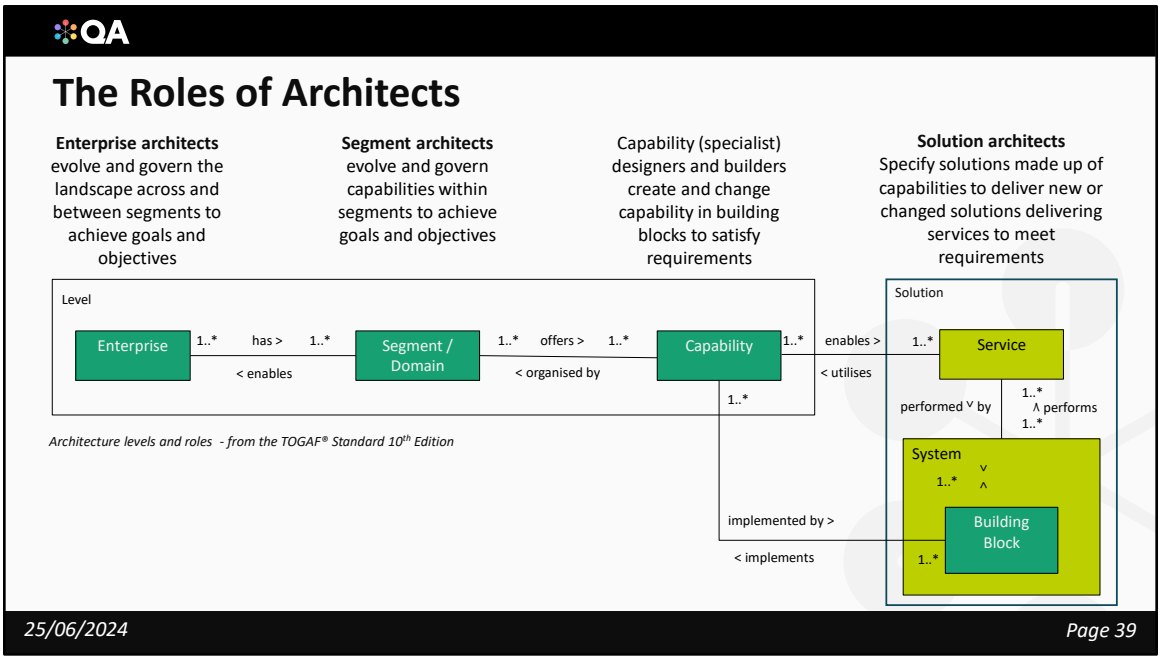
"The purpose of architecture is to optimize across the often-fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy."

- To enable effective change.
- To ensure that an enterprise has the resources, and elements needed to achieve its strategy and operational goals and objectives and delivered the desired outcomes.

Scope:

To work at the relevant level of composition and detail, enterprise – segment/domain – capability/solution – system

From – The TOGAF® Standard– Introduction and Core Concepts - Section 1.1



The architect takes on responsibility for organising, specifying and documenting, communicating and assuring services, systems, sub-systems, building block structure and interaction, to enable change activity to take place in a managed fashion across an enterprise in line with its current operational and strategic targets and specific requirements. They also provide leadership and knowledge for the utilisation of technical potential and change to deliver business benefit.

Consider what architecture is, and the reason it is required in organisations. The drivers for architecture may be both internal and external, and you should be able to describe these drivers and the effect they have on specific domains.

The Need for Architecture

"Furthermore, a good Enterprise Architecture enables you to achieve the right balance between business transformation and continuous operational efficiency. It allows individual business units to innovate safely in their pursuit of evolving business goals and competitive advantage. At the same time, the Enterprise Architecture enables the needs of the organization to be met with an integrated strategy which permits the closest possible synergies across the enterprise and beyond."

"Acts as a source of knowledge that enables problems to be diagnosed and solutions designed using solution architecture."

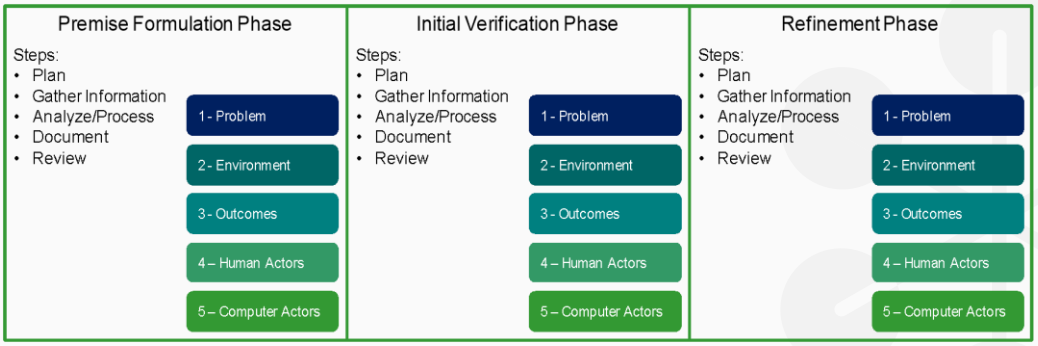
From – The TOGAF® Standard– Introduction and Core Concepts - Section 1.1

and

From – Solution Architecture Foundations – Mark Lovatt - BCS – 2022

Problem Solving & Legal Requirements

A business scenario is developed over several phases that formulate, verify, and refine a premise of the business requirements and constraints, to identify the **problem** (including technical, **legal** and regulatory) driving an effort. (From TOGAF 10th Edition Series Guide.)



The Benefits of Architecture

An effective Enterprise Architecture can bring important benefits to the organization.

These include:

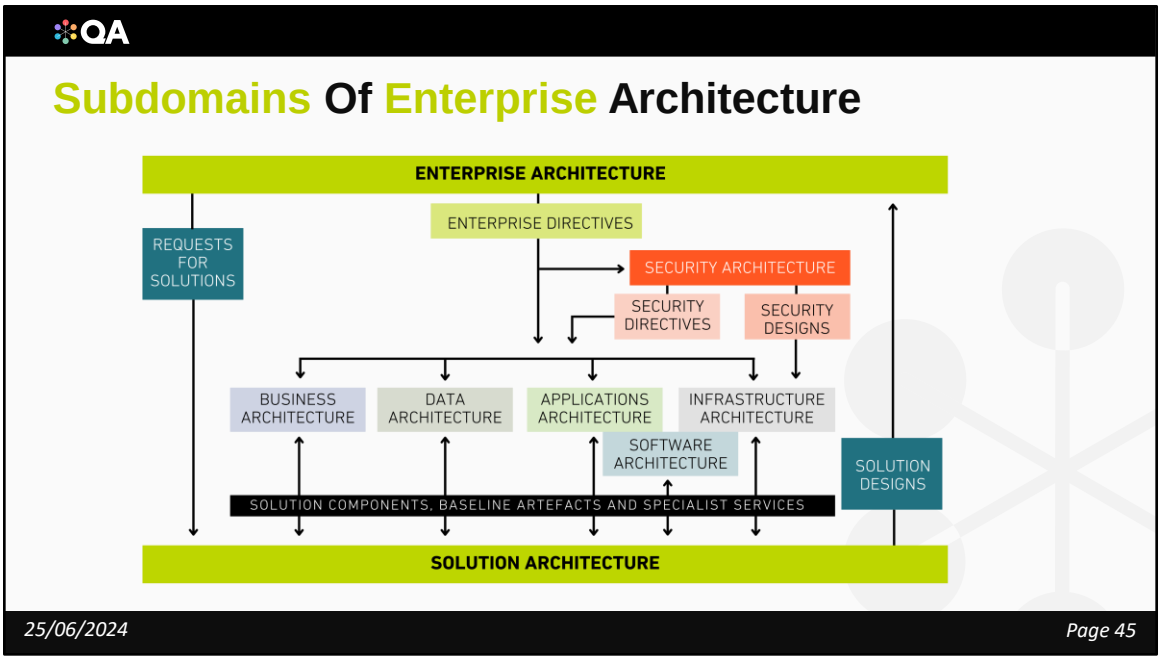
- More effective strategic decision-making by C-Level executives and business leaders:
- More effective and efficient business operations:
- More effective and efficient Digital Transformation and operations:
- Better return on existing investment, reduced risk for future investment:
- Faster, simpler, and cheaper procurement:

From – The TOGAF® Standard– Introduction and Core Concepts - Section 1.1

Section 1 - Architecture Domains and Frameworks

1. The role of architecture in an organisation.
2. **Different architecture domains and the relationships between them.**
3. Key architecture frameworks.
4. How industry and professional standards are relevant to architecture.

Domains refer to distinct areas of focus within the field of information technology architecture.



From - Solution Architecture Foundations - Mark Lovatt - BCS 2021

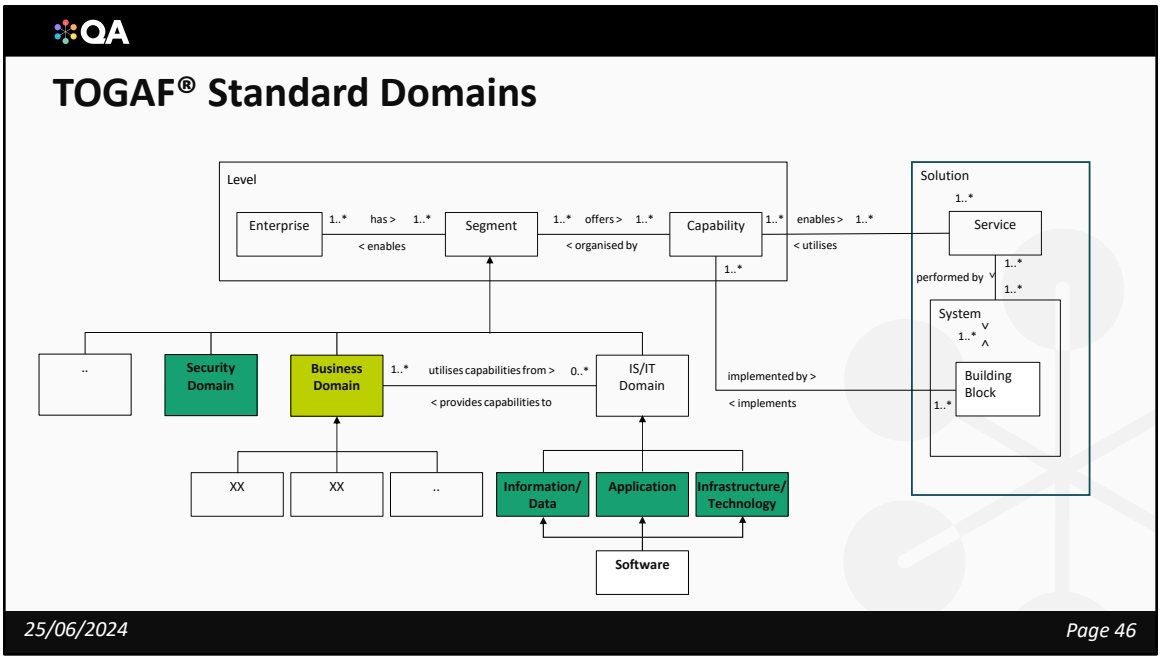
The diagram shows the BCS Solution Architecture source material presentation of the relationships between various architecture domains, showing flows of information, artefacts and services at all levels.

The BCS Solution Architecture based model identifies six initial domains Business, Security, Information, Application, Infrastructure and Software.

Consider:

What are the domains, and do they reflect your experience?
How and what information is shared between the domains?
How do you best communicate between stakeholders working in each domain at each level?
Is the hierarchy fixed or merely illustrative for consideration?
What would be the main activities and outputs of each domain and the dependencies and relationships between them.
How is communication between domains optimised

You should be able to discuss the role of each architecture domain and the dependencies and relationships between them and identify the main activities and purpose of each domain should be considered, as well as the role each plays in the organisation.



For those who have come from a TOGAF background, it is useful to identify how TOGAF presents the domains in comparison to the Solution Architecture Foundations perspective.

The diagram shows the TOGAF® Standard source material viewpoint presentation of the relationships between various architecture domains, showing flows of information, artefacts and services at all levels.

The TOGAF® Standard identifies four initial domains Business (any number of these) Data, Application and Technology (BDAT) with a cross-cutting viewpoint for Security.

Consider:

What are the domains, and do they reflect your experience?
How is information shared between the domains?
How do you best communicate between stakeholders working in each domain at each level?
Is the hierarchy fixed or merely illustrative for consideration?
What would be the main activities and outputs of each domain and the dependencies and relationships between them.

You should be able to discuss the role of each architecture domain and the dependencies and relationships between them and identify the main activities and purpose of each domain should be considered, as well as the role each plays in the organisation.

Section 1 - Architecture Domains and Frameworks

1. The role of architecture in an organisation.
2. Different architecture domains and the relationships between them.
3. **Key architecture frameworks.**
4. How industry and professional standards are relevant to architecture.

IT architectural frameworks are structured methodologies that provide guidelines, best practices, and principles for designing, planning, and managing complex systems and solutions.

QA

Architectural Frameworks

Architecture Frameworks

- Zachman
- TOGAF
- NAF
- C4

Security Frameworks

- CyBOK
- SABSA
- NIST - CSF

There are many different architecture frameworks. The ones listed are some of the most familiar.

We provide several following slides to refresh your memory for these frameworks. We will not go through these in detail, rather we will progress fairly quickly through these and provide references for you to review when working through the workbook and preparing for the BCS assessment.

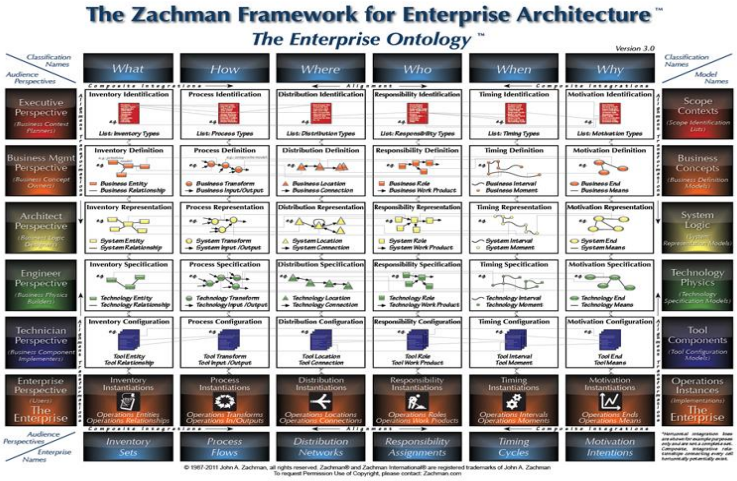
Consider:

The purpose of Architecture Frameworks
How Frameworks contribute to better process and ways of working
Your personal experience of adhering to frameworks
How the following frameworks differ
Their differences and limitations

You should be able to discuss the architecture and security frameworks below and be able to describe the application of these frameworks and how they impact architecture activity.

QA

ZACHMAN

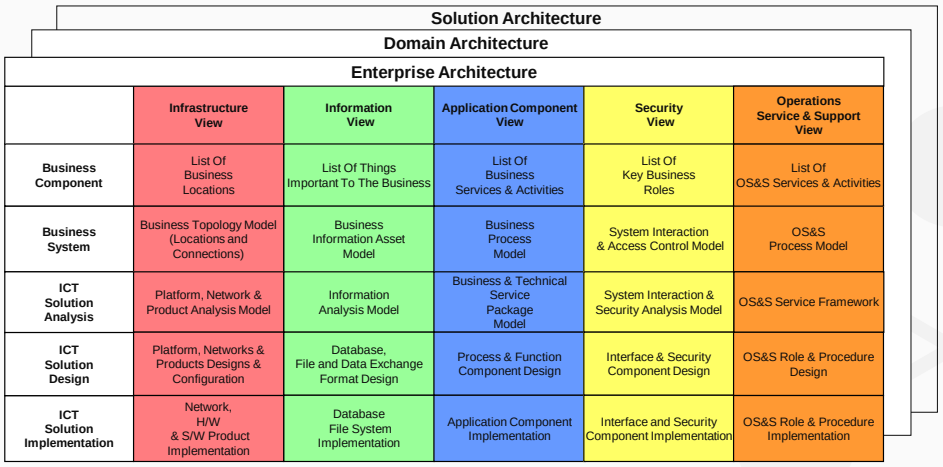


"The Zachman Framework is an enterprise ontology and is a fundamental structure for enterprise architecture which provides a formal and structured way of viewing and defining an enterprise. The ontology is a two-dimensional classification schema that reflects the intersection between two historical classifications. The first are primitive interrogatives: What, How, When, Who, Where, and Why. The second is derived from the philosophical concept of reification, the transformation of an abstract idea into an instantiation. The Zachman Framework reification transformations are: identification, definition, representation, specification, configuration and instantiation."

From Wikipedia

https://Zachman-feac.com

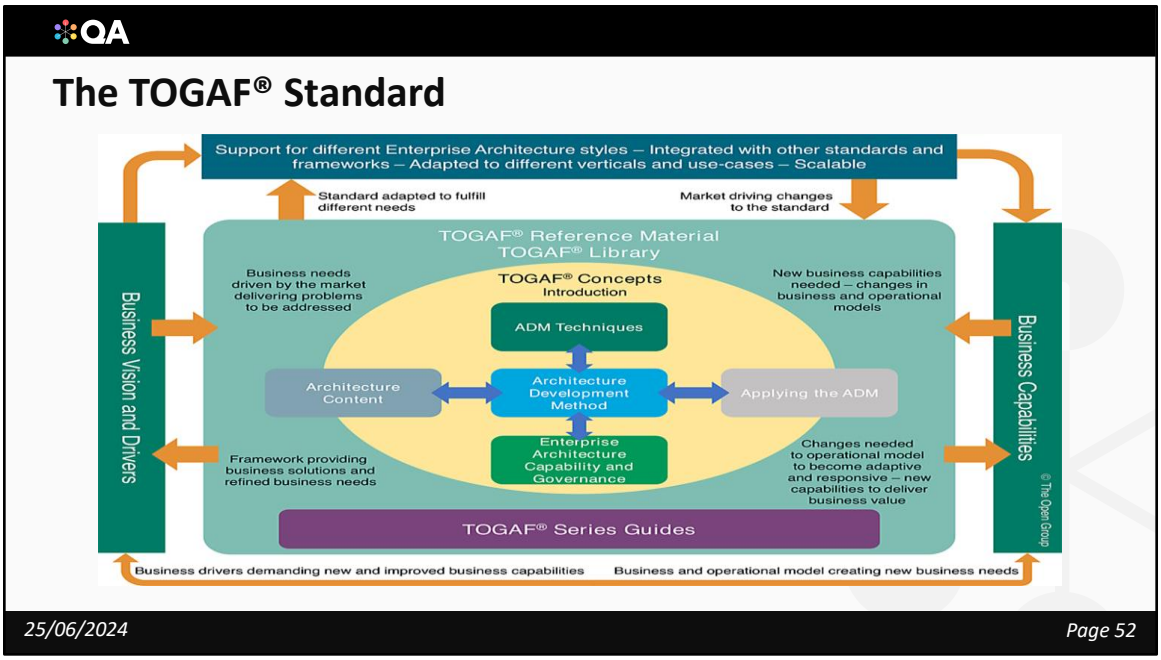
Example of Adapted ZACHMAN



	Infrastructure View	Information View	Application Component View	Security View	Operations Service & Support View
Business Component	List Of Business Locations	List Of Things Important To The Business	List Of Business Services & Activities	List Of Key Business Roles	List Of OS&S Services & Activities
Business System	Business Topology Model (Locations and Connections)	Business Information Asset Model	Business Process Model	System Interaction & Access Control Model	OS&S Process Model
ICT Solution Analysis	Platform, Network & Product Analysis Model	Information Analysis Model	Business & Technical Service Package Model	System Interaction & Security Analysis Model	OS&S Service Framework
ICT Solution Design	Platform, Networks & Products Designs & Configuration	Database, File and Data Exchange Format Design	Process & Function Component Design	Interface & Security Component Design	OS&S Role & Procedure Design
ICT Solution Implementation	Network, H/W & S/W Product Implementation	Database File System Implementation	Application Component Implementation	Interface and Security Component Implementation	OS&S Role & Procedure Implementation

An example of a customisation of the Zachman framework for an organisation aligning it to three perspectives of enterprise, domain and solution and naming of the rows and columns to a language used within a specific organisation. In this case it identified the specific models/representations that were used within the organisation.

In most cases it is likely that you will need to adapt/customise a specific architecture framework for your organisation's specific needs.



The TOGAF® Standard fundamental Content is presented as six free-standing documents:

Architecture Development Method - This document describes the TOGAF Architecture Development Method (ADM) — an iterative approach to developing an Enterprise Architecture.

ADM Techniques - This document contains a collection of techniques available for use in applying the TOGAF® approach and the TOGAF® ADM.

Applying the ADM - This document contains guidelines for adapting the TOGAF® ADM to address the specific style of architecture required in a practical context.

Architecture Content - This document describes the TOGAF® Content Framework and a structured metamodel for architectural artifacts, the use of re-usable Architecture Building Blocks (ABBs), and an overview of typical architecture deliverables.

Enterprise Architecture Capability and Governance - This document discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function

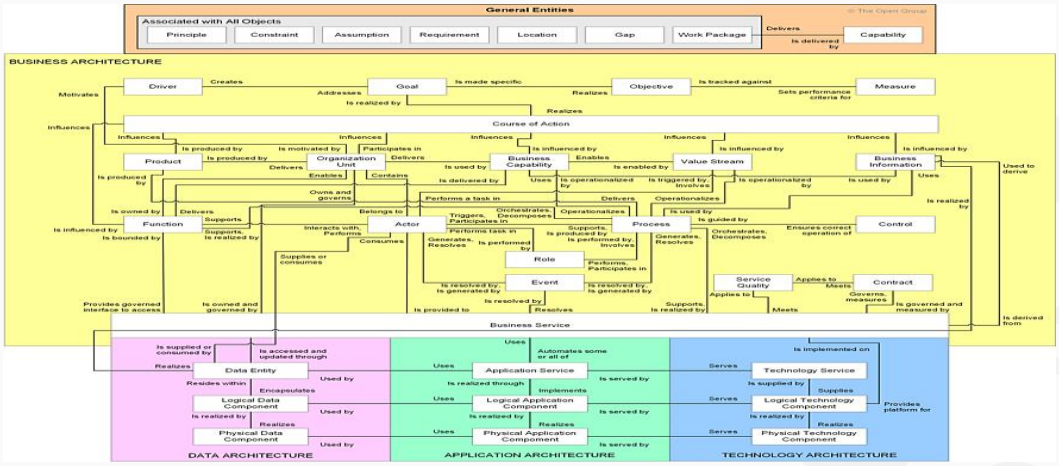
Besides the core framework content covered by the six documents explained above, the standard provides guidance to address specific concerns and use-cases through the TOGAF® Series Guides.

The TOGAF® Series Guides are designed to support more specific needs from practitioners who need further explanation or more detail than that provided in the core content.

Not all the TOGAF® Series Guides will be relevant in every situation. However, Enterprise Architects who are planning the deployment of the TOGAF® Standard should be aware of the guidance available.

The TOGAF® Standard, 10th Edition (90-Day Evaluation License) (opengroup.org)

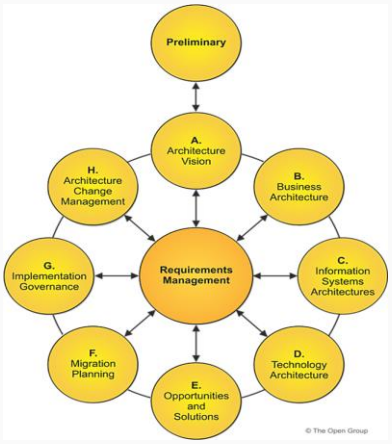
The TOGAF® (Metamodel)



To support development of the enterprise's metamodel, the TOGAF® Library includes a Foundation-level Core Enterprise Metamodel, detailed in the TOGAF® Standard - Architecture Content. It shows types of entity, and relationships between them, that are likely to be required in modeling most enterprises and provides a context for the artifacts suggested in the ADM.

The TOGAF® Standard, 10th Edition (90-Day Evaluation License) (opengroup.org)

TOGAF® 10th Edition (Architecture Development Method)



25/06/2024 *Page 54*

The TOGAF® Architecture Development Method (ADM) provides a tested and repeatable process for developing architectures. The ADM includes establishing an architecture framework, developing architecture content, transitioning, and governing the realization of architectures.

All of these activities are carried out within an iterative cycle of continuous architecture definition and realization that allows organizations to transform their enterprises in a controlled manner in response to business goals and opportunities.

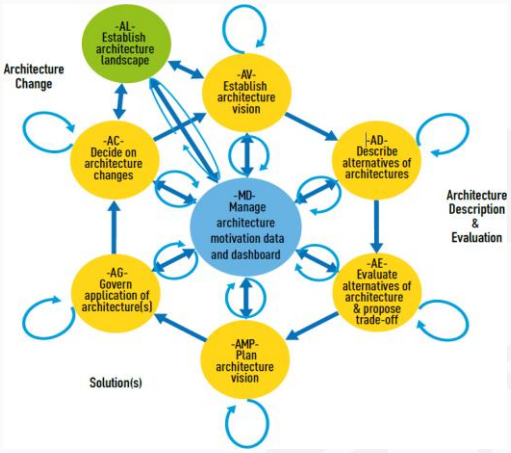
The TOGAF® Standard, 10th Edition (90-Day Evaluation License) (opengroup.org)

QA

NAF

The NATO Architecture Framework Version 4 (NAFv4) is to provide a standard for developing and describing architectures for both military and business use.

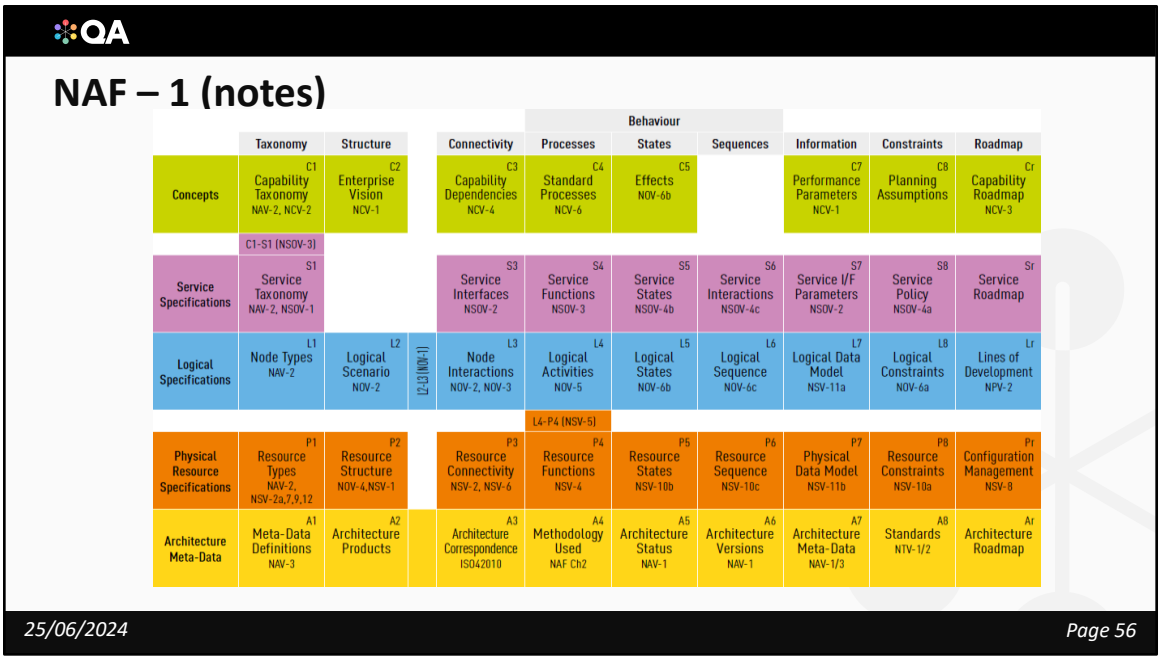
It provides a standardised way to develop architecture artefacts, by defining Methodology, Viewpoints and a Meta-Model



NATO Architectural Framework

The aim of the NATO Architecture Framework Version 4 (NAFv4v2) is to provide a standard for developing and describing architectures for both military and business use. It provides a standardised way to develop architecture artefacts, by defining a Methodology (how to develop architectures and run an architecture project), Viewpoints (conventions for the construction, interpretation and use of architecture views for communicating the enterprise architecture to different stakeholders), a Meta-Model (the application of commercial meta-models identified as compliant with NATO policy), and a Glossary, References and Bibliography.

NATO - NATO Architecture Framework, Version 4
NAFv4_2020.09.pdf (nato.int)



NAF details the viewpoints that make up NAFv4 v2. These are presented as a grid representation to organize the various subjects of concern (rows) and aspects of concern (columns), logically and consistently to aid architects, as shown below:

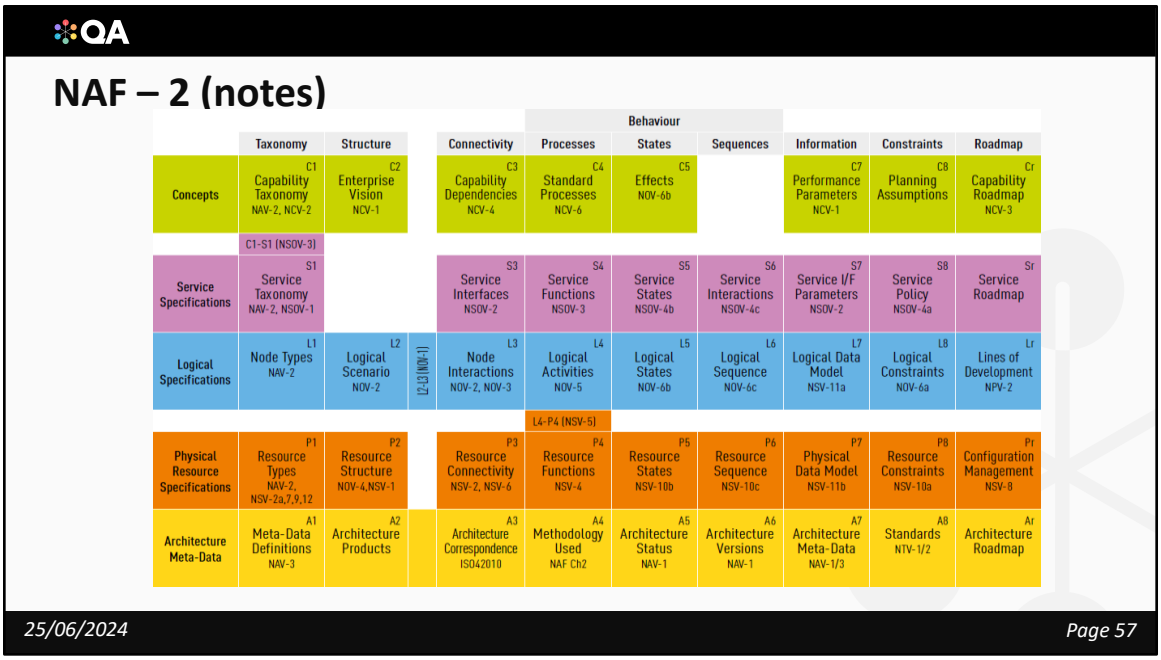
The NAF Approach to Architecture

Architectures are developed to support strategic planning, transformation, and various types of analyses (i.e., gap, impact, risk) and the decisions made during each of those processes. Additional uses include identifying capability needs, relating needs to systems development and integration, attaining interoperability and supportability, and managing investments. The following describes architecture usage at two different levels4:

Enterprise Level – architectures, particularly federated architectures, are used at the enterprise level to make decisions that improve: • human resource utilization, • deployment of assets, • investments, • identification of the enterprise boundary (external interfaces) and assignment of functional responsibility, and • structuring the functional activities in terms of projects.

Project Level – architectures are used at the project level to identify capability requirements and operational resource needs that meet business objectives. Project architectures may then be integrated to support decision making at the enterprise level. Architectures facilitate decision making by conveying the necessary information. Setting architectures within the enterprise context ensures complete, actionable information for more reliable decisions. The following describes architecture data usage for different types of decisions:

Portfolio management – identifies objectives and goals to be satisfied with regards to owned assets (capabilities and systems) and processes to be governed. Capability and Interoperability Readiness– Assesses capabilities and their implementation (systems, platforms, services and aggregated solutions) against needs and their net-readiness to identify gaps in interoperable features.



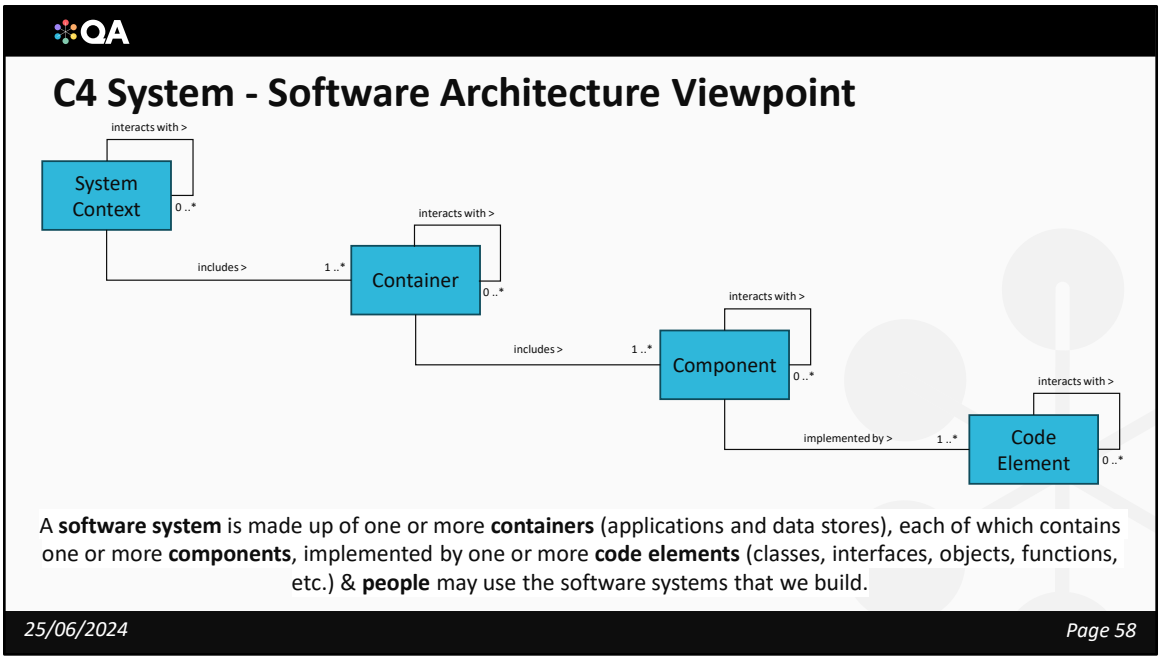
Operational Concept Planning – Examines how various mission participants, processes, roles, responsibilities, and information need to work together, to recognize potential problems that may be encountered, and to identify quick fixes that may be available to accomplish a mission.

Acquisition Programme Management and System Development – Expresses the plan and management activities to acquire and develop system concepts, design, and implementation (as they mature over time), which enable and support operational requirements and provide traceability to those requirements. This process must be compliant with the Enterprise objective and operational requirements. It refines operational analysis, performs system analysis, and improves both materiel and non-materiel solution analysis.

Modelling and Simulation – Modelling and simulation techniques can be used in order to assess the business and mission analysis. For example, in the military context through the implementation of mission threads5 and scenarios6, thus providing an environment for thorough testing of identified use cases.

Interoperability between Architectures – Architectures must not be produced for the sake of architectures themselves, but as a means to achieve higher level enterprise objectives (i.e. objectives in NATO). Architecture related processes should be seen as a technique for managing complexity rather than activities to produce models. A common set of architecture processes, such as those specified in NAF, is judged to be the best way of achieving success in the formation of a federation of systems approach. This concept is not only valid for NATO itself, but also between NATO, Nations and NATO's various partners (Non-NATO Nations, International Organizations (IOs) and Non-Government Organizations (NGOs).

NATO - NATO Architecture Framework, Version 4
NAFv4_2020.09.pdf (nato.int)



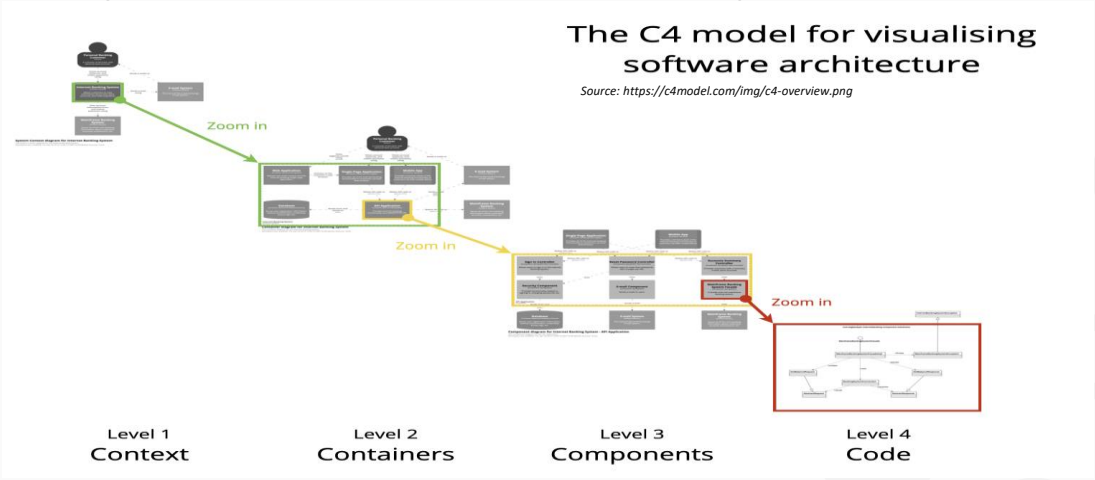
C4 Software Architecture Viewpoint

As an industry, we do have the Unified Modeling Language (UML), ArchiMate and SysML, but asking whether these provide an effective way to communicate software architecture is often irrelevant because many teams have already thrown them out in favour of much simpler "boxes and lines" diagrams. Abandoning these modelling languages is one thing but, perhaps in the race for agility, many software development teams have lost the ability to communicate visually.

The C4 model was created to help software development teams describe and communicate software architecture, both during up-front design sessions and when retrospectively documenting an existing codebase. It's a way to create maps of your code, at various levels of detail, in the same way you would use something like Google Maps to zoom in and out of an area you are interested in.

https://c4model.com/

C4 System - Software Architecture Viewpoint



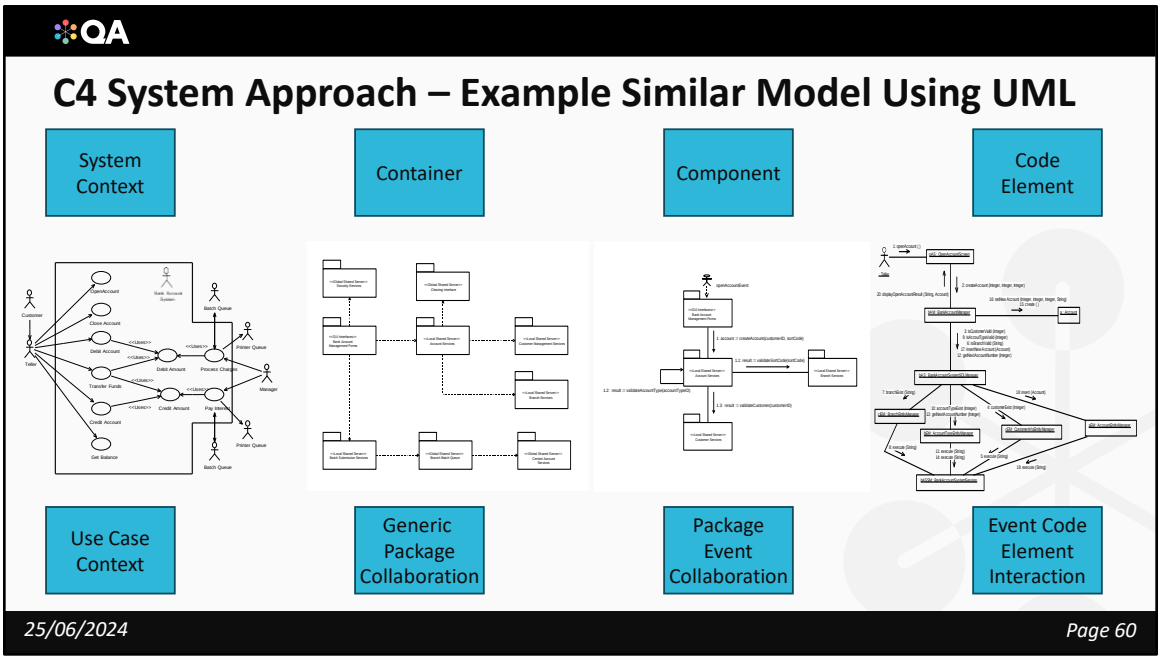
25/06/2024 Page 59

The C4 model is designed to address static structure of a software system. It Starts with a high-level view of the system (Context), then drills down into increasingly detailed views (Container, Component, and Code). This helps separate concerns making it easier to understand complex software architectures

It is useful for communicating software architecture to different stakeholders

A set of hierarchical abstractions (software systems, containers, components, and code)
A set of hierarchical diagrams (system context, containers, components, and code)
Notation independent
Tooling independent

https://c4model.com/



However, you can also easily use standard madling languages for greater standardisation and specificity than arbitrary box and line diagrams to more directly communicate subtleties and avoid misinterpretation for those who share a more complete representation.

The example here is from a bank account system model using the same levelling concepts (also supplemented by more detail and textual explanation where needed).

QA

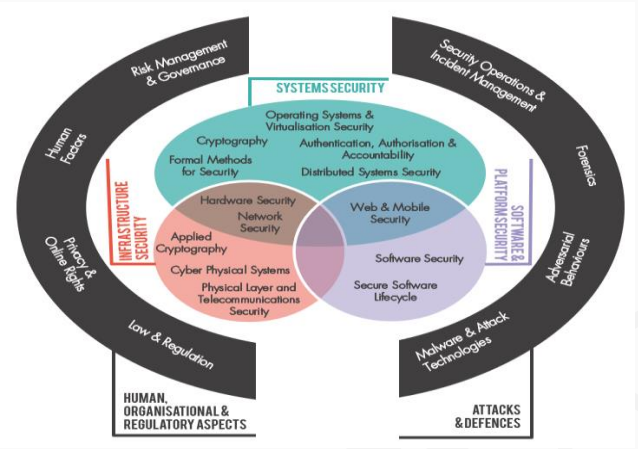
CyBOK

The Cyber Security Body Of Knowledge

The CyBOK is divided into 21 top-level Knowledge Areas (KAs)

Grouped into five broad categories:

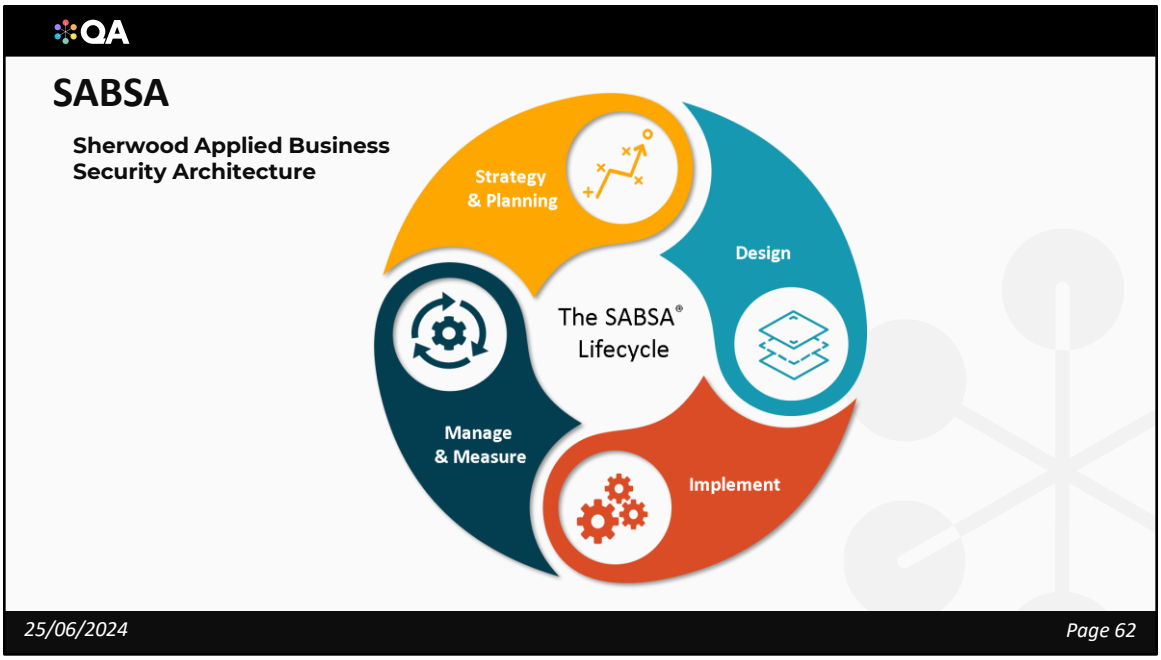
- Human, Organisational & Regulatory Aspects
- Attacks and Defences
- Systems Security
- Software and Platform Security
- Infrastructure Security



University of BRISTOL
Bristol Cyber Security Group

A comprehensive Body of Knowledge to inform and underpin education and professional training for the cyber security sector. The CyBOK project aims to bring cyber security into line with the more established sciences by distilling knowledge from major internationally recognised experts to form a Cyber Security Body of Knowledge that will provide much-needed foundations for this emerging topic. The project, funded by the National Cyber Security Programme, is led by the University of Bristol.

https://www.cybok.org/generallearning/



Sherwood Applied Business Security Architecture

SABSA is a proven methodology for developing business-driven, risk and opportunity focused Security Architectures at both enterprise and solutions level that traceably support business objectives. It is also widely used for Information Assurance Architectures, Risk Management Frameworks, and to align and seamlessly integrate security and risk management into IT Architecture methods and frameworks.

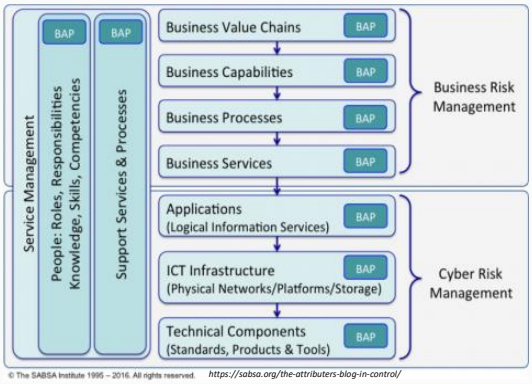
The SABSA framework and methodology is used successfully around the globe to meet a wide variety of Enterprise needs including Risk Management, Information Assurance, Governance, and Continuity Management. SABSA has evolved since 1995 to be the 'approach of choice' for organisations in 50 countries and in sectors as diverse as Banking, Homeless Management, Nuclear Power, Information Services, Communications Technology, Manufacturing and Government.

SABSA ensures that the needs of your Enterprise are met completely and that security services are designed, delivered and supported as an integral part of your business and IT management infrastructure. Although copyright protected, SABSA is an open-use methodology, not a commercial product. SABSA is comprised of a series of integrated frameworks, models, methods and processes, used independently or as an holistic integrated enterprise solution,

https://sabsa.org/sabsa-executive-summary/

QA

SABSA Includes



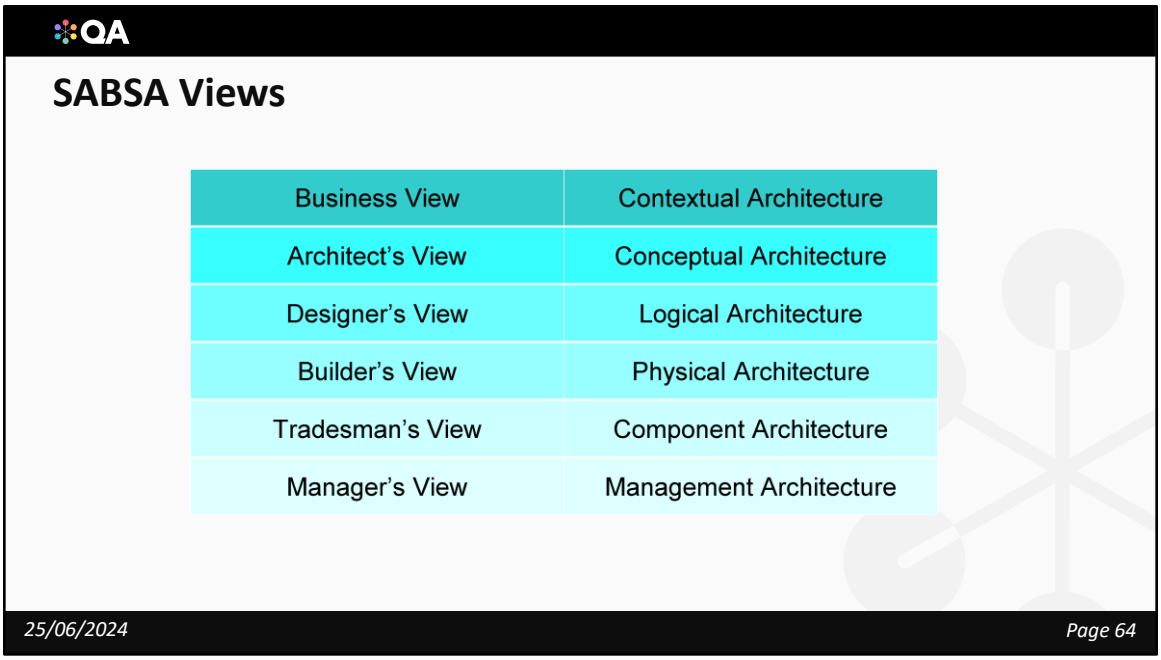
- Business Requirements Engineering Framework (known as Attributes Profiling)
- Risk and Opportunity Management Framework
- Policy Architecture Framework
- Security Services-Oriented Architecture Framework
- Governance Framework
- Security Domain Framework
- Through-life Security Service Management & Performance Management Framework

SABSA is comprised of a series of integrated frameworks, models, methods and processes, used independently or as an holistic integrated enterprise solution, including:

Business Requirements Engineering Framework (known as Attributes Profiling)
Risk and Opportunity Management Framework
Policy Architecture Framework
Security Services-Oriented Architecture Framework
Governance Framework
Security Domain Framework

Through-life Security Service Management & Performance Management Framework
The SABSA Institute develops and maintains the method and certifies and accredits the professional Architects who use it in approximately 50 countries around the world.

https://sabsa.org/sabsa-executive-summary/

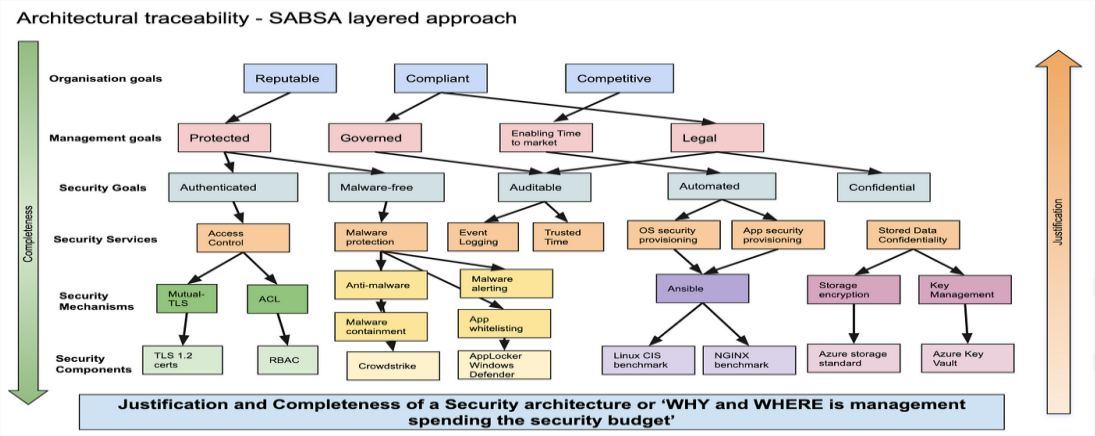


SABSA is a 'Through-Life' method and framework: it applies throughout the entire lifecycle from Business Requirements Engineering to management of the solutions delivered.

https://sabsa.org/sabsa-executive-summary/

QA

SABSA Strategy to Implementation



SABSA is a 'Through-Life' method and framework: it applies throughout the entire lifecycle from Business Requirements Engineering to management of the solutions delivered.

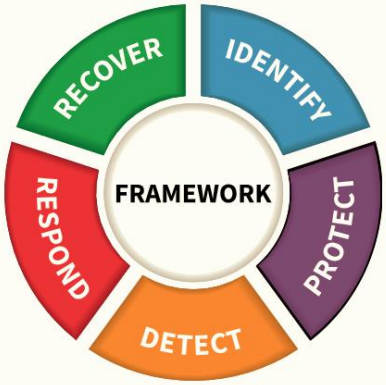
https://sabsa.org/sabsa-executive-summary/

QA

NIST – National Institute Of Standards Of Technology Cyber Security Framework

- NIST develops cybersecurity standards, guidelines, best practices and other resources to meet the needs of U.S. industry, federal agencies and the broader public. Activities range from producing specific information that organizations can put into practice immediately to longer-term research that anticipates advances i technologies and future challenges.

- Some NIST cybersecurity assignments are defined by federal statutes, executive orders and policies. For example, the Office o Management and Budget (OMB) mandates that all federal agencies implement NIST's cybersecurity standards and guidance for non-national security systems. Cybersecurity activities also a driven by the needs of U.S. industry and the broader public.



25/06/2024

NIST also advances understanding and improves the management of privacy risks, some of which relate directly to cybersecurity. Priority areas to which NIST contributes – and plans to focus more on – include cryptography, education and workforce, emerging technologies, risk management, identity and access management, measurements, privacy, trustworthy networks and trustworthy platforms.

https://www.nist.gov/cybersecurity

Section 1 - Architecture Domains and Frameworks

1. The role of architecture in an organisation.
2. Different architecture domains and the relationships between them.
3. Key architecture frameworks.
4. **How industry and professional standards are relevant to architecture.**

QA

STANDARDS

- **Integrity**
- **Competence**
- **Ethics**
- **Career development**

25/06/2024 Page 68

There are many different standards. We provide several following slides to refresh your memory for these standards. We will not go through these in detail, rather we will progress quickly through these and provide references for you to review when working through the workbook and preparing for the BCS assessment.

Standards enable us to follow internal and external legal and regulatory requirements, standardised things and their interfaces and standardised ways of working.

Within an organisation and an enterprise viewpoint there are many different ways to breaks down different concerns. Most architecture frameworks approach this by identifying potential segments/domains (areas of commonality and control) that may be considered to help organise and evolve capabilities, solutions and services. In practice there can be any number of domains depending upon your specific organisation / enterprise situation.

Consider:

Why is conformance to standards useful?
Which of these standard are relevant to your role and how you apply them?
How different situations may require using standards in different ways with different degrees of relevance.
How do they impact upon architecture activities and at what level?
The purpose of aligning work practices and professional knowledge to standards.
Which standards are relevant in your role/industry?
How different scenarios, projects, or industries may impact the relevant standards.

Note that the only standards that you should be examined about are in the following slides.

https://www.iso.org/standards.html

We will be referring to the standards in this list. Consider how they are applied in practice and the impact they have on architecture activities.

https://www.iso.org/standards.html

QA

ISO 42010
(Software Systems and Enterprise Architecture Description)

This standard specifies:

- Requirements for the structure and expression of an architecture description (AD) for various entities, including software, systems, enterprises, systems of systems, families of systems, products (goods or services), product lines, service lines, technologies and business domains.
- Requirements for an architecture description framework (ADF), an architecture description language (ADL), architecture viewpoints and model kinds in order to usefully support the development and use of an AD.
- Conformance to the requirements for an AD, ADF, ADL, architecture viewpoint and model kind.

This standard does not specify:

- The processes, architecting methods, models, notations, techniques or tools by which an AD is created, utilized or managed.

ISO/IEC/IEEE 42010:2022 - Software, systems and enterprise — Architecture description

QA

ISO 25000 Series
(System and Software Quality Requirements & Evaluation)

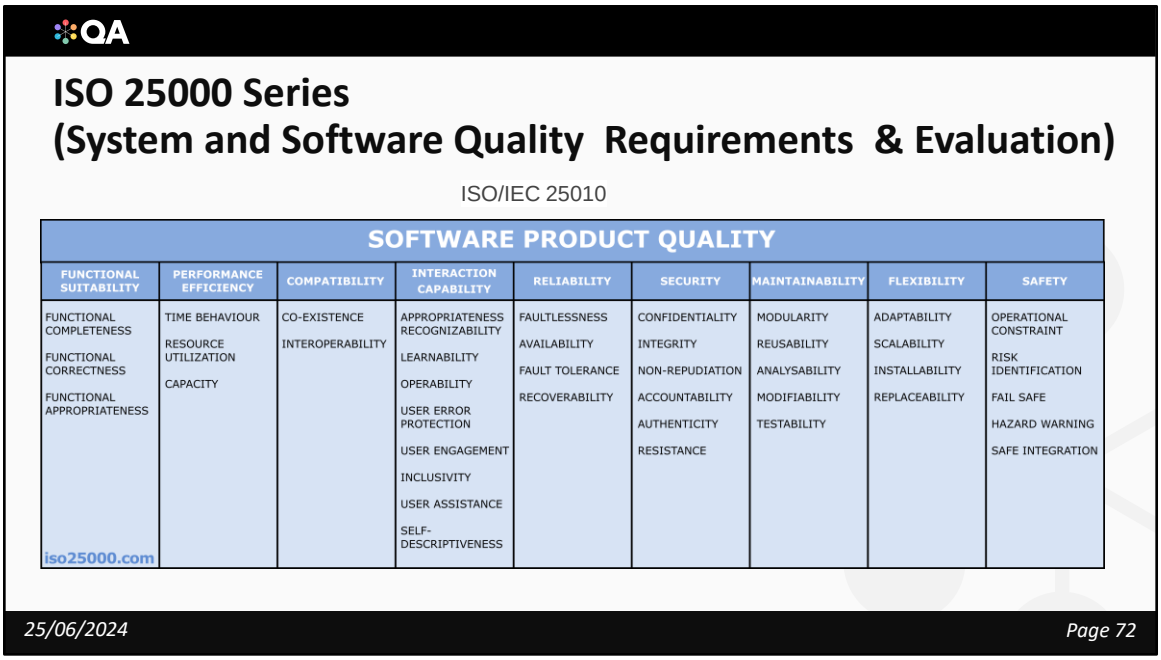
- Provides guidance for the use of the new series of International Standards named Systems and Software Quality Requirements and Evaluation (SQuaRE).
- The purpose of ISO/IEC 25000:2014 is to provide a general overview of SQuaRE contents, common reference models and definitions, as well as the relationship among the documents, allowing users of the Guide a good understanding of those series of standards, according to their purpose of use.
- It also contains an explanation of the transition process between the old ISO/IEC 9126 and the ISO/IEC 14598 series and SQuaRE.

25/06/2024 *Page 71*

The ISO Definition Of Software Quality

Software quality is the "capability of a software product to conform to requirements while for others it can be synonymous with customer- or value-creation or even defect level. Software quality measurements can be split into three parts: process quality, product quality which includes internal and external properties and lastly, quality in use, which is the effect of the software.

ISO 25000 STANDARDS

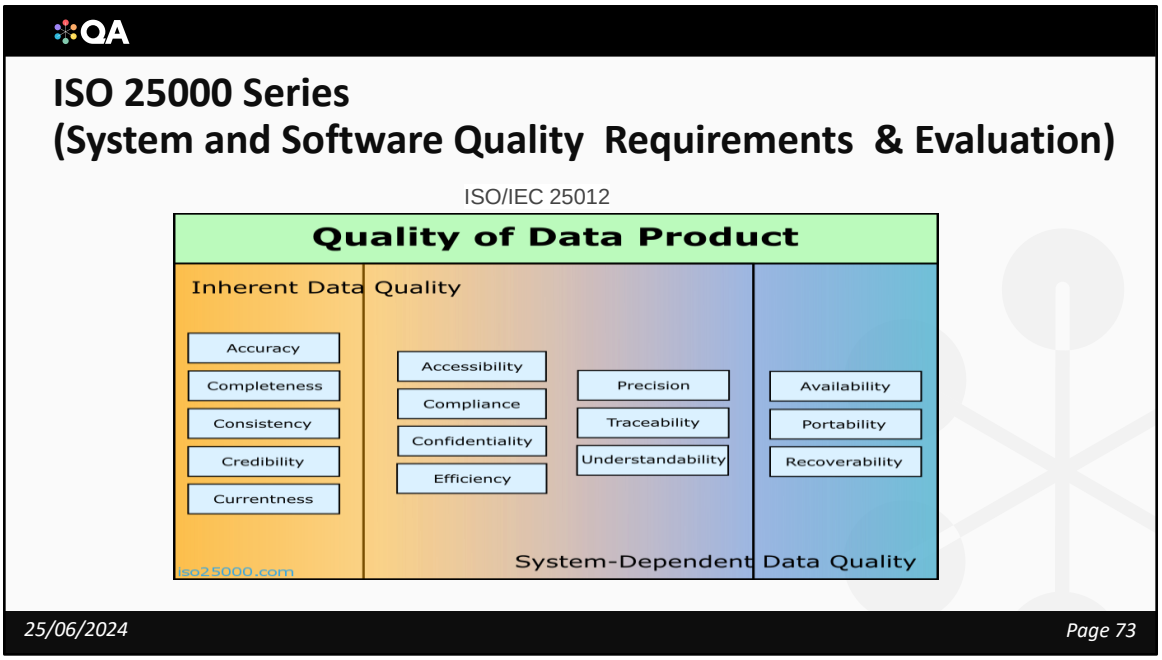


The ISO Definition Of Software Quality ISO/IEC 25010

Software quality is the "capability of a software product to conform to requirements while for others it can be synonymous with customer- or value-creation or even defect level. Software quality measurements can be split into three parts: process quality, product quality which includes internal and external properties and lastly, quality in use, which is the effect of the software.

The quality model is the cornerstone of a product quality evaluation system.
The quality model determines which quality characteristics will be taken into account when evaluating the properties of a software product.
The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value.
Those stakeholders' needs (functionality, performance, security, maintainability, etc.) are precisely what is represented in the quality model, which categorizes the product quality into characteristics and sub-characteristics.
The product quality model defined in ISO/IEC 25010 comprises the nine quality characteristics

ISO 25000 STANDARDS



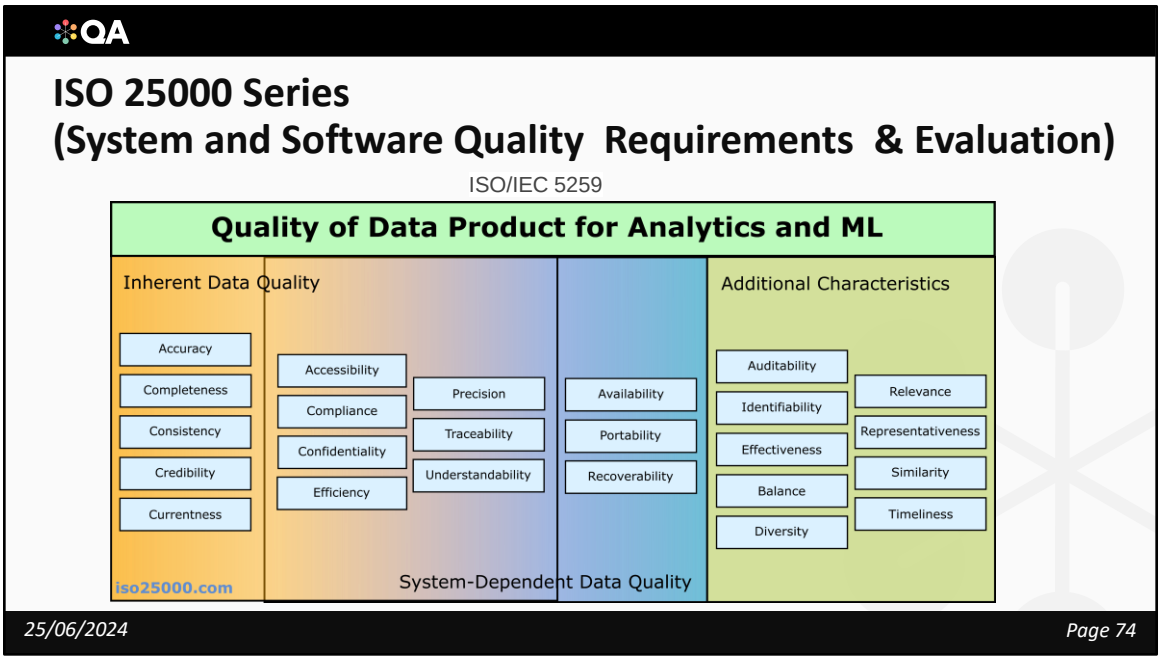
The ISO Definition Of Quality of Data Product ISO/IEC 25012

The Data Quality model represents the grounds where the system for assessing the quality of data products is built on. In a Data Quality model, the main Data Quality characteristics that must be taken into account when assessing the properties of the intended data product are established.

The Quality of a Data Product may be understood as the degree to which data satisfy the requirements defined by the product-owner organization. Specifically, those requirements are the ones that are reflected in the Data Quality model through its characteristics (Accuracy, Completeness, Consistency, Credibility, Currentness, Accessibility...).

The Data Quality model defined in the standard ISO/IEC 25012 is composed of 15 characteristics,

ISO 25000 STANDARDS



The ISO Definition Of Quality of Data Product For Analytics & Machine Learning ISO/IEC 5259

ISO/IEC 5259 describes a data quality model for data analytics and artificial intelligence based on machine learning (ML). This standard is based on the ISO/IEC 25012 model, and like ISO/IEC 25012 differentiates between two types of characteristics: inherent and system-dependent, to which it adds a new category of additional characteristics.

Inherent Data Quality

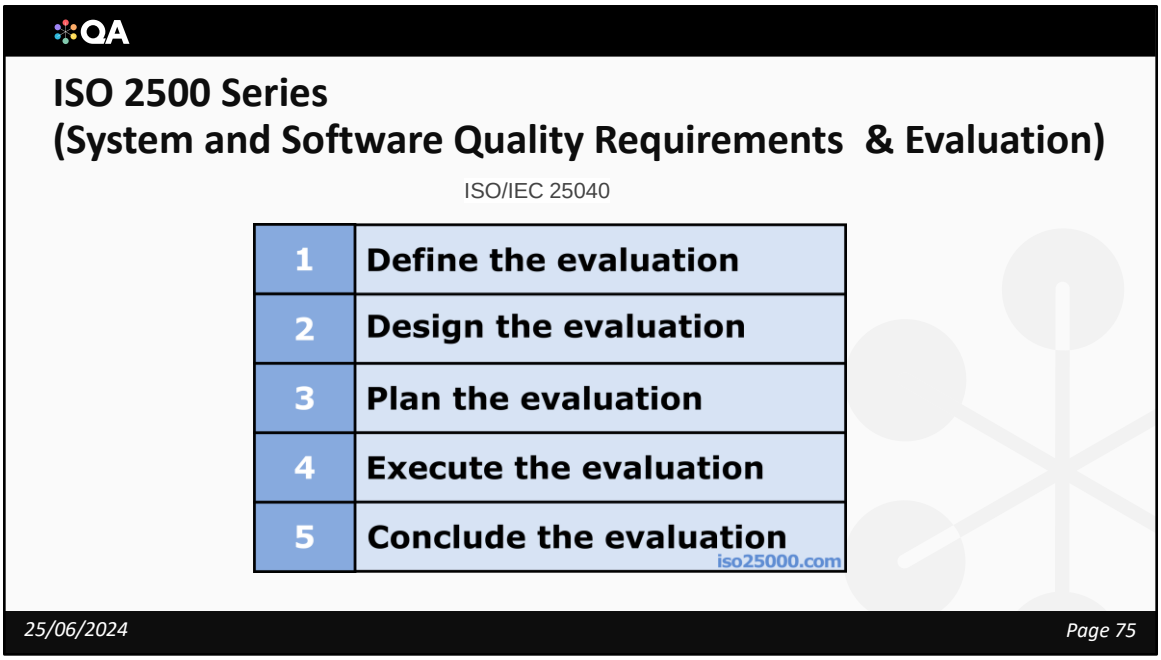
Accuracy
The degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.
It has two main aspects:

Syntactic Accuracy: Syntactic accuracy is defined as the closeness of the data values to a set of values defined in a domain considered syntactically correct.
Semantic Accuracy: Semantic accuracy is defined as the closeness of the data values to a set of values defined in a domain considered semantically correct.

Completeness
The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use.

Consistency
The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. It can be either or both among data regarding one entity and across similar data for comparable entities.

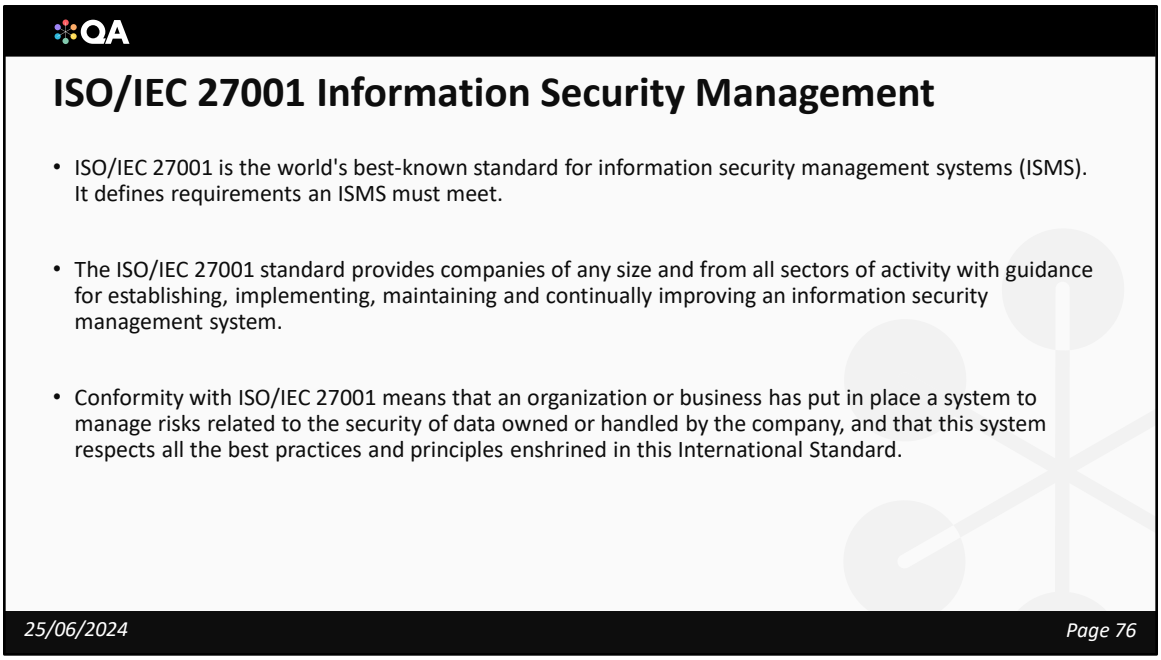
ISO 25000 STANDARDS



The ISO Definition Of Evaluating Quality OF Software product ISO/IEC 5259

SO/IEC 25040 provides a process description for evaluating quality of software product and states the requirements for the application of this process. The evaluation process is composed of five activities.

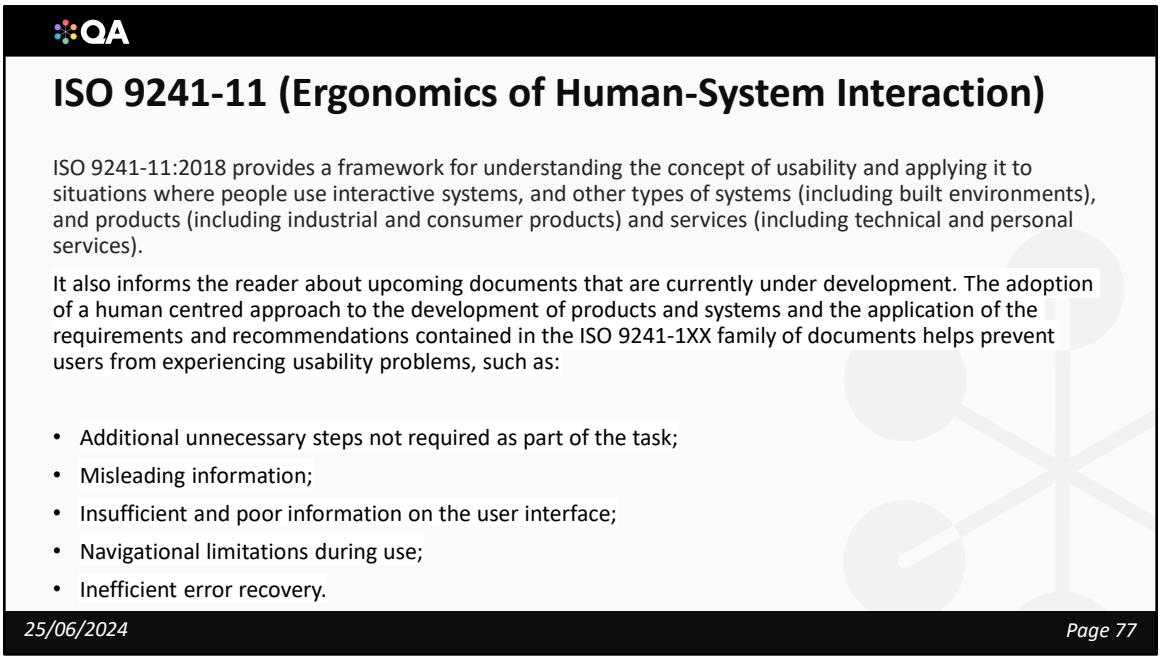
ISO 25000 STANDARDS



With cyber-crime on the rise and new threats constantly emerging, it can seem difficult or even impossible to manage cyber-risks. ISO/IEC 27001 helps organizations become risk-aware and proactively identify and address weaknesses.

ISO/IEC 27001 promotes a holistic approach to information security: vetting people, policies and technology. An information security management system implemented according to this standard is a tool for risk management, cyber-resilience and operational excellence.

https://www.iso.org/standard/27001



This document provides an overview of ISO 9241 software ergonomic standards in the form of executive summaries of these standards, in particular the parts in the ISO 9241-1XX family of documents. In addition, it provides executive summaries for ISO 9241-11, ISO 9241-210 and ISO 9241-220, which have specific relevance to the design of software-based interactive systems.

This document is intended for the following types of users:

Managers, who are involved in planning and managing product, system and/or service development projects, who are to be informed on the human-centred design approach and on guidance on software ergonomics;
Developers, who will apply the guidance in these documents during the development process (either directly, based on training, or by using tools and style guides which incorporate the guidance);
User interface design roles (including interaction designers, information architects, user interface designers, visual designers and content creators), who will apply the guidance in these documents during the creation and design process (either directly, based on training, or by using tools and style guides which incorporate the guidance);
User researchers, who are responsible for identifying user needs and inform context of use of a product, system or service;
Evaluators, who are responsible for ensuring that products, systems or services meet the recommendations contained in these documents;
Buyers, who will reference these documents in contracts during product procurement;
Designers of user interface development tools and style guides to be used by user interface designers and developers.

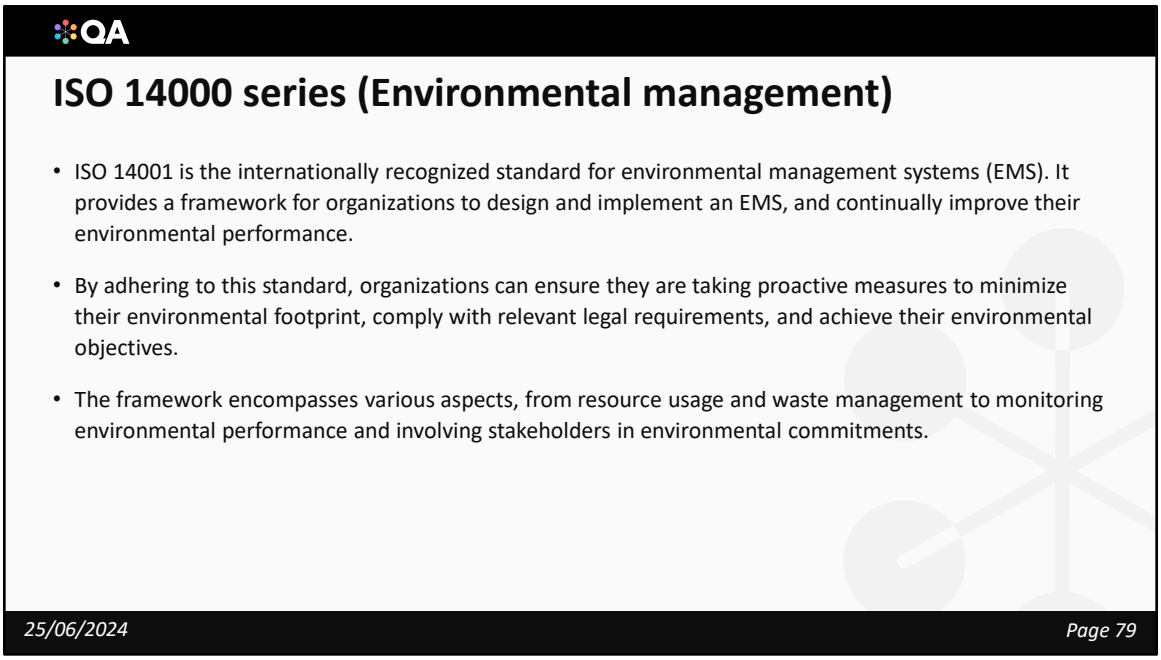
https://www.iso.org/standard/63500.html

QA

ISO 15704 Enterprise Modelling And Architecture

- Specifies a reference base of concepts and principles for enterprise architectures that enable enterprise development, enterprise integration, enterprise interoperability, human understanding and computer processing. This document further specifies requirements for models and languages created for expressing such enterprise architectures.
- Specifies those terms, concepts and principles considered necessary to address stakeholder concerns and to carry out enterprise creation programmes as well as any incremental change projects required by the enterprise throughout the whole life of the enterprise. This document forms the basis by which enterprise architecture and modelling standards can be developed or aligned.
- Does not define standard enterprises, standard organizational structures, standard enterprise processes, or standard enterprise data. In addition, this standard does not specify enterprise modelling processes.

https://www.iso.org/standard/71890.html



In an age of heightened environmental consciousness and increasing global challenges such as climate change, biodiversity loss, and resource depletion, organizations have a pivotal role to play. ISO 14001 offers a structured approach for businesses to address these pressing concerns. By adopting this standard, organizations signal a commitment not only to regulatory compliance but also to ongoing environmental improvement. This proactive approach to environmental management can result in tangible benefits, such as reduced waste, energy conservation, and cost savings.

Furthermore, it enhances an organization's reputation, fosters stakeholder trust, and often constitutes a critical step for engaging in global trade and supply chains. Simply put, ISO 14001 stands as a testament to an organization's dedication to a sustainable future, blending environmental responsibility with strategic business growth

https://www.iso.org/standard/60857.html

ISO 38500 (Corporate governance of IT)

This standard provides guiding principles for members of governing bodies of organizations and those that support them on the effective, efficient and acceptable use of information technology (IT) within their organizations. It applies to:

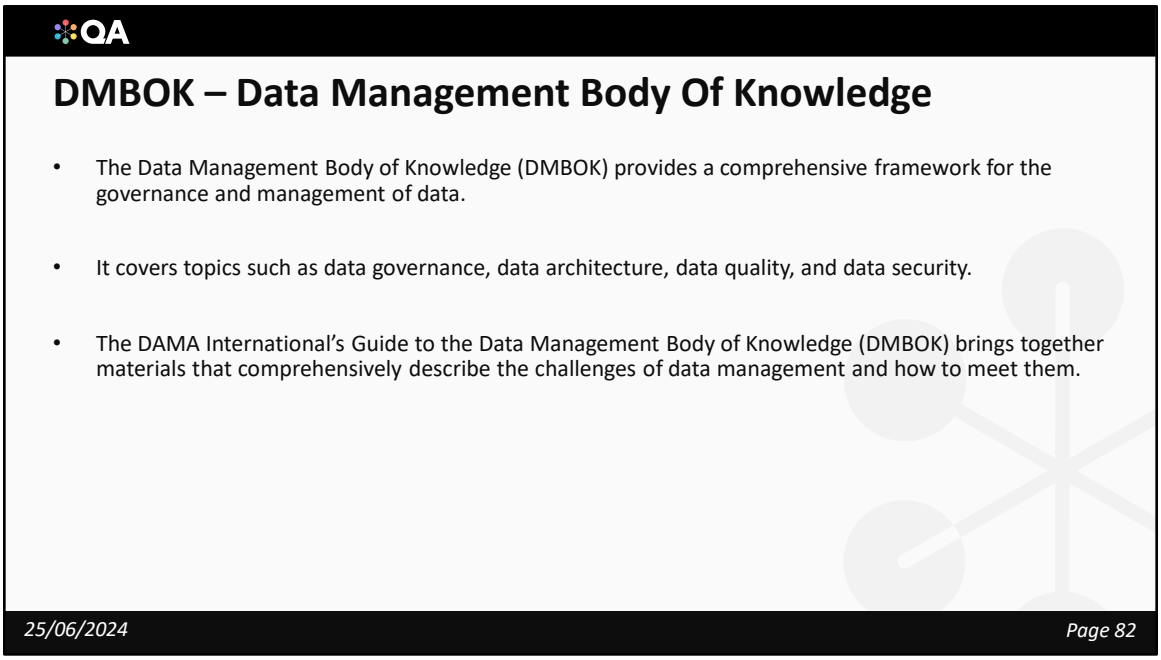
- The governance of the organization's current, and future, use of IT;
- The governance of IT as a domain of governance of organizations.
- All organizations, including public and private companies, government entities, and not-for-profit organizations;
- Organizations of all sizes, from the smallest to the largest, regardless of the extent of their use of IT.

https://www.iso.org/standard/81684.html

Bodies Of Knowledge

Selected Set:

- **DMBOK** (Data Management Body of Knowledge)
- **BIZBOK** (Business Architecture Body of Knowledge)
- **BTABoK** (Business Technology Architecture Body of Knowledge)



The DMBoK includes the latest processes, best practices, and principles within Data Management. It focuses on 11 main topics:

Data Governance
Data Architecture
Data Modelling & Design
Data Storage & Operations
Data Security
Data Integration & Interoperability
Documents & Content
Reference & Master Data
Data Warehousing & Business Intelligence
Metadata
Data Quality

In addition to that, the latest edition of the DMBoK contains dedicated sections for data ethics, big data, data integration and interoperability, and Data Management maturity assessment. To illustrate how these different subjects are interconnected, DAMA International created the Data Management Framework or DAMA Wheel. It includes the 11 knowledge areas and puts Data Governance at the center of the wheel, in order to highlight how it underpins every Data Management activity. The Goals of the DMBoK Are:

To provide information about best practices, roles and responsibilities, deliverables and metrics, and maturity models for Data Management
To standardize management practices across the field
To serve as a functional framework for the implementation of these practices in any business context
To establish a formal vocabulary for these concepts
To clarify the scope of what these practices can and cannot do
To provide a vendor-neutral overview of management practices and potential alternatives for specific situations

https://technicspub.com/dmbok/

BIZBOK – Business Architecture Body Of Knowledge

- The Business Architecture Body of Knowledge (BIZBOK) is a guide to best practices and methods in business architecture.
- It provides a common standard for the development and deployment of business architecture.

A Guide to the Business Architecture Body of Knowledge® (BIZBOK® Guide) provides an industry standard framework for business architecture practitioners and individuals who wish to use business architecture to address business challenges.

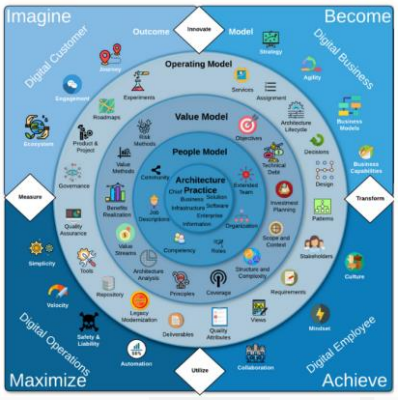
This practical guide comes in the form of best practices, gleaned from numerous companies and business architecture leaders.

Practitioners of business architecture understand the importance of having a comprehensive yet user-friendly handbook for the growing number of organizations embracing this important discipline.

https://cdn.ymaws.com/www.businessarchitectureguild.org/resource/resmgr/bizbok_8_5/bizbok_v8.5_final_part1.pdf

BTABoK – Business Technology Architecture Body Of Knowledge

- The Business Technology Architecture Body of Knowledge (BTABoK) provides a framework for aligning business and IT strategies.
- It covers topics such as business process management, enterprise architecture, and IT governance.



The Business Technology Architecture Body of Knowledge (BTABoK) is a free public archive of Business Technology architecture best practices, skills, and knowledge developed from the experience of individual and corporate members of Iasa, the world's largest architecture professional organization. BTABoK provides the tools and resources needed by individuals and organizations to set industry standards for professional career development and well as hiring practices and incorporation of IT architects into established or developing institutions.

The Business Technology Architecture Body of Knowledge (BTABoK) has been developed from the experience and practice of individual and corporate members. In addition to being a reference, a knowledge base and a list of professional capabilities, the BTABoK is meant to be essential in implementing an architecture practice within an organization, without significant changes to other standards, roles, practices and lifecycles. An architecture practice provides high levels of value in both business and technical strategy, and that value can be measured as a contribution to the organization. Architecture is appropriate and essential for extremely small businesses, as well as the largest businesses in the world, including non-profit, government, and defence.

The BTABoK is first and foremost intended for individual practicing architects, though it does include significant portions dedicated to organizational excellence. It is meant as a people framework as opposed to a process, methodology or standard. It is meant to be the living body of knowledge for the practice of an architect attempting to fulfill their duty to their customer or employer. While Iasa includes corporate examples and concerns, the BTABoK must be useful by the independent practitioner and must assume that a single architect is as important as a group of architects. In addition, the BTABoK must be consistent and useful at any degree of scale, regardless of the number of architects involved in the endeavour, whether that be the delivery of a global fortune 100 business strategy or the delivery of a single solution for a small business.

The BTABoK is inclusive of all specializations, sub-specializations and career levels of the architect profession. Although the title includes the terms information technology, the content has been written by keeping business, enterprise, solutions, software, information and infrastructure architects and their

derivations in mind. Practitioners from each of these specializations continue to help in the development and maintenance of the body of knowledge.

https://iasa-global.github.io/btabok/

STANDARDS

- **Integrity**
- **Competence**
- **Ethics**
- **Career development**

25/06/2024 *Page 85*

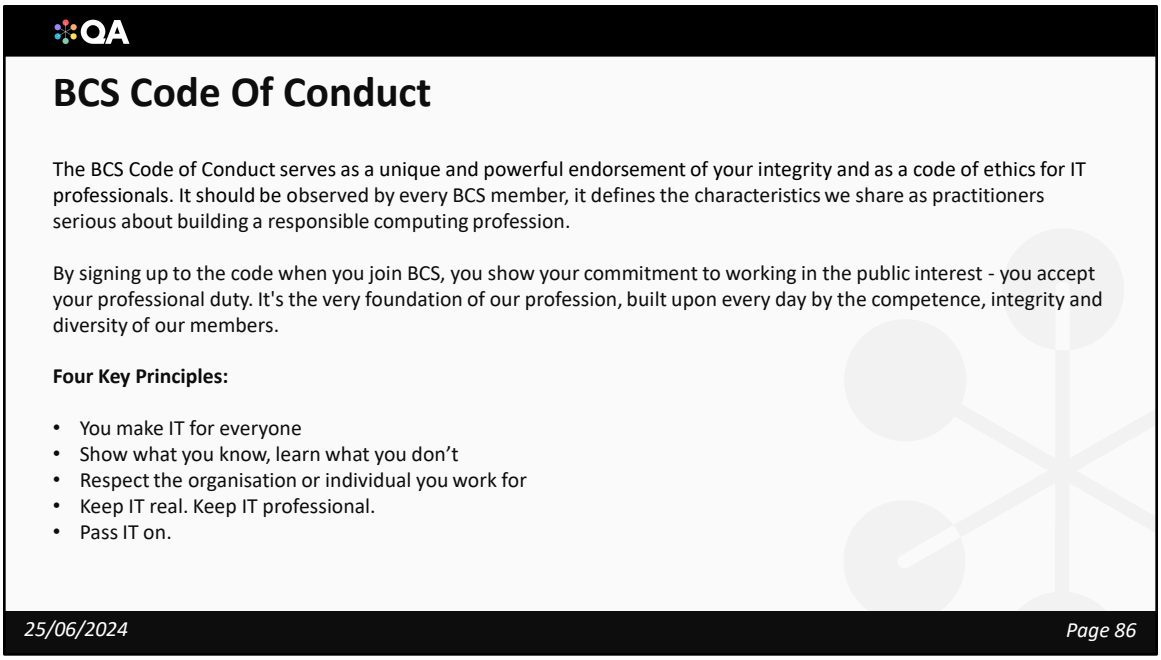
Consider:

The purpose of aligning work practices and professional knowledge to standards.
What standards are relevant in your role/industry?
How different scenarios, projects, or industries may impact the relevant standards.

Professional standards included in the exam are:

BCS Code of Conduct
DDaT (Digital, Data, and Technology Professional Capability Framework)
SFIA+ (Skills Framework for the Information Age Plus)

You may be asked for examples of how these standards are considered as guidance for ways of working, how they are applied in practice and the impact they have on architecture activities.

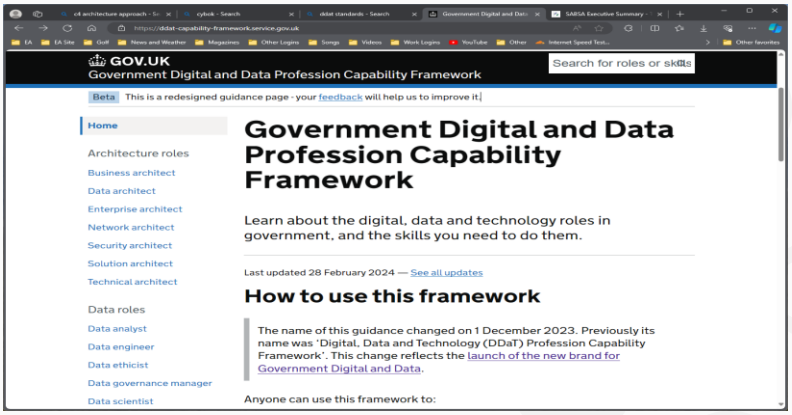


As a professional body, the British Computer Society (known as BCS, the Chartered Institute for IT), has a responsibility to set rules and professional standards to direct the behaviour of its members in professional matters. It is expected that these rules and professional standards will be higher than those established by the general law and that they will be enforced through disciplinary action which can result in expulsion from membership. Members are expected to exercise their own judgement (which should be made in such a way as to be reasonably justified) to meet the requirements of the code and seek advice if in doubt.

https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf

DDaT (Digital, Data, and Technology Professional Capability Framework)

The UK government framework for digital profession skill.



Anyone can use this framework to:

Learn what the different digital and data roles do in government
Understand what skills are needed at each role level

Professionals in Government Digital and Data can use this framework to:

Identify skills that they can develop
Assess their current skill levels in preparation for performance and development conversations
Learn about the typical responsibilities and skills of their colleagues

Line managers and team leaders in government can use this framework to:

Identify skill gaps in their teams and opportunities for development
Inform development goals and conversations
Forecast their organisation workforce needs, to make sure they have the right skills to achieve objectives

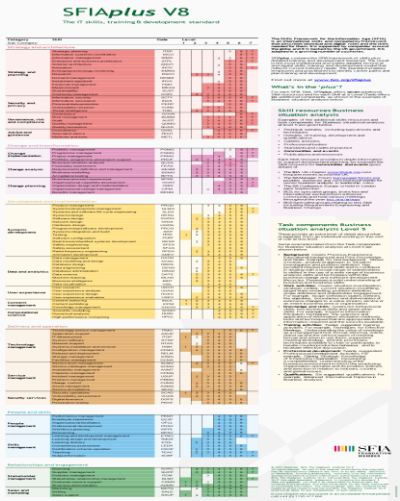
Hiring managers in government can use this framework to:

Create effective and consistent job adverts
Assess the suitability of a candidate during interview

https://ddat-capability-framework.service.gov.uk/

SFIA +

- SFIA - the skills framework for the information age
- SFIA is a globally-recognised standard, developed by industry experts and backed by Government as best practice for identifying and developing IT capability.
- Crucially SFIA offers a common skills language, accessible to IT, HR and L&D professionals, which breaks down barriers between teams, departments and organisations when it comes to transferring and recruiting digital talent.



SFIA plus was developed by BCS to bring the SFIA framework to life, adding detailed training and development resources. For each SFIA skill at each level, SFIA plus offers six additional task definitions and eight additional skills resources.

The result is the industry's most established and widely adopted IT skills model, enabling both employers and practitioners to identify career pathways and plan development aligned with business transformation.

https://sfia-online.org/en

Please open your workbooks to topic 1

Section 02

Architecture And The Organisation

Elements to be addressed:

Architecture and the Organisation

Explain the role of architecture to other business areas.
Discuss use of the business change lifecycle as an input for architecture.
Discuss the business needs and the intended solution.

QA

IT architecture interacts with other areas of the business by aligning technology strategies and solutions with overall business goals.

The Course Schedule

DAY 1 09:00 - 16:30

0.0 Introduction and Definitions

1.1 Refresh on Architecture Domains and Frameworks

1.2 Practice Exercises

2.1 Refresh on Architecture and the Organisation

2.2 Practice Exercises

Architecture and the Organisation

1. **The role of architecture to other business areas.**
2. The business change lifecycle as an input for architecture.
3. The business needs and the intended solution.

Mission, Strategy, And Structure

- Organisation mission, vision and purpose
- Organisational strategy
- Organisational structure

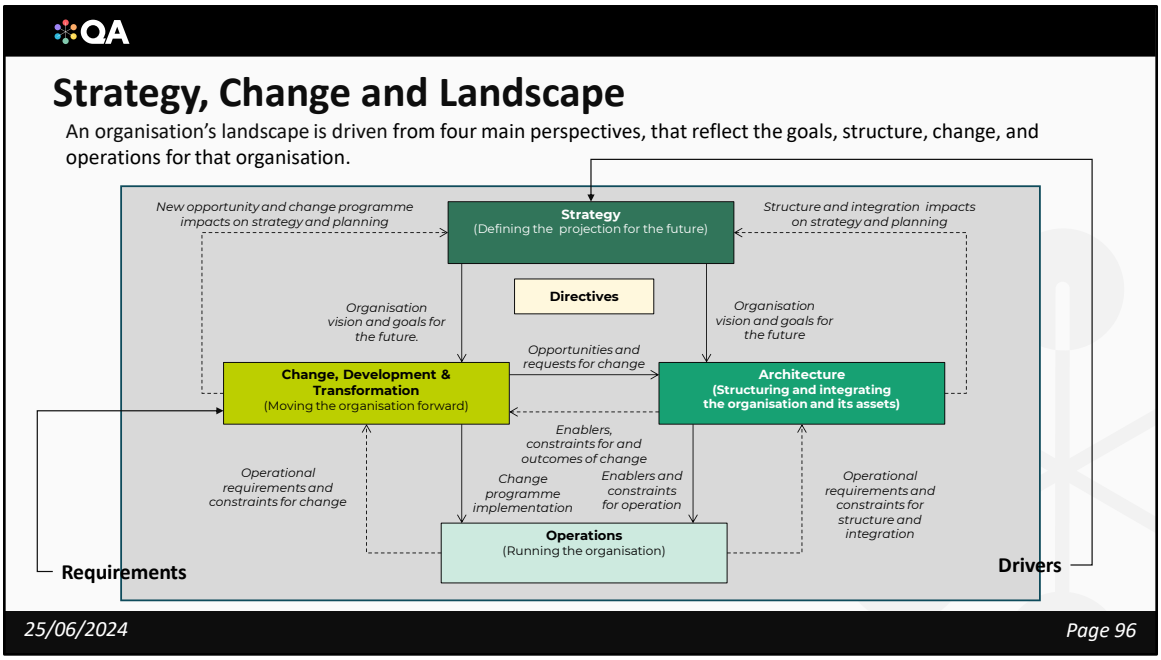
Consider:

How different organisations vary in size, personnel, global reach, etc.
How does the makeup of an organisation impact the role of architecture?
What aspects of an organisation influence the IT architectural approach?
What is the purpose of each business domain?

Business shape is driven by many perspectives:

Mission
Vision
Purpose
Strategy
Structure
Multiple specific business domain/function outcomes

You should be able to demonstrate your understanding of a business situation, linking it to the organisation's strategy, mission, vision and purpose, discuss the structure of the organisation, and describe the relationship between architecture and other business areas.



Strategy defines mission, drivers, goals and objectives with expected change areas of business capability, value steams, information sets and potential courses of action.

Drivers are the causes of consideration for change that may come from inside or outside of the organisation. Drivers may be relatively general and high level while requirements are at a greater levels of details (SMART).

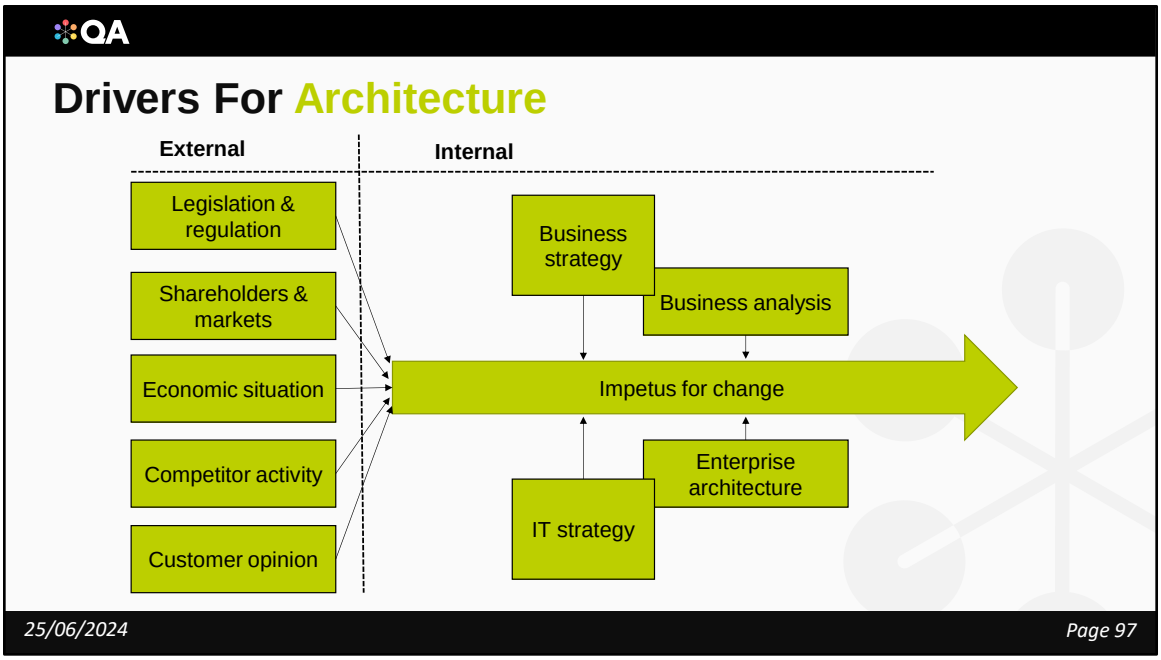
Directives (Principles, Policies, Standard, Best Practices and Business Rules) provide a bridge between strategy, the organising architecture and the solution implementation.

Architecture provides a description of the building blocks that make up the resources of the enterprise to implement its needed capabilities organised into implemented systems and services and manged by some set of domains providing control and management of those capabilities and their use.

Change organises the specific units of specification, development and implementation that will cause the operation of the organisation to be changed in line with its strategy and specific requirements.

Operations implements and manages the executing processes and entities that perform the work of the organisation delivering outcomes that satisfy the organisation's goals, objectives and requirements.

Requirements (things that are needed) may come from inside or outside of the organisation and are an input to all of these organisational elements providing the specific detailed targeted inputs and outcomes that implement the strategy. Requirements can be identified from any relevant stakeholder associated with a change. Requirements should be in line with agreed goals and objectives and directly feed specifically into change activity.

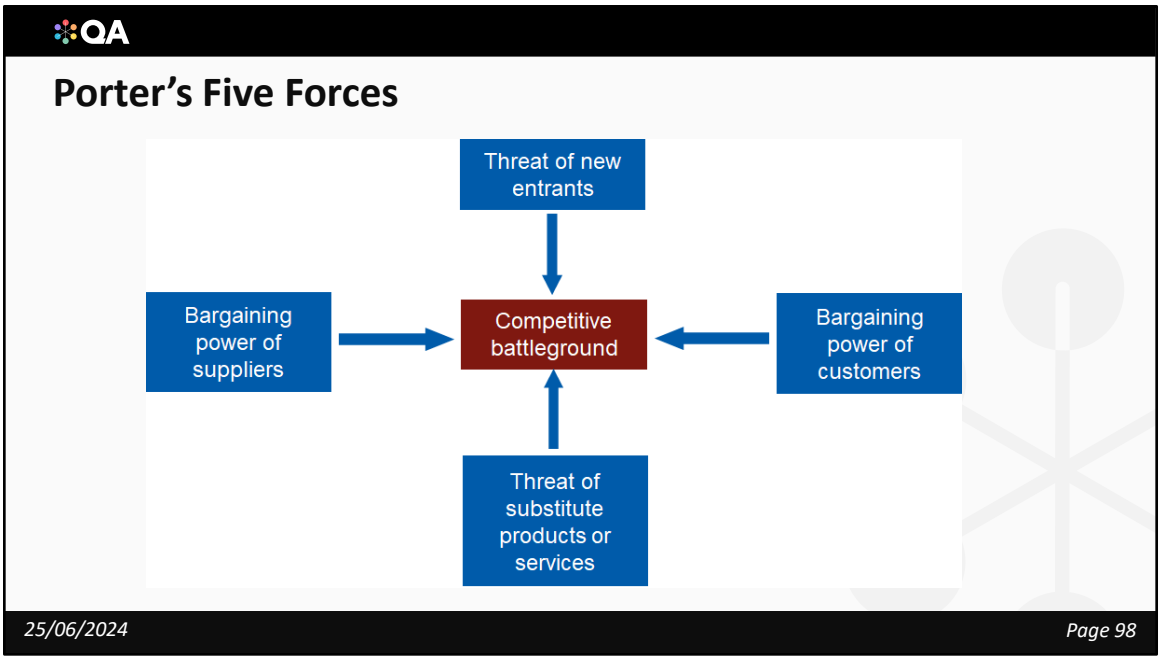


The drivers for architecture may be both internal and external.
Many different types of situation may generate the desire for (or drivers) for change.
Business analysis uncovers the drivers and generalised goals and objectives together with the greater detail of specific requirements.

Goals, objectives and requirements should be Specific Measurable Actionable Realisable Time-Bound

Actual drivers that can arise from these sources are:

Corporate re-organisation
Availability of IT resources
Management prerogative
Habit



Michael Porter at Harvard identified 5 main competitive forces for organisations (initially in his1979 Harvard Business Review article) and then further explored in the 1980s in Competitive Advantage and Competitive Strategy.

These identify the types of changes a market or economy could experience.

There is always a significant impact of external legislation and regulations, which may often be in conflict with each other.

https://www.isc.hbs.edu/strategy/business-strategy/Pages/the-five-forces.aspx

“PESTLE analysis identifies and evaluates how Political, Economic, Social, Technological, Legal, and Environmental factors impact business operations. The strategic planning framework helps business and project managers in the decision-making process.

A PESTLE analysis helps them determine how these factors will affect a business’s performance and strategy in the long term. It gives a bird’s-eye view of the whole environment that business managers need to consider while contemplating a certain business idea or plan, based on the 6 elements of PESTLE analysis.”

https://pestleanalysis.com/what-is-pestle-analysis/

POPIT

Is a generalised way of thinking about how a business is organised and the interaction between its elements. In most situations all five of these elements must be considered and addressed.

https://www.improvementservice.org.uk/business-analysis-framework/investigate-situation/popit

Engaging With Other Business Functions

Engaging with:

- IT
- Ops
- Finance
- Other roles

Consider your understanding of a business situations, linking them to the organisation's strategy, mission, vision and purpose.

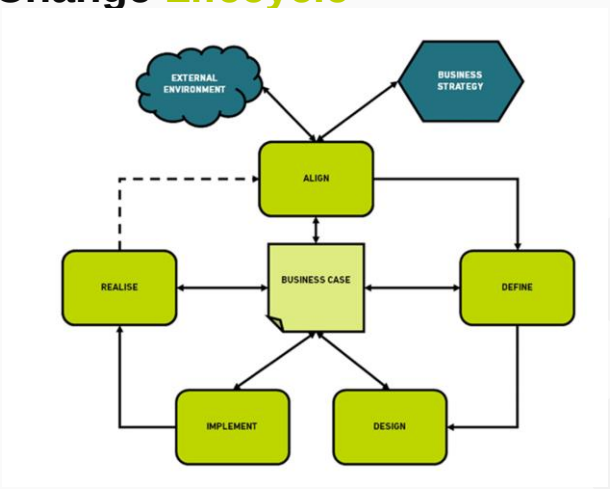
Consider the relationship between architecture and other business areas.

Architecture and the Organisation

1. The role of architecture to other business areas.
2. **The business change lifecycle as an input for architecture.**
3. The business needs and the intended solution.

QA

Business Change Lifecycle



From - Solution Architecture Foundations – Mark Lovatt - BCS - 2022

Consider:

The Business Change Lifecycle

Alignment
Ensuring that the organisation's objectives and strategy are aligned with the external business world and that proposed changes align with internal policies and architectures.

Definition
Taking a closer look at a proposed business situation in order to uncover root causes of problems, conduct a thorough analysis, recommend relevant and feasible changes, and define the requirements.

Design
The detailed specification, development and testing of the solution, including the business process and related tasks, and the software that is needed to support them.

Implementation
The planning and preparation for the deployment of the business changes.

Realisation
The review of the predicted benefits with a view to identifying those that have been achieved and taking further action to support the achievement of those still to be achieved.

You should be able to the stages of the business change lifecycle and relate them to a given business situation

Architecture and the Organisation

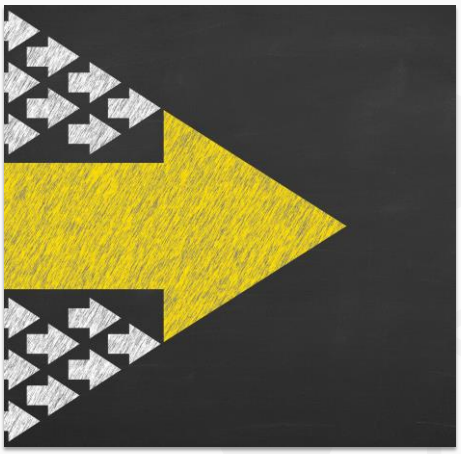
1. The role of architecture to other business areas.
2. The business change lifecycle as an input for architecture.
3. **The business needs and the intended solution.**

QA

Business Needs and Intended Solution

Define the following:

- Problem statement
- Directive hierarchy
- Target solution hierarchy
- High level design



25/06/2024 Page 105

Consider:

How a problem statement provides clarity
How a directive hierarchy influences a solution (E.g. enterprise architecture, organisational principles, policies, business rules, etc.)
How a target solution hierarchy influences solution design.
How a high-level design links to a detailed design and implementation.

You should be able to discuss how business needs are established and considered when planning a solution, and how these are demonstrated in the high-level design. The need to understand a problem statement, alongside the consideration or organisational principles, policies and business rules should be explained.

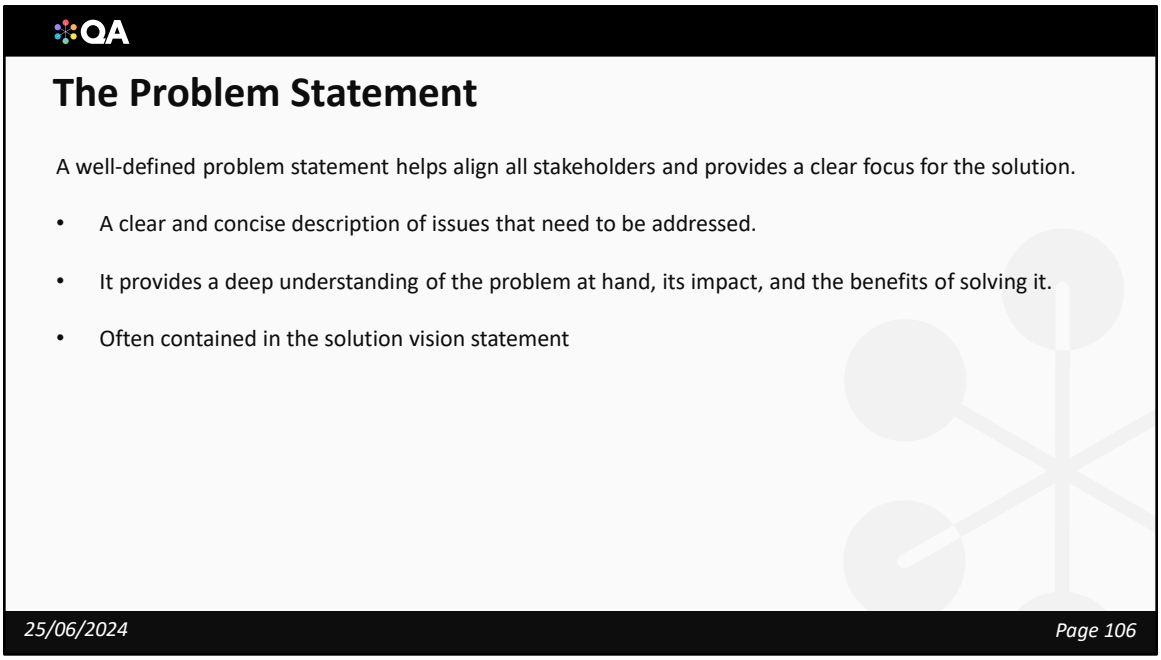
Architecture is not an isolated function but is deeply intertwined with all aspects of an organisation.

It provides a structured approach to aligning the IT landscape with business needs.
Provides a structured approach to solution development.

Always consider the most appropriate approach to the

Decision-making processes
Risk management
Resource allocation across different business functions.

There will be different approaches for different types of problem/opportunity. Ensure you have understood the best balance of planning/intention and discovery/emergence.



Address the problem and solution states:

Current / Baseline
Intermediate/Transition
Final / Target

It is important to understand the problem statement and consider organisational principles, policies, standard, best practices and business rules when planning a solution.

Consider how business needs are established and addressed when planning a solution, and how are these demonstrated in a high-level design.

QA

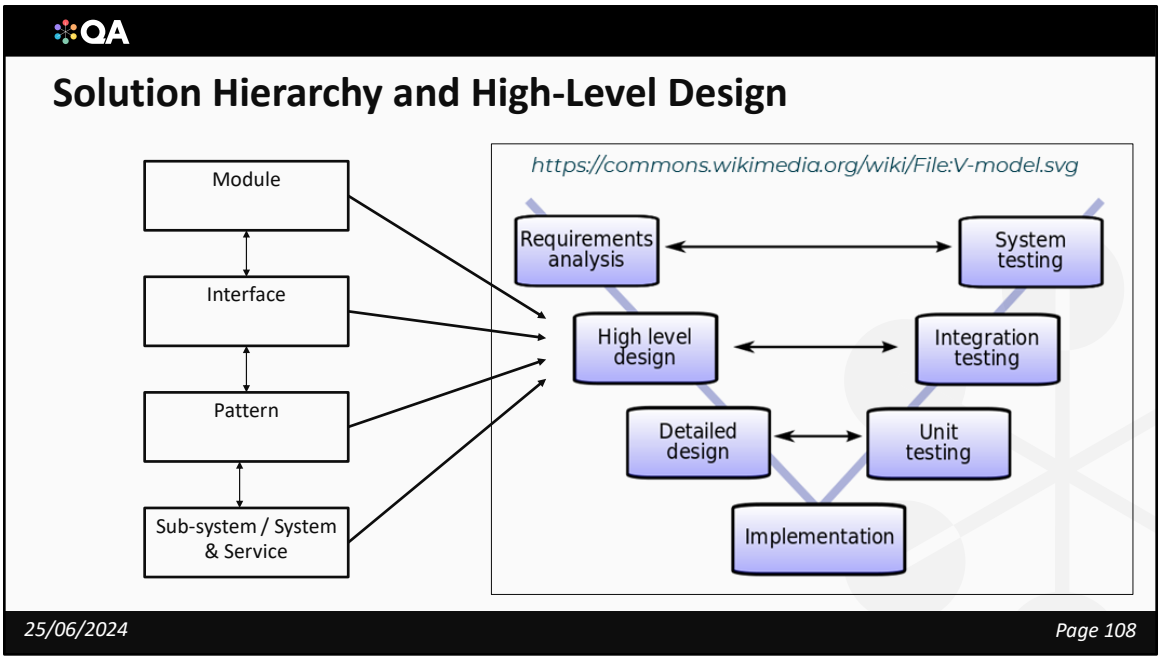
The Directives Hierarchy

Directives are key properties of our architecture and guide how our solutions should be created. They are minimally defined as principles, policies and business rules, but are often extended to include standards and best practices.

Item	Name	Details
1	Principles	Statements of general good practice that should normally be applied.
2	Policies	Describe how principles should be applied in day-to-day work.
3	Standards	Specific things (specifications and/or specific implementations and products) that should be used in particular circumstances.
4	Best practices	Practices that should be adopted and followed.
5	Business rules	Directs and constrains how business should be carried out. Actionable rules derived from the higher directives.

The directives hierarchy provides a structured set of elements to shape the evolving architecture landscape.

Architects should strive to follow them but when the solution conflicts with the directives seek review the issues and possibilities and gain appropriate approvals before progressing.



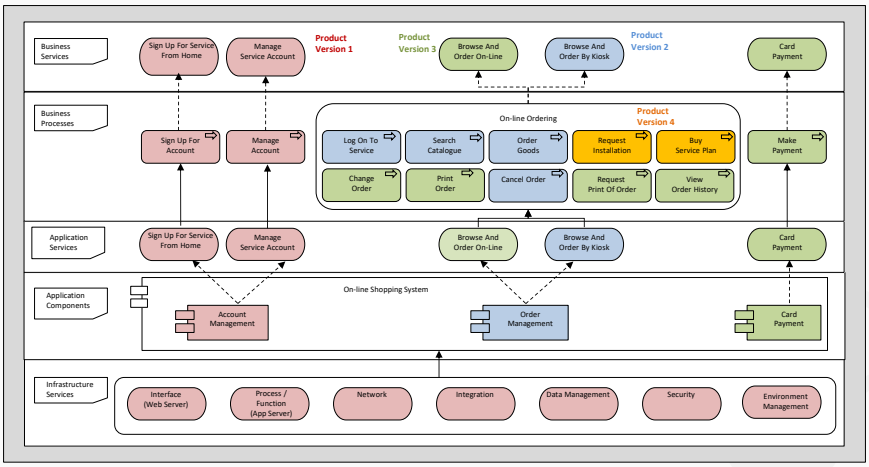
A high-level design shows the modular structure of a solution and its interactions. Architects provide specifications for :

The capabilities to be included in the solution.
Organising the elements into managed sub-systems and systems.
The modules/building blocks/components in the solution and what each should do.
What interfaces are offered.
The patterns that control their interaction
The services supported and the service levels delivered.

When creating a high-level design, start by understanding the business needs thoroughly. Translate these needs into business and technical requirements and design an outline solution that meets these requirements while aligning with the organisation's strategic goals. The high-level design provides a comprehensive overview of the solution, including the main components, their interactions, and how they support the business needs. Consider how the business needs have been established and considered when planning a solution, and how these are demonstrated in the high-level design. Understand the interaction between the problem statement, and the organisational principles, policies, standards, best practices and business rules.

Detailed designers and builders (software, buildings, devices etc.) take these specifications and provide their internal module/building block/component implementations performing additional internal design work to ensure that the requested specifications can be delivered for each specific change being designed, deployed and operationalised.

Example Hierarchical Solution Context Diagram



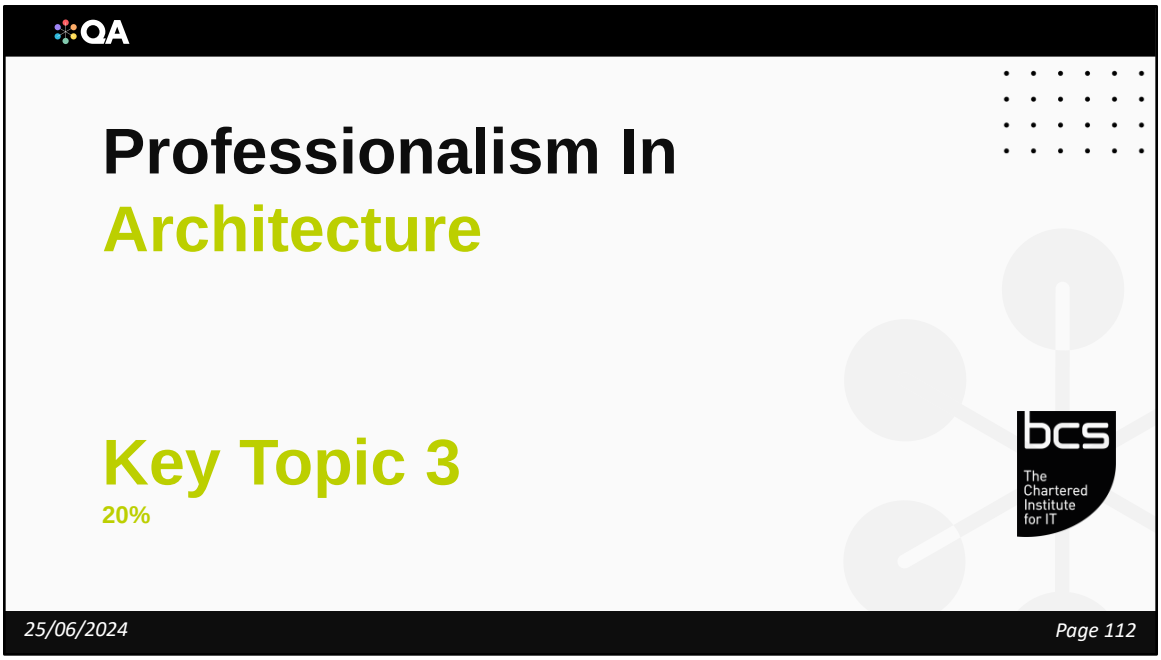
A solution can be presented as a set of interacting architecture layers with the ultimate business service target at the top and the infrastructure execution at the bottom. The data architecture would interact with the application services and components and the information architecture would interact with the business services and processes.

The example shown is an Archimate context diagram including iterative implementation of solution chunks/sprints/work packages.

Please open your workbooks to topic 2

Section 03

Professionalism In Architecture



Elements to be addressed:

Discuss compliance with professional standards.

Discuss types of stakeholders.
Explain specific stakeholder needs.
Discuss how to influence others.
Apply a practical customer focus.
Discuss the roles and dynamics of a successful team.
Discuss social, ethical and economic concerns to the role of an architect.

The Course Schedule

DAY 2 09:00 - 16:30

3.1 Refresh on Professionalism in Architecture

3.2 Practice Exercises

4.1 Refresh on Corporate Governance

4.2 Practice Exercises

5.1 Refresh on Architectural, Processes, Tools and Vision

5.2 Practice Exercises

Professionalism In Architecture

1. **Discuss compliance with professional standards.**
2. Discuss types of stakeholders.
3. Explain specific stakeholder needs.
4. Discuss how to influence others.
5. Apply a practical customer focus.
6. Discuss the roles and dynamics of a successful team.
7. Discuss social, ethical and economic concerns to the role of an architect.

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. **Discuss types of stakeholders.**
3. Explain specific stakeholder needs.
4. Discuss how to influence others.
5. Apply a practical customer focus.
6. Discuss the roles and dynamics of a successful team.
7. Discuss social, ethical and economic concerns to the role of an architect.

A stakeholder is an individual, group, or entity that has an interest, involvement, or influence in a project, organisation, or decision, and can be affected by its outcomes.

Stakeholders

- Stakeholder categories and management strategies
- Technical and non-technical stakeholders
- Stakeholder communication plan

25/06/2024 Page 117

Consider:

How different stakeholders can be organised into various categories (e.g. technical, business sponsor, etc.)
How does the type of stakeholder affect how you might interact with them?
What are the priorities of technical and non-technical stakeholders?
How does a stakeholder communication plan help with stakeholder communication?
What is included in the plan?

You should be able to describe the various stakeholders involved in a given project and identify potential ways to manage and work alongside them. What adjustments may be required to your own ways of working when communicating with technical versus non-technical stakeholders.

Stakeholder Categories and Management Strategies

Stakeholders can be categorised in various ways, such as:

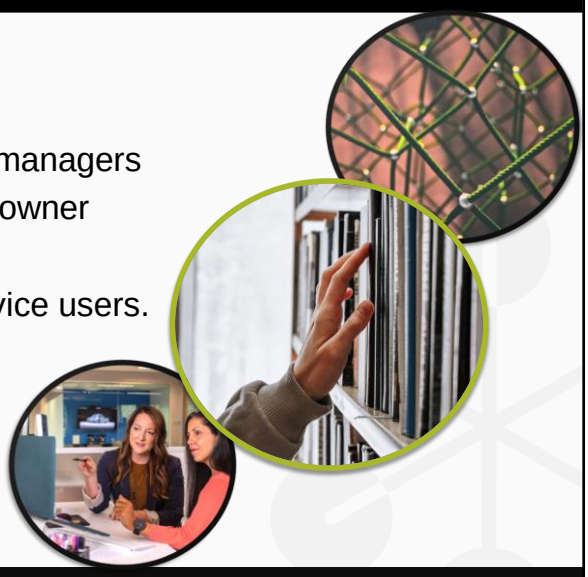
- By their role in the project
- Their level of influence
- Their interest in the project's outcomes

Understanding these categories can help in developing effective management strategies.

- Stakeholder management is a critical ongoing component of enterprise architecture.
- Identify the key stakeholders in a transformation
- Understand their needs and expectations
- Manage their input and involvement to ensure the project's success

Stakeholder Categories

- Business owners and senior managers
- Business sponsor or product owner
- End users or business actors
- Customers and business service users.
- Architect



25/06/2024 Page 119

Consider:

What and who is a stakeholder?
Can you be your own stakeholder?
The importance of categorising stakeholders.
Management styles, communication styles, etc.

Anyone with an interest in or concern about a systems is a stakeholder. Stakeholders can be categorised in various ways.

You should be able to describe the various stakeholders involved in a given project, identify potential ways to manage and work alongside them and identify the adjustments required to your ways of working when communicating with technical versus non-technical stakeholders.

Stakeholder Communication Plan

A stakeholder communication plan outlines how and when you will communicate with stakeholders.

- It helps manage expectations
- Increases stakeholder engagement
- Ensures everyone is on the same page

"It contains a list of decisions about how to engage with the stakeholders in the register. "

"This will often include a RACI matrix as this is key to the type of interaction that will be required"

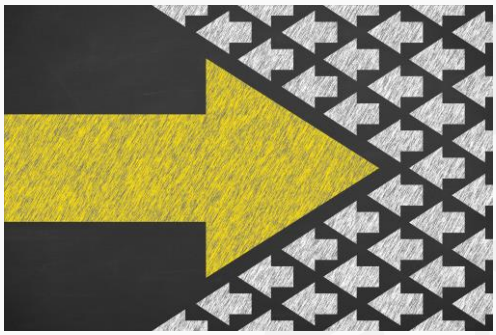
From - Solutions Architecture Foundations, Mark Lovatt. BCS. 2021

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. Discuss types of stakeholders.
3. **Explain specific stakeholder needs.**
4. Discuss how to influence others.
5. Apply a practical customer focus.
6. Discuss the roles and dynamics of a successful team.
7. Discuss social, ethical and economic concerns to the role of an architect.

Stakeholder Needs

- Views
- Viewpoints
- Conflicting requirements



Consider:

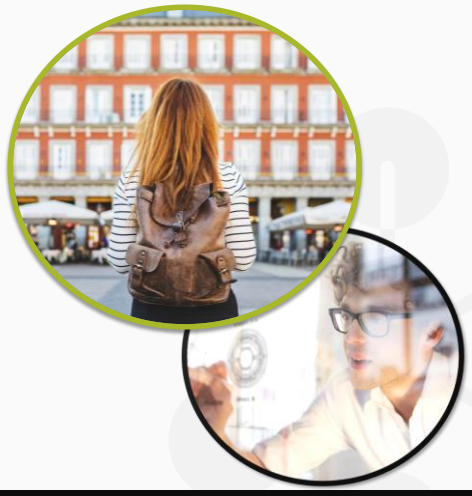
What is the difference between views and viewpoints?
Which types of stakeholders are likely to have conflicting view/viewpoints?

You should be able to address how specific stakeholder concerns, viewpoints can be used to demonstrate the behaviour of certain parts of a solution that are of particular interest. You should e able to describe how different stakeholders may have conflicting ideas of requirements and how to address these.

Views And Viewpoints

A **viewpoint**….. Defines a view's scope (the concerns addressed) and style (documentation conventions). It should be stored for reuse by architects in the same organisation.

A **view** is…an instantiation of a viewpoint, documenting a specific set of concerns/requirements to demonstrate to one or more stakeholders that you are addressing their needs/wants from a relevant perspective.



Consider:

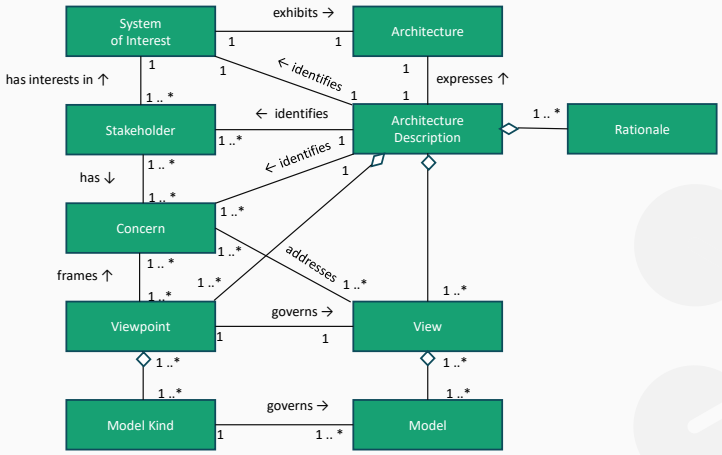
Why would you want to consider the viewpoints of other stakeholders?
What is the benefit to the solution?
How different stakeholders may have conflicting ideas of requirements and how to address these.

Strive to:

Address stakeholder concerns
Minimise the risk of error
Improve stakeholder communication and management
Provide effective perspectives about the problem/opportunity for change

You should be able to identify specific stakeholder concerns, viewpoints that can be used to demonstrate the behaviour of certain parts of a solution that are of particular interest, and describe how different stakeholders may have conflicting ideas of requirements and how to address these.

Single System Viewpoints & Views



25/06/2024 Page 124

The ISO /IEC/IEE 42010:2011 /now replaced with 2022 provides a model for many of the concepts of an architecture from the perspective of a system (not an enterprise).

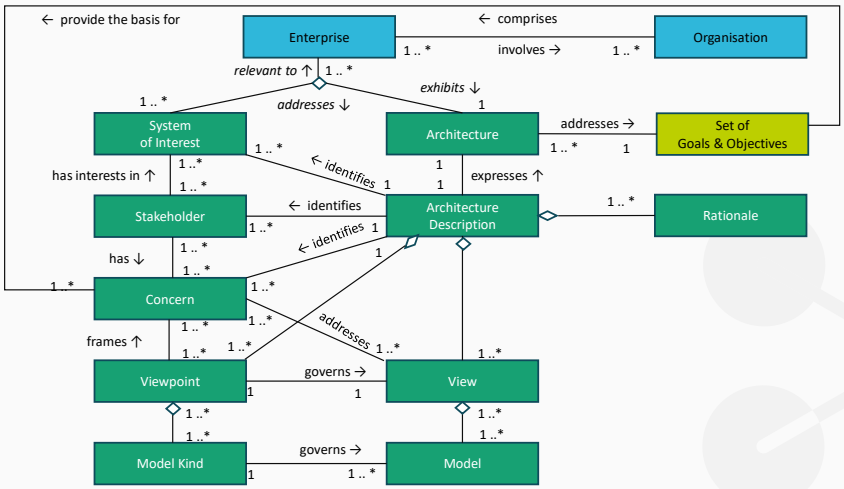
The model assumes that there is a singleton system being described and the relations, optionalities and cardinalities reflect that.

To address specific stakeholder concerns, viewpoints can be used to demonstrate the behaviour of certain parts of a solution that are of particular interest. Candidates should describe how different stakeholders may have conflicting ideas of requirements and how to address these

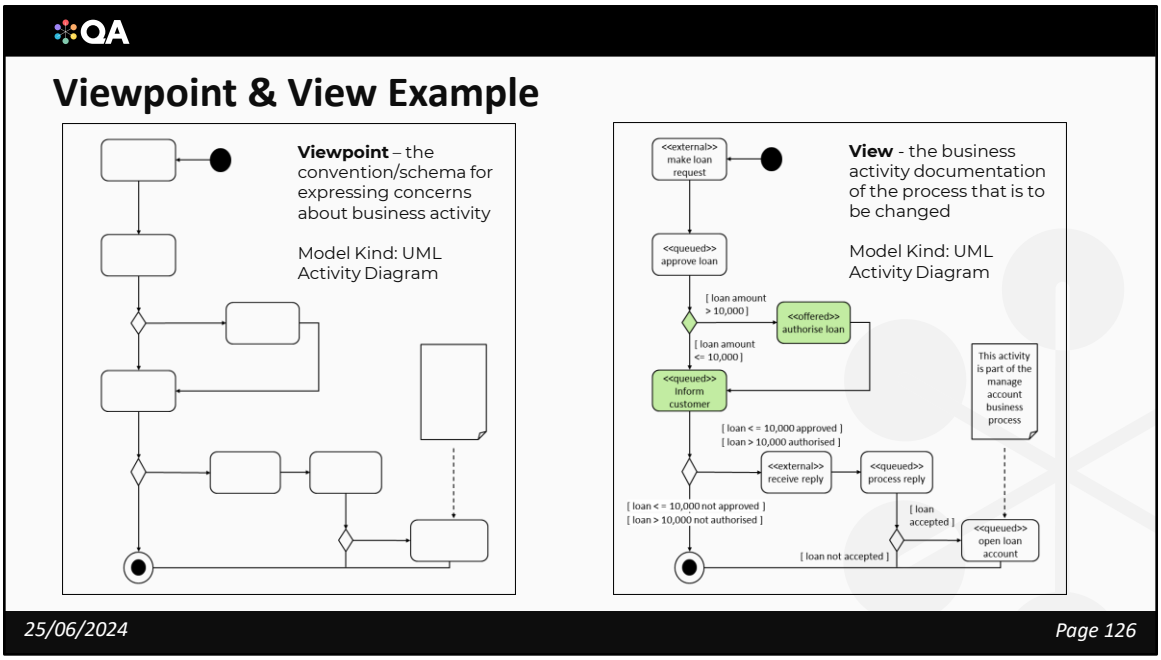
ISO/IEC/IEEE 42010:2022 - Software, systems and enterprise — Architecture description

QA

Enterprise Architecture Viewpoints & Views



An enterprise (or a specific solution) may of course include many systems. This might change several of the other relations, optionalities and cardinalities in this standard viewpoint. The model is extended here to just reflect the fact that there may be many enterprises, and systems that share an architecture and / or have different combinations of systems and architectures.



Views and viewpoints are crucial tools in enterprise architecture. They allow architects to focus on specific aspects of the system, making complex systems more understandable and manageable. They also help in communicating with stakeholders, as each stakeholder might be interested in a different view of the system.

Conflicting Requirements

Strategies for managing conflicting stakeholder requirements:

- Conflicting requirements are a common challenge in projects. Strategies for managing these conflicts can include negotiation, prioritisation, and seeking compromise.

Consideration point:

- Consider a past project or a hypothetical scenario with conflicting stakeholder requirements. What are potential strategies for resolving the conflict and the potential outcomes of each strategy.
- Strategies for managing these conflicts can include negotiation, prioritisation, and seeking compromise.
- Resolve within person/group conflicts first and then in order from the most superior directives and guidelines down to the most subordinate.

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. Discuss types of stakeholders.
3. Explain specific stakeholder needs.
4. **Discuss how to influence others.**
5. Apply a practical customer focus.
6. Discuss the roles and dynamics of a successful team.
7. Discuss social, ethical and economic concerns to the role of an architect.

Influencing Others – 1 (notes)

- Recognising limits
- Contribute to decision-making
- Influencing outside normal reporting lines
- Being a subject matter expert

You should be able to discuss the ways in which an architect must influence others, including using your own subject knowledge and expertise to advise senior staff outside of the normal hierarchy. You should be able to demonstrate your understanding of when to seek help or escalate an issue which is beyond the limits of their authority.

What is the context and status of your architecture role.

Who might a solution or enterprise architect normally report to?
How does the role of the architect balance both business and technical requirements while minimising technical and other types of debt?
How can being a subject matter expert affect an architect's ability to influence others?
What tools/procedures /mechanisms can an architect utilise to help with decision-making (e.g. Architecture Review Boards, Technical Design Authorities, etc.)

Recognise the limits of own authority

This refers to understanding the boundaries of one's role and responsibilities within an organisation.
It involves knowing what decisions one is authorised to make and when it's necessary to escalate issues or decisions to higher levels of management.

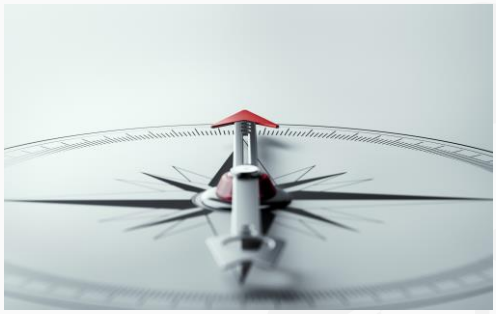
Influencing Scope:

An enterprise architect might have the authority to make decisions about the architecture of a system, but not about the overall business strategy.
Recognising this limit means they would defer to business leaders or stakeholders when strategic business decisions need to be made.

QA

Influencing Others – 2 (notes)

- Recognising limits
- Contribute to decision-making
- Influencing outside normal reporting lines
- Being a subject matter expert



Contribute to business decision making.

Use your expertise and understanding of the business to provide valuable input into business decisions.
Provide technical advice, suggesting improvements, or identifying potential risks.
Contribute to business decision making by advising on the potential impact of different technology choices on the business's strategic goals.

Influencing outside of normal reporting lines.

Facilitate decisions and actions of individuals or groups who are not directly under one's managerial control
Develop persuasion, negotiation, and relationship-building skills
Collaborate with project managers or other stakeholders to prioritise certain architectural requirements, even though these individuals do not report directly to the architect.

Being a subject matter expert

A subject matter expert (SME) is an individual with a deep understanding of a particular topic, technology, process, or area of the business.
SMEs use their expertise to provide advice, solve complex problems, and contribute to decision-making processes.
An enterprise architect could be a SME in cloud computing technologies and use this expertise to guide the business's cloud strategy.
Be clear on when your role as an SME or an Architect is the appropriate one and do not confuse them.

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. Discuss types of stakeholders.
3. Explain specific stakeholder needs.
4. Discuss how to influence others.
5. **Apply a practical customer focus.**
6. Discuss the roles and dynamics of a successful team.
7. Discuss social, ethical and economic concerns to the role of an architect.

Customer Focus

- Identify customer needs (Use case diagrams, user stories)
- UX/UI
- Accessibility
- Metrics of customer satisfaction

Consider:

How can having a practical customer focus positively impact solutions?
How can customer satisfaction be captured? E.g. Surveys, reviews, user experience testing, etc.
What resources are available to ensure accessibility? e.g. Web content Accessibility Guidelines (WCAG), Assistive technologies, Accessibility testing tools, Inclusive design principles, etc.

You should be able to explain the steps and actions that can be taken to show a commitment to the customer throughout the design and development of a solution. Describe tools and metrics that can be used to measure the success of a solution and the level of customer satisfaction.

A stakeholder communication plan outlines how and when you will communicate with stakeholders.

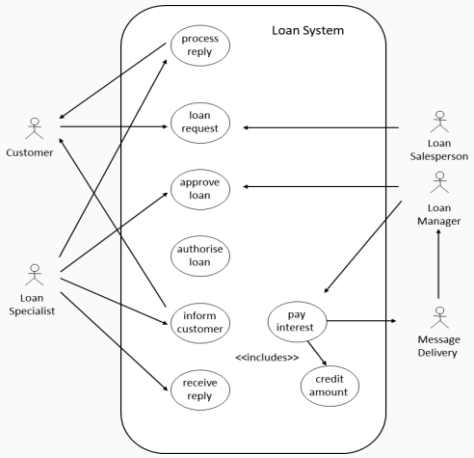
“A use case describes a unit of functionality that provides value to the user of a system.”

From - The Unified Software Development Process, Jacobson, Booch, Rumbaugh Addison Wesley 1999

“A user story describes functionality that will be of value to a user of a system or software.”

From - User Stories Applied, Mike Cohn Addison Wesley 2004

Use Cases - User Focus & Architectural Completeness



Use Case: Pay Interest
Description:
The due interest on all deposit accounts for the last month has been calculated and credited to the accounts and a report produced showing the interest paid.

Actors:
Manager
Printer Queue

Assumptions:
None

Steps:
- The Manager starts the pay interest Use Case (on the 28th of each month)
- The pay interest job is submitted to the system
- For each deposit account held within the deposit account file, the interest due is calculated and the credit amount Use Case is executed for each account that has interest due.
  [Exception: - Deposit Account Value <= 0]
- When all of the accounts have been processed a report is created and stored in the system showing the interest paid on each account and the total amount of interest paid; an e-mail is sent to the to the Manager informing them to access and review the report; and the use case terminates

Alternate Course:
None.

Exceptions:
[Deposit Account Value <= 0]
If a deposit account is found with a balance of zero or less then no interest is calculated, the credit amount use case is not executed for that account, and the report entry for the account will clearly highlight the account value.

Focus on what the user wants/needs and how to architect the system and set of systems that can deliver those want/needs.

Incorporate a diagram and a written description.
Identify the performance and quality attributes for how the system should perform and be tested.
Provide the basis for and a flow into subsequent analysis and design elements.

Use Cases:

Have managed sizes from sagas (large and long running) to singleton simple activities
Can capture the complexity of user requests (when requests are complex) and where needed how those requests generic a control flow of events and what the control requirements are.
Include the subsequent concept of interface objects, control objects and entity objects.
Target an architecture of interacting modules
Have managed sizes from sagas (large and long running) to singleton simple activities.

User Story - The Beginning Of A Conversation

As a **manager in the bank, I would like** the due interest on all deposit accounts to be calculated each month, a report sent to me showing the interest paid on each account, and a final confirmation provided by me before the total is credited to the customer accounts, **in a** secure and reliable manner, **in order to** properly manage deposit account products and be aware of any obvious issues with customer accounts.

Scenarios:

(Describe example scenarios of how this may play out in different situations – if needed)

Acceptance Criteria:

- Only authorised managers can request a calculation/payment run
- An authorised manager must review and sign off the payments before they are finalised
- All accounts must be correctly processed within the overnight processing window
- If any failure occurs the customer accounts must not be in error

Focus on what the user wants/needs.

Provide a written description in the form of a story about the external requests on a system.
Add descriptions of the tests that would show how the system satisfies those user wants/needs (acceptance criteria).

User Stories:

Have managed sizes from epics (large and long) to small elaborated by conversations and annotations based on user discussion.
Leave specific details and scheduling steps to later.
Says nothing about how that could be architected into an effective modular system.

You need to reflect the needs of multiple types of users. Some organisations have had to identify multiple different types of stories to capture all stakeholder wants/needs (e.g. Solution user stories, Process owner stories, Platform provider stories, Engineering stories and Designer stories).

User Interface (UI) User Experience (UX)

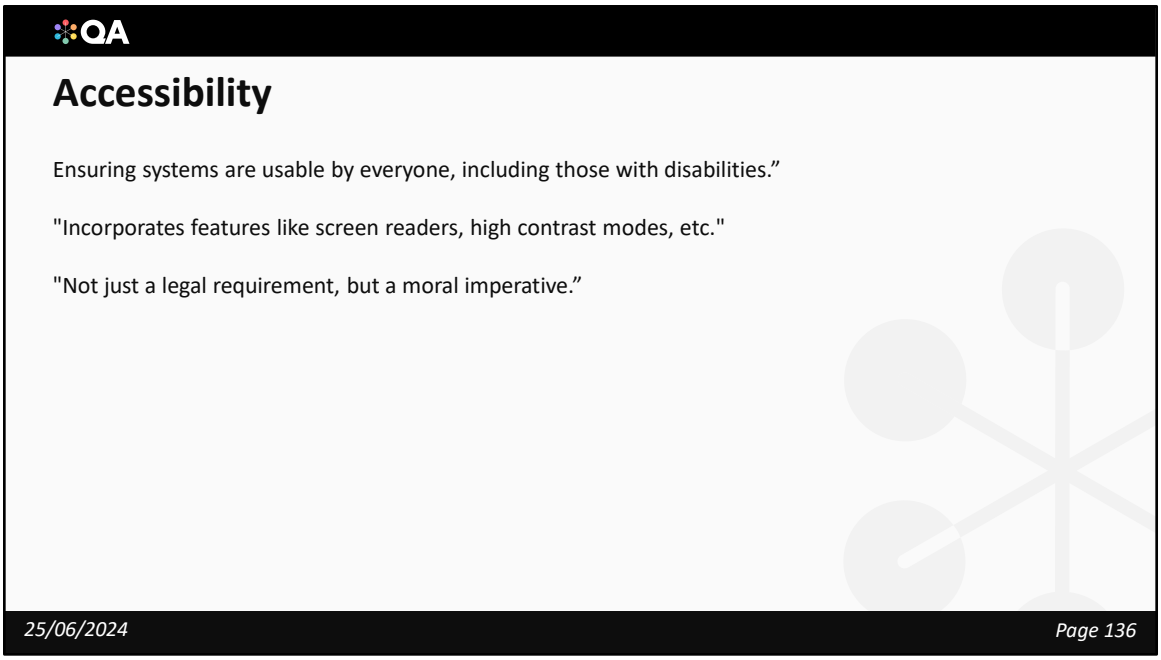
UX (User Experience):

Addresses how the system is experienced. Does the solution reflect the way the user wishes to work and meet their needs?
Aspects: Flow, feeling, effectiveness, satisfaction

UI (User Interface):

Addresses the interaction with the solution, including the look and feel of the solution.
Aspects: Screen, widgets, colours, grouping

Together, they shape the user's journey through the system to deliver the required outcomes..



“Making a website or mobile app accessible means making sure it can be used by as many people as possible.
This includes those with:

impaired vision
motor difficulties
cognitive impairments or learning disabilities
deafness or impaired hearing

At least 1 in 5 people in the UK have a long-term illness, impairment or disability. Many more have a temporary disability. Accessibility means more than putting things online. It means making your content and design clear and simple enough so that most people can use it without needing to adapt it, while supporting those who do need to adapt things.
For example, someone with impaired vision might use a screen reader (software that lets a user navigate a website and ‘read out’ the content), braille display or screen magnifier. Or someone with motor difficulties might use a special mouse, speech recognition software or on-screen keyboard emulator.”

Understanding accessibility requirements for public sector bodies - GOV.UK (www.gov.uk)

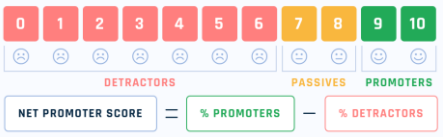
Digital Accessibility specialist group | BCS

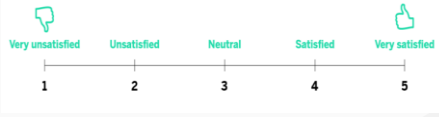
QA

Customer / User Satisfaction

Metrics of customer satisfaction:

- Tools to quantitatively gauge product success."
- "Examples: Net Promoter Score (NPS), Customer Satisfaction Score (CSAT)."
- "Helps in iterative improvement of the system."





25/06/2024 *Page 137*

NPS
The NPS assumes a subdivision of respondents into "promoters" who provide ratings of 9 or 10, "passives" who provide ratings of 7 or 8, and "detractors" who provide ratings of 6 or lower. The net promoter score results from a calculation that involves subtracting the percentage of detractors from the percentage of promoters collected by the survey item. The NPS is typically interpreted and used as an indicator of customer loyalty. In some cases, it has been argued to correlate with revenue growth relative to competitors within an industry, although it has also been demonstrated that NPS scores vary substantially between industries. NPS has been widely adopted by Fortune 500 companies and other organizations.[Proponents of the Net Promoter approach claim the score can be used to motivate an organization to become more focused on improving products and services.[As of 2020, versions of the NPS are now used by two-thirds of Fortune 1000 companies.

Criticism of NPS
While the net promoter score has gained popularity among business executives and is considered a widely used instrument for measuring customer loyalty in practice, it has also generated controversy in academic and market research circles.[3] Scholarly critique has questioned whether the NPS is at all a reliable predictor of company growth.[16] Other researchers have noted that there is no empirical evidence that the "likelihood to recommend" question is a better predictor of business growth than other customer-loyalty questions (e.g., overall satisfaction, likelihood to purchase again, etc.), and that the "likelihood to recommend" question does not measure anything different from other conventional loyalty-related questions.[17]

Net promoter score - Wikipedia

CSAT
Is a generic terms for 'customer satisfaction' and is a key performance indicator that tracks how satisfied customers are with your organisation's products and/or services. CSAT is measured by one or more variation of this question that usually appears at the end of a customer feedback survey usually with a five point scale.

Very unsatisfied / Unsatisfied / Neutral / Satisfied / Very satisfied

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. Discuss types of stakeholders.
3. Explain specific stakeholder needs.
4. Discuss how to influence others.
5. Apply a practical customer focus.
6. **Discuss the roles and dynamics of a successful team.**
7. Discuss social, ethical and economic concerns to the role of an architect.

QA

Dynamic Teams

- Agile roles
- Belbin's team roles
- High performing teams

25/06/2024 Page 139

Consider:

From your experience, what aspects make a successful high-performing team? How can these be encouraged in a team?
How can a dynamic and agile team assist in IT architecture teams?
How can using Belbin's team roles and Agile roles help facilitate team dynamics?

Links:
https://www.belbin.com/about/belbin-team-roles
https://www.bcs.org/articles-opinion-and-research/6-agile-questions-answered-a-simple-guide-to-agile/

You should be able to discuss the key roles within Agile, and the characteristics of high performing teams. Candidates shall be able to categorise team members using Belbin's team role definitions. Different roles contribute to different aspects of the project. Understanding these roles can help in team formation and conflict resolution.

Agile Roles

Product Owner:
- Holds the vision for the product.
- Prioritises the backlog.
- Represents the customer's interests.

Scrum Master:
- Facilitates the team's work.
- Addresses obstacles.
- Ensures Agile practices are followed.

Development Team:
- Cross-functional group.
- Delivers high-quality increments.
- Holds daily stand-up meetings.

Agile is a collaborative approach. Each role is distinct but interconnected.

The success of Agile projects depends on clear communication and understanding of these roles.

You should be able to discuss the key roles within Agile, and the characteristics of high performing teams.

https://www.bcs.org/articles-opinion-and-research/6-agile-questions-answered-a-simple-guide-to-agile/

Belbin’s Team Roles

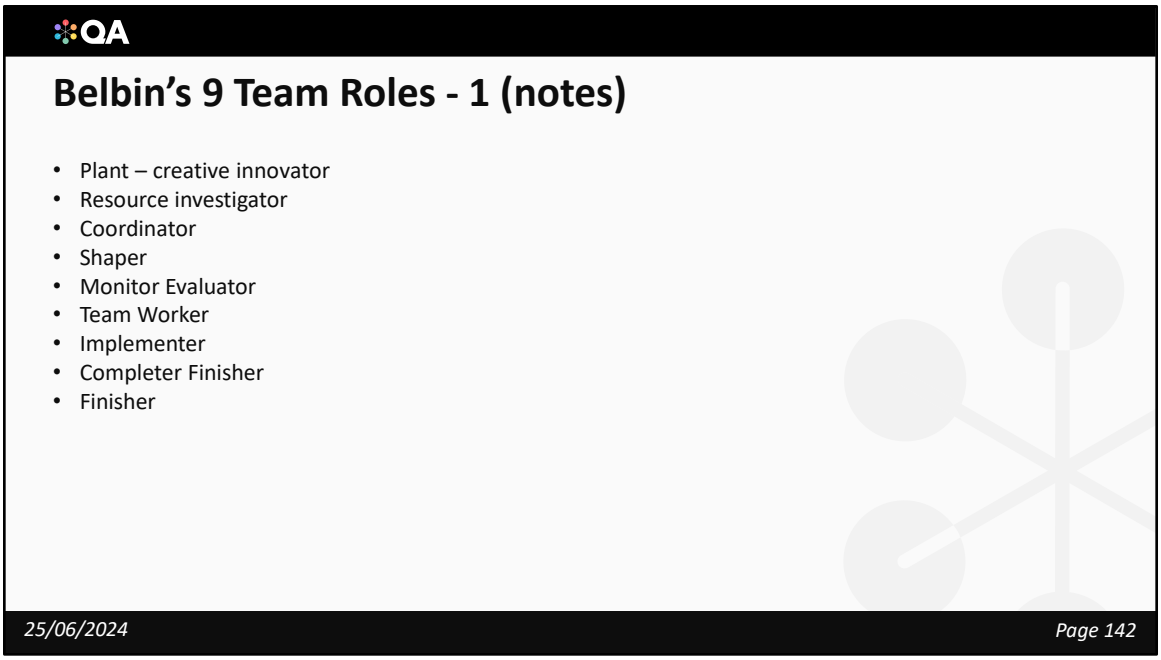
Belbin identified 9 team roles that are essential for a team's success.

Examples: Implementer, Coordinator, Shaper, Plant, etc.“

- Different roles contribute to different aspects of a project.
- Understanding these roles can help in team formation and conflict resolution.
- In a balanced team, you'd ideally have a mix of these roles to ensure a well-rounded approach to problem-solving and project management.

You should be able to categorise team members using Belbin’s team role descriptions.

https://www.belbin.com/about/belbin-team-roles



1. Plant (PL)
The Plant is the creative innovator who comes up with new ideas and approaches. They thrive on praise but can be introverted and tend to ignore given parameters and constraints.

Strengths: Innovative, imaginative, free-thinking. Generates original ideas and solves difficult problems.
Potential Weaknesses: Might ignore incidentals, and can be too preoccupied to communicate effectively

2. Resource Investigator (RI)
The Resource Investigator is outgoing and enthusiastic, exploring opportunities and developing contacts.

Strengths: Outgoing, enthusiastic. Explores new opportunities and develops contacts.
Potential Weaknesses: Over-optimistic and can lose interest once the initial enthusiasm has passed.

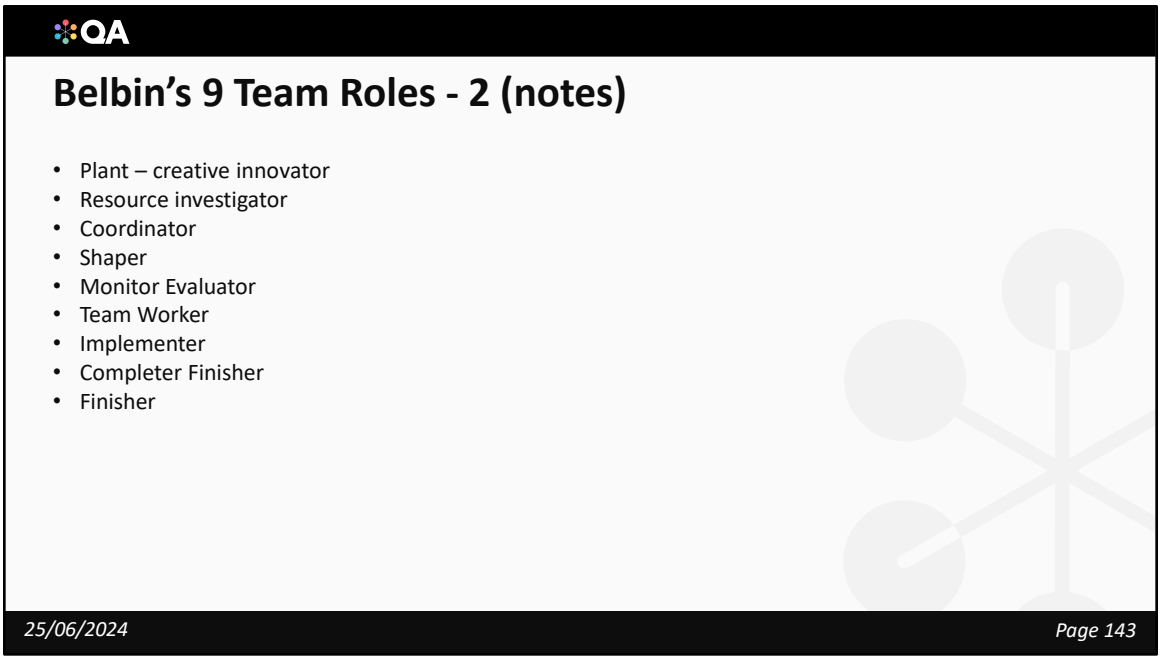
3. Coordinator (CO)

The Coordinator is a mature, confident figure who can clarify group objectives, promote decision-making, and delegate effectively.

Strengths: Mature, confident, identifies talent. Clarifies goals.
Potential Weaknesses: Can be seen as manipulative and might offload their share of the work.

4. Shaper (SH)
The Shaper is dynamic and loves to challenge the team. They drive others to overcome obstacles.

Strengths: Challenging, dynamic, thrives on pressure. Has the drive and courage to overcome obstacles.
Potential Weaknesses: Can be argumentative and may offend people's feelings.



5. Monitor Evaluator (ME)
The Monitor Evaluator is strategic and discerning, seeing all options and judging accurately.

Strengths: Sober, strategic, discerning. Sees all options and judges accurately.
Potential Weaknesses: Sometimes lacks the drive and ability to inspire others and can be overly critical.

6. Teamworker (TW)
The Teamworker is the glue that holds the team together, helping ensure that the team remains cohesive.

Strengths: Cooperative, perceptive, diplomatic. Listens and averts friction.
Potential Weaknesses: Can be indecisive in crunch situations and tends to avoid confrontation.

7. Implementer (IMP)
The Implementer turns the team's ideas into actions and plans.

Strengths: Practical, reliable, efficient. Turns ideas into actions.
Potential Weaknesses: Can be a bit inflexible and slow to respond to new possibilities.

8. Completer Finisher (CF)
The Completer Finisher ensures thorough, timely completion. They have a strong inward sense of the need for accuracy.

Strengths: Painstaking, conscientious, anxious. Searches out errors. Polishes and perfects.
Potential Weaknesses: Can be inclined to worry unduly and reluctant to delegate.

9. Specialist (SP)
The Specialist brings specialised knowledge to the team.

Strengths: Single-minded, self-starting, dedicated. Provides knowledge and skills in rare supply.

Potential Weaknesses: Tends to contribute only on a narrow front and can dwell on technicalities.

Professionalism In Architecture

1. Discuss compliance with professional standards.
2. Discuss types of stakeholders.
3. Explain specific stakeholder needs.
4. Discuss how to influence others.
5. Apply a practical customer focus.
6. Discuss the roles and dynamics of a successful team.
7. **Discuss social, ethical and economic concerns to the role of an architect.**

Social, Ethical And Economic Concerns

- Ethics
- Green IT
- Corporate social responsibility
- Sustainable design
- Inclusive and accessible design

Consider:

How important are social and ethical concerns in your organisation?
How can these concerns negatively impact an organisation?
What tools/methods/strategies are available to assist in the implementation of sustainable business approaches?

You should be able to discuss the practical steps that can be taken to demonstrate consideration of social, ethical and economic issues in architecture. The importance of inclusive and accessible design should be able to be explored with examples given of how this is implemented.

https://www.bcs.org/articles-opinion-and-research/the-politics-of-ethics-and-sustainability/

Ethics & Green IT

Definition of Ethics: "Moral principles that govern behaviour or the conducting of an activity."

Ethical Considerations in Architecture: Respecting user privacy, ensuring data security, transparency in decision-making, and avoiding bias in AI models.

Balancing Business and Ethics: While business goals are vital, ethical considerations should never be compromised. For instance, while data can be a valuable asset, user consent and data protection are paramount.

What is Green IT?: "Practices and technologies that reduce the environmental impact of IT operations."

Benefits: Reduced energy consumption, minimised e-waste, and a positive brand image.

Steps for Green IT: opt for energy-efficient hardware, promote server virtualisation, and recycle or refurbish old equipment.

Corporate Social Responsibility

Definition of CSR: "A self-regulating business model that helps a company be socially accountable to itself, its stakeholders, and the public.“

Role of Enterprise Architects: Architects can champion sustainable solutions, promote ethical tech practices, and influence CSR initiatives.

Example: Tech giants like Microsoft have committed to being carbon negative by 2030, emphasising the role of sustainable tech solutions

Sustainable Design

- Sustainable design is about creating tech solutions that have a minimal negative impact on the environment.
- This can involve choosing sustainable materials, optimising for energy efficiency, and considering the entire lifecycle of a product. Avoiding built in obsolescence
- **Benefits:** Long-term cost savings, reduced carbon footprint, and enhanced brand reputation.
- **Strategies:** Optimise software for energy efficiency, choose sustainable materials for hardware, and consider the product's entire lifecycle.

Inclusive & Accessible Design

- Inclusive design ensures that products are usable by as many people as possible, while accessible design specifically caters to those with disabilities.
- Both are crucial in ensuring that technology is equitable and usable for all.

Inclusive Design:
Designing products to be usable by as many people as possible without the need for adaptation.

Accessible Design:
Design that ensures products and services are usable by those with disabilities.

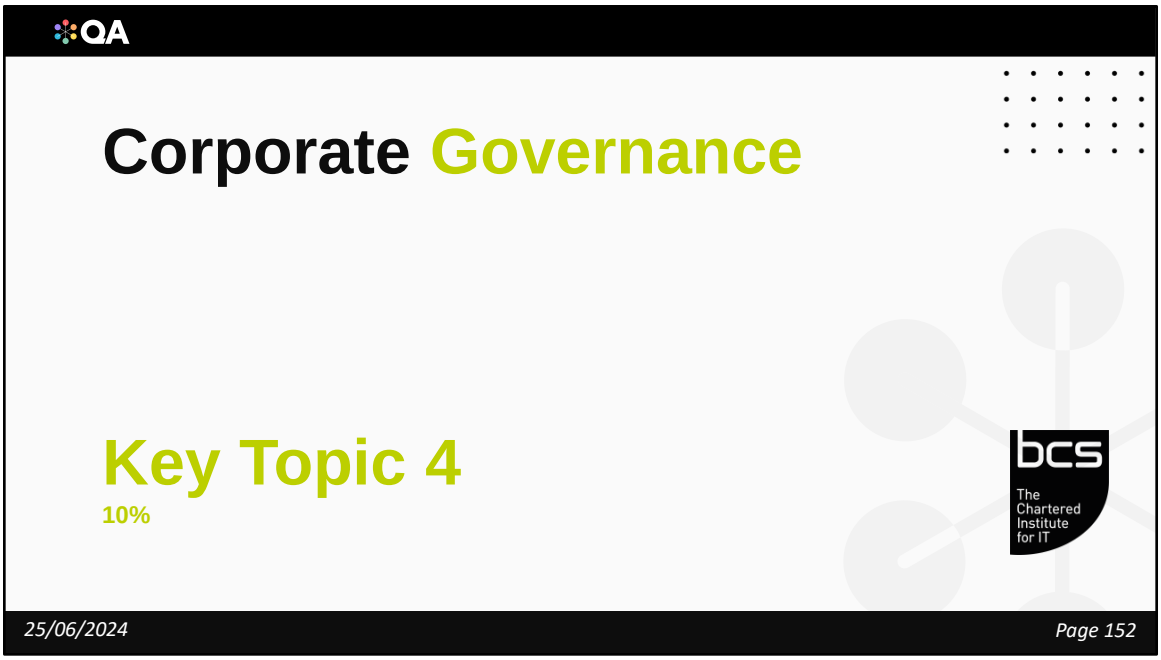
Importance:
With billions of users worldwide, it's crucial that tech solutions cater to everyone, regardless of their physical or cognitive abilities.

Please open your workbooks to topic 3

Learn. To Change.

Section 04

Corporate Governance



Elements to be addressed:

Corporate Governance

What is governance
Explain corporate governance and the impact it has on the role of an architect.
Discuss key architecture governance concepts, roles, and artefacts.
Explain the importance of risk management to the role of an architect.

The Course Schedule

DAY 2 09:00 - 16:30

3.1 Refresh on Professionalism in Architecture
3.2 Practice Exercises

4.1 Refresh on Corporate Governance
4.2 Practice Exercises

5.1 Refresh on Architectural, Processes, Tools and Vision
5.2 Practice Exercises

Corporate Governance

1. **Corporate Governance.**
2. Corporate governance and the impact it has on the role of an architect.
3. Key architecture governance concepts, roles, and artefacts.
4. The importance of risk management to the role of an architect.

Governance means controlling activity and decision making to ensure that the change delivered matches the specification agreed with the business.... through the use of processes and organisational structures.

Corporate Governance

1. Corporate Governance.
2. **Corporate governance and the impact it has on the role of an architect.**
3. Key architecture governance concepts, roles, and artefacts.
4. The importance of risk management to the role of an architect.

Corporate Governance and Architecture

- Enterprise architecture governance
- Solution architecture governance
- Policy and process
- Risks of non-compliance

Consider:

What methods/tools are available to solution/enterprise architects to reduce instances of non-compliance? E.g. Architecture review boards, Authorities, etc.
How to organisational policies and processes impact the architecture? E.g. influencing architectural principles.
What are the potential risks of non-compliance (both internal and external)? E.g. Non-adherence to GDPR.

You should be able to demonstrate your understanding of corporate governance, and the direct impact it has on the role of an architect. The need for consideration of and compliance with various policies and processes should be understood, as well as the potential impact of noncompliance.

Architecture Governance Aspects

- Corporate Governance
- Technology Governance
- Information Technology Governance
- Architecture Governance

Architecture governance ensures that solutions address the relevant goals, objectives, requirements and constraints and satisfy the relevant controlling directives at the relevant levels of composition: enterprise, segment/domain and capability/solution.

These are usually expressed in a series of directives (principles, policies, standards, best practises and general business rules).

Note that each individual solution may need to work with many other solutions that overlap and interact at the capability, segment/domain and enterprise levels. The architect must navigate between the levels.

From - The TOGAF® Standard

Solution Implementation Governance

"Governance must include assuring that products that are delivered match the specification but also account for the fact that the specification may need to be refined or altered during the delivery of the solution."

"For solution architecture, the business is represented by stakeholders who agree the specification for the solution and are also involved in the governance of its delivery."

From - Solutions Architecture Foundations, Mark Lovatt. BCS. 2021

Policy & Processes

Policies:

Official statements that outline specific governance requirements, standards, or procedures.

Processes:

Processes are systematic series of actions or steps taken to ensure governance.
Processes ensure consistency, predictability, and adherence to standards.

Risks Of Non-Compliance

External Legal And Regulatory Risks:
Falling foul of the national, supranational and legislative demands placed on the enterprise with consequent impact.

Internal Management And Control Risks:
Company policy, codes f conduct and practice sometimes also encapsulate a form of contractually defined sanction or penalty that has the force of law.

Operational Risks:
Disruptions, inefficiencies, and potential failures.

Reputational Risks:
Damage to the company's public image and trust.

Financial Risks:
Fines, penalties, and potential lawsuits.

Governance Concepts

- Conformance level
- Compliance level
- Dispensation

25/06/2024 Page 162

Consider:

How to act within policies, rules, laws, etc.
The level of governance within an organisation (risk assessment, compliance, etc.)
Exemptions or alterations from established principles, standard, etc.

You should be able to discuss the application and use of the key concepts in architecture governance, and their purpose. Understand the role of the architecture board and the accountability of the governing architect should be related to an organisational context.

Conformance, Compliance and Dispensation

Conformance Level:
The extent to which a solution or process aligns with established architectural standards.

Compliance Level:
The degree to which a solution or process meets legal, regulatory, and organisational requirements.

Dispensation:
An official exemption from adhering to certain governance standards, granted under specific circumstances.

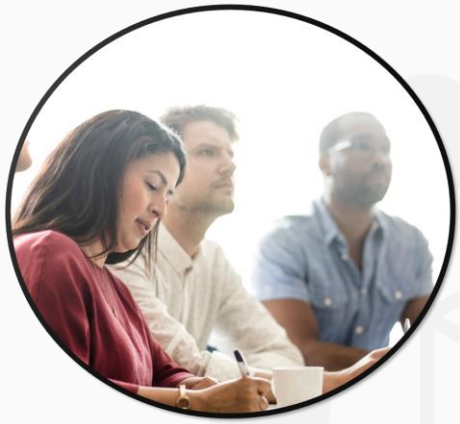
Corporate Governance

1. Corporate Governance.
2. Corporate governance and the impact it has on the role of an architect.
3. **Key architecture governance concepts, roles, and artefacts.**
4. The importance of risk management to the role of an architect.

QA

Governance Roles

- Architecture board
- Governing architect
- Design authority



25/06/2024 Page 165

Consider:

Which governing bodies are responsible for overseeing and making strategic decisions regarding the design, implementation, and evolution of an organization's information technology infrastructure and systems.

How a senior-level professional who establishes and enforces architectural standards, policies, and guidelines to ensure coherence and alignment within an organization's technology landscape
Which designated body or group should be responsible for evaluating, approving, and guiding architectural decisions to ensure consistency and adherence to established design principles

You should be able to discuss the application and use of the key concepts in architecture governance, and their purpose. Including the role of the architecture board and the accountability of the governing architect related to an organisational context.

Architecture Board

A governing body responsible for overseeing architectural decisions, standards, and practices.

Purpose:

Oversee and guide architectural initiatives.

Responsibilities:

Set standards, review designs, grant dispensations, and ensure alignment with business goals.

Context:

The board's decisions have a ripple effect, influencing technological direction, investments, and business outcomes.

Governing Architect and Design Authority

Governing Architect:

- The individual accountable for ensuring that architectural solutions align with governance standards and business objectives.
- Ensures alignment, and liaises between business and IT.

Design Authority:

- A role or entity responsible for ensuring that the design of a solution adheres to architectural and governance standard
- Validates solution designs, ensures compliance, and acts as a governance gatekeeper.

Governance Artefacts

- Compliance review
- Compliance checklist
- Capability maturity model

Governance Artefacts

Compliance Review:

A formal assessment to determine if a solution meets governance standards.

Compliance Checklist:

A tool used to evaluate and ensure that a solution adheres to required governance criteria.

Capability Maturity Model:

A framework used to assess the maturity and capability of organisational processes.

Corporate Governance

1. Corporate Governance.
2. Corporate governance and the impact it has on the role of an architect.
3. Key architecture governance concepts, roles, and artefacts.
4. **The importance of risk management to the role of an architect.**

Risk Management

- Risk appetite
- Documenting Risk

Consider:

How does the type/industry of organisation affect risk appetite?
How can an organisation's appetite for risk influence architectural decisions?
What are the risks associated with granting (or not granting) exemptions or alterations from established principles, standards, etc.
Who is responsible for risk management? Note: This may vary between different organisations.

You should be able to discuss the importance of understanding the organisation's risk appetite and the need for documentation to record any risks that have been identified and mitigations which have been implemented.

https://www.bcs.org/articles-opinion-and-research/risk-assessment/

QA

Risk Management

Typically comprises:

- Risk Identification: identifying suitable events.
- Risk Evaluation: assessing the events identified for probability and impact, and so rating the importance of the events.
- Risk Response: deciding on a suitable response to the risks and implementing the response.
- Risk Documentation: recording the risk parameters.
- Risk Monitoring and Control: periodic and ad-hoc monitoring of risk, optimisation of responses, withdrawal of risks due to events that cannot now occur, change in evaluations/responses …

“Coordinated activities to direct and control an organisation with regard to risk.” ISO 31000

https://www.iso.org/iso-31000-risk-management.html/

Risk Appetite

Risk Appetite:

- The level of risk an organisation is prepared to accept, every organisation has a threshold for the amount of risk it's willing to take on.
- For architects, it's crucial to understand this appetite. It guides decision-making, ensuring solutions proposed don't exceed the organisation's comfort zone.
- It's essential for architects to align solutions with the organisation's risk tolerance levels

Documenting Risk

Documenting Risk:

Documenting risks ensures transparency and helps in tracking and mitigation.

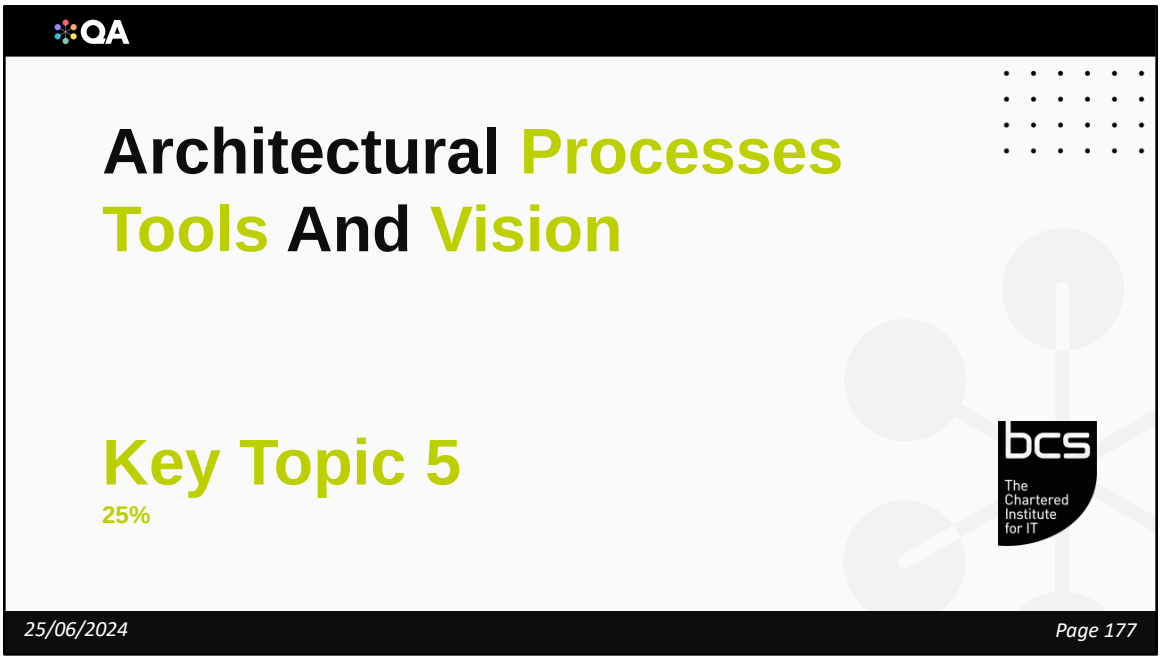
- A well-documented risk aids in informed decision-making.
- Risk management isn't a one-time task. It's a continuous process.
- From spotting potential risks to assessing their impact, devising strategies to counter them, and continuously monitoring the landscape - it's a cycle that architects are deeply involved in.

Key Artifact: Risk Register / Risk Mitigation Worksheets

Please open your workbooks to topic 4

Section 05

Architectural Processes Tools And Vision



Elements to be addressed:

Architectural Process, Tools, and Vision

Discuss the practical application of the solution architecture framework.
Discuss three varieties of the Software Development Lifecycle (SDLC).
Discuss the design of a target solution architecture.
Discuss the use of common architecture models.
Discuss the use of common information systems modelling techniques.
Explain different types of business requirements.
Explain a method of preparing a gap analysis.
Discuss a business case for presentation.

The Course Schedule

DAY 2 09:00 - 16:30

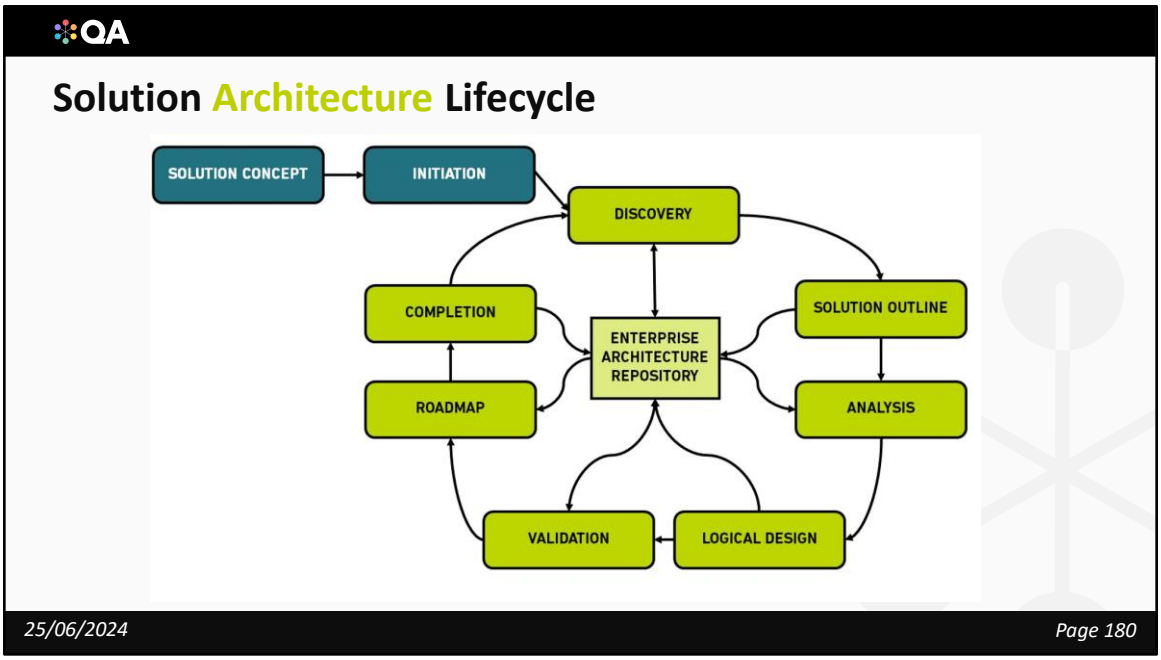
3.1 Refresh on Professionalism in Architecture
3.2 Practice Exercises

4.1 Refresh on Corporate Governance
4.2 Practice Exercises

5.1 Refresh on Architectural, Processes, Tools and Vision
5.2 Practice Exercises

Architecture Process Tools and Vision

1. **The practical application of the solution architecture framework**.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. Gap Analysis.
8. A business case for presentation.



From – Solution Architecture Foundations, Mark Lovatt, 2022

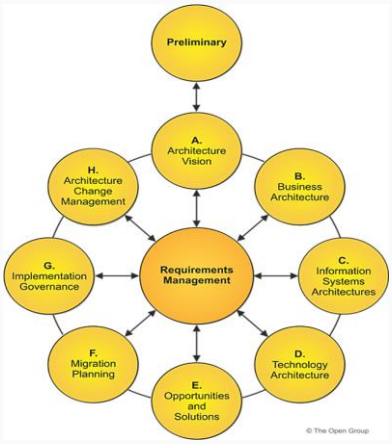
Consider:

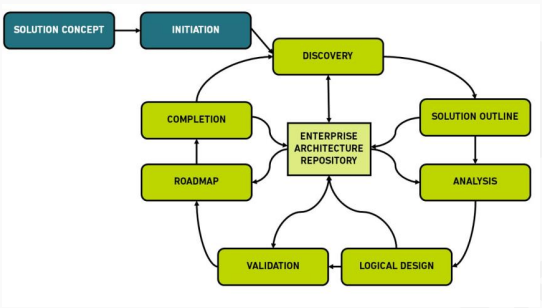
The solution architecture lifecycle.
The roles and responsibilities involved in the lifecycle.
How would the type of solution influence the artefacts used?
How would the roles of enterprise and solution architect differ?

You should be able to describe the stages of the solution architecture frame

Initiation: is the starting point of the process where the business authorises solution architecture work.
Discovery: investigating the situation, engaging with stakeholders and gathering inputs.
Solution Outline: describing one or more solutions in high-level terms to stimulate feedback from the business.
Analysis: deciding on one or two solution options to take forward and specifying the scope of change to be undertaken.
Logical Design: developing a single model of the solution and its components or building blocks.
Validation: testing and assuring the design addresses the concerns of stakeholders whilst maximising positive impact and disruption for the business.
Roadmap: a structured delivery plan with stakeholder priorities and timelines; this is a key document for the governance of the implementation of the solution.
Completion: managed by programme and project management working with solution development teams and service management who will implement and deploy the solution, during which time solution architecture retains a governance role.

Solution Architecture Lifecycle





25/06/2024 *Page 181*

From – The TOGAF® Standard

Consider other architecture change methods such as The TOGAF ADM. This shows a cycle of change for all architecture deliverables at all composition viewpoints; enterprise, segment, capability and solution.

Vision
Solution Definition
Solution Implementation Planning
Solution Implementation & Governance

Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. **Varieties of the Software Development Lifecycle (SDLC).**
3. The design of a target solution architecture.
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. Gap Analysis.
8. A business case for presentation.

QA

Software Development Lifecycle

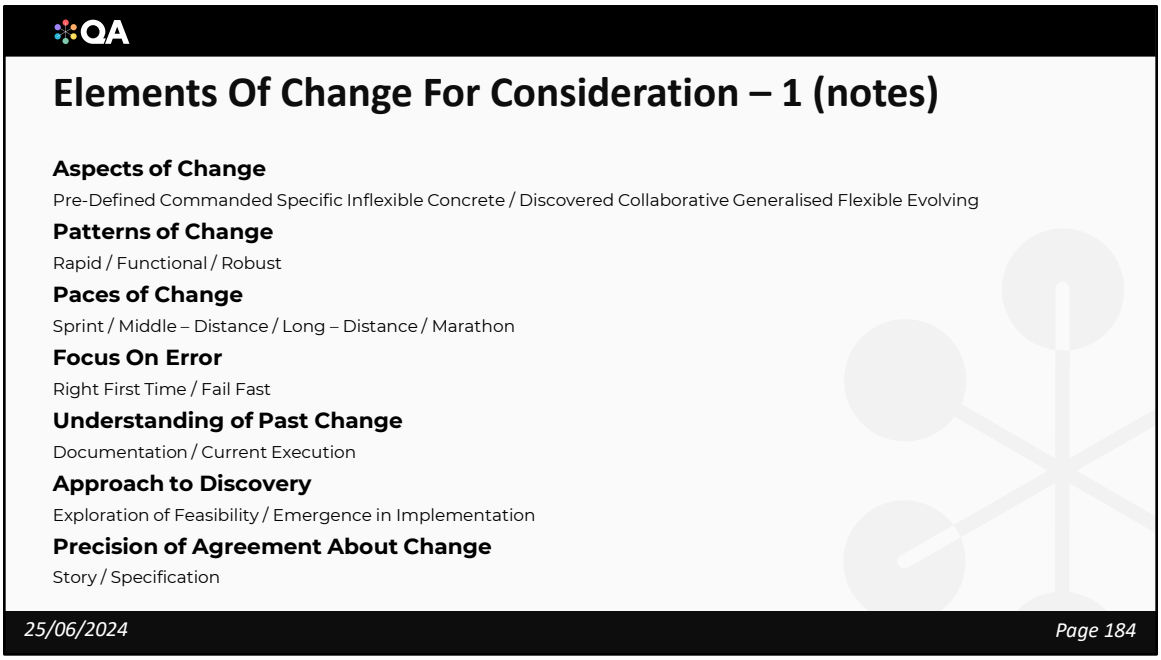
- Waterfall
- Iterative
- Agile

25/06/2024 Page 183

Consider:

What are the similarities and differences between the three varieties of SDLC listed?
How might the type of change/solution influence the use of SDLC?
Are there more lifecycles to consider?

You should be able to discuss the practical application of each of the varieties of the SDLC and be able to critically compare the suitability of each lifecycle to a given scenario.



Aspects: Change and development can follow many different paths. At the extremes of the distribution of change types are aspects such as:

Pre-Defined Commanded Specific Inflexible Concrete
Discovered Collaborative Generalised Flexible Evolving

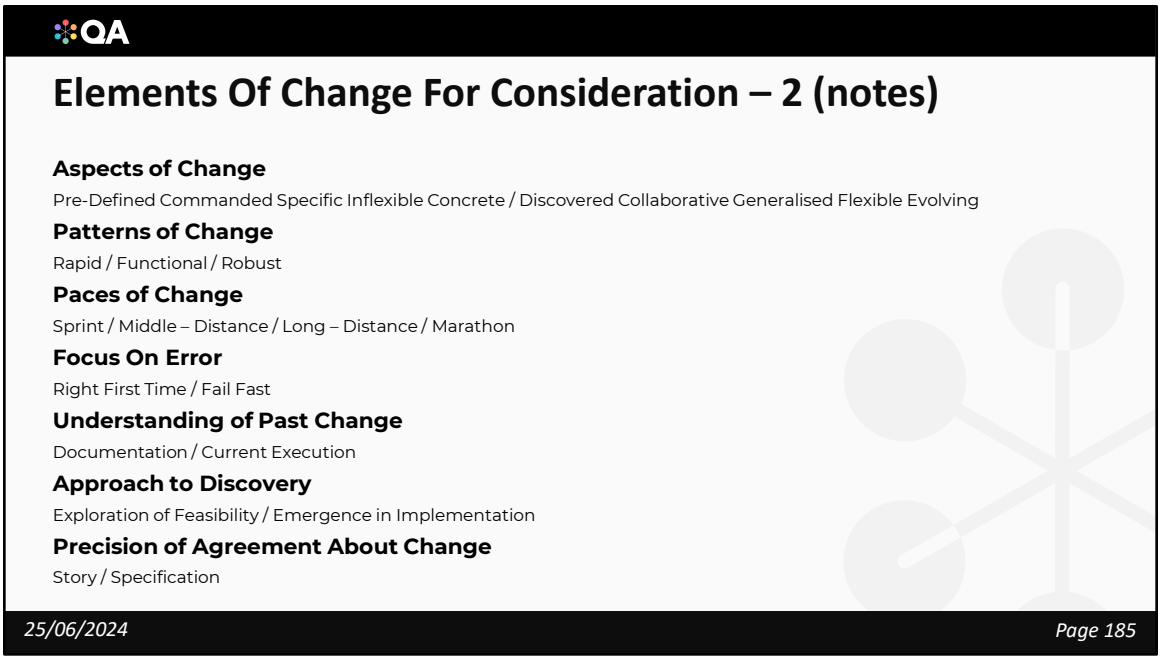
Patterns: There is no right or best approach to change prior to an understanding of the nature of each specific change cycle. We can consider that some situations require different approaches: (taken from the TOGAF® Series Guide – Enabling Enterprise Agility")

Rapid - Fast exploratory change with no guarantees of success
e.g. Latest technology introduction in low-risk live environments / Attempts to reach critical time milestones in low-risk environments or desperate situations.

Functional - Frequent delivery of working functionality towards an aspirational target that is itself changing
e.g. Generalised noncritical business processes

Robust - Careful implementation of large-scale complex elements that must pre-agree of their shape and interactions in order to be deployable and work at first use
e.g. Nuclear Power Stations and Aircraft Control systems.

Paces: There is no one specific pace of change, different speeds deliver properties of progression and most complex change requires all four elements not blind adherence to only one pace.
As an example, In athletics there are four paces :



Sprint – 100% effort and resource use in the shortest time with little visibility of what is around you.

Middle - Distance – fast progress but balance of resource use, smoothness of flow, powerful finish in response to changing conditions around you, but one mistake may be fatal.

Long - Distance – multiple speeds but usually careful transitions while husbanding resources, there may be time to make up for mistakes but over commitment in a short time is usually fatal.

Marathon – consistent pace with gradual occasional changes, there is time to correct for mistakes, resources must be kept for the later stages, environmental conditions may change completely.

Focus: There is no one approach to error/correctness, that depends on the nature of the problem:

Right First Time - Ensure that what is created performs as closely to what is needed and planned.
Fail Fast - Find out when what is aspired to cannot perform as planned and needed.

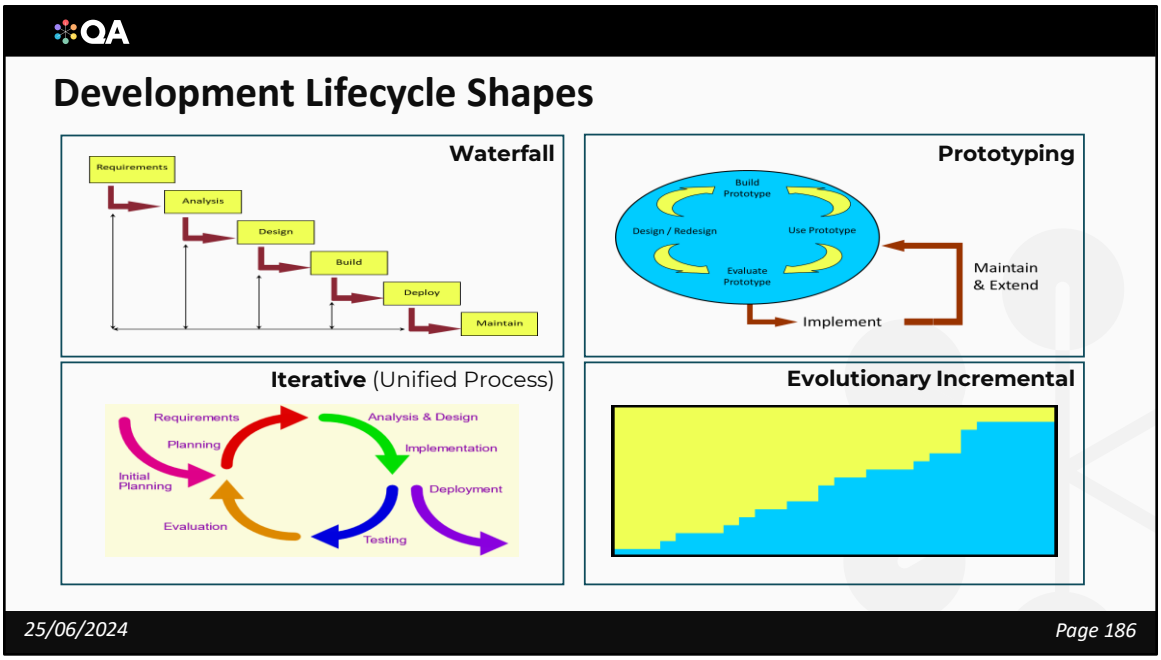
Approach: Uncertainty can be reduced prior to build and/or during build:

Exploration of Feasibility - Allows significant restart and rework to move towards an effective solution reducing the risk of failure before commitment of resources to implement.
Emergence in Implementation - Enables change while finalising and deploying but increases risk of failure and waste of resources in implementation..

Precision: How much detail and pre-decision do we need from our sponsors / users:

Specification vs Story
Planned Design vs Evolved Design

In any large-scale complex situation, all of these are likely to be needed.



Waterfall

A solution development process that is sequential, such as that described by Royce. The sequence of stages is usually analysis, design, build, test and roll out. The idea is that engineers proceed from one kind of work to the next without significant iteration or parallelism between stages.

Prototyping / RAD

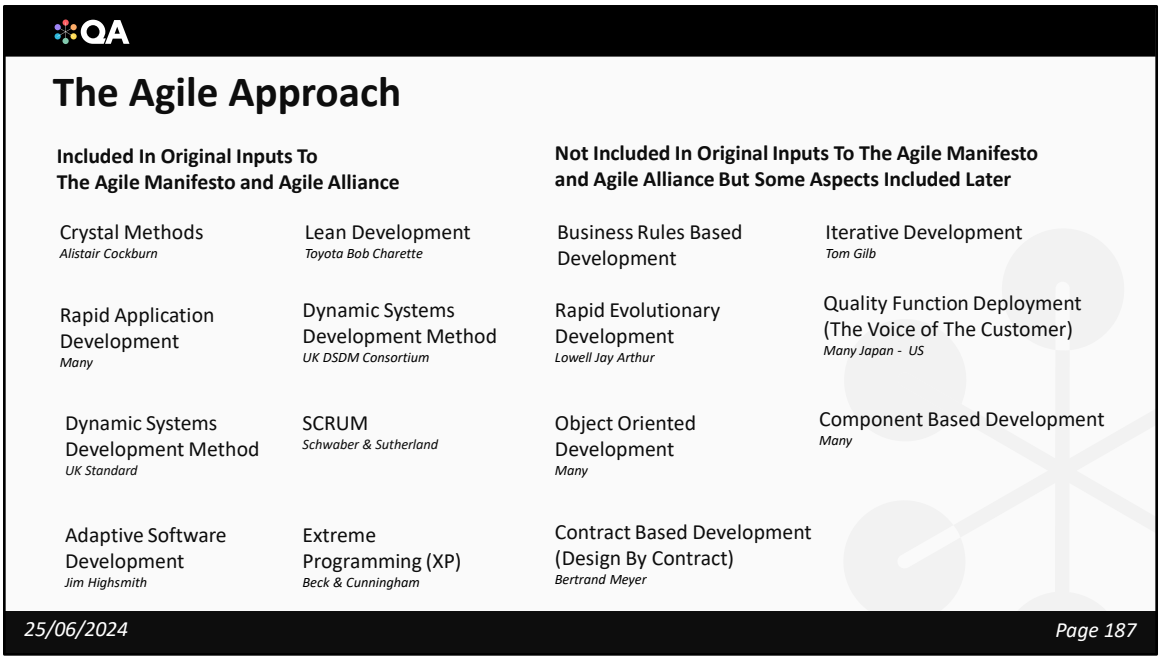
A solution development process, such as DSDM, that provides a framework for building and maintaining systems to meet tight timescales through the use of incremental prototyping in a controlled project environment.

Iterative

A solution development process such as the Unified Process, that proceeds by imitations of change of various sizes, meaning that a working subset of the full solution is delivered as early as possible. Iterations are profiled for different phases; inception, elaboration, construction and transition.

Evolutionary / Iterative

A solution development process, such as that described by Tom Gilb in Principles of Software Engineering Management, that asks how little can be delivered not how much. It focuses on a series of steps which lead to complete long term goal satisfaction as a number of steps are sketched the focus is mainly on the next step.



Agile Development

A solution development process that is not only iterative, but also flexible about the requirements, the solution and the process being followed. There is no formal process it is rather an aggregation of techniques, characterised by short-cycle iterative development, early testing for usability and performance, and flexible requirements. User involvement and feedback is a mandatory prerequisite in agile development as it is in all lifecycles of development.

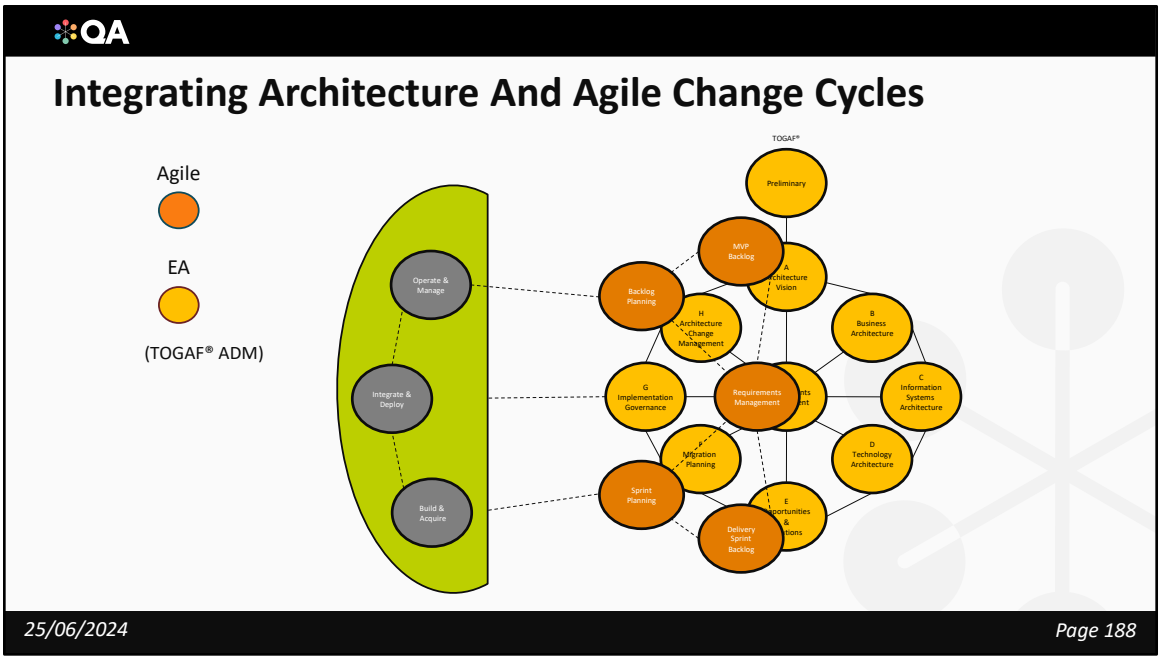
Originally constructed at Snowbird Utah in 2001 and formally described by the Agile Alliance in 2002.

The Agile Manifesto

“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

That is, while there is value in the items on the right we value the items on the left more.”



Enterprise architecture as described by the Open Group (the TOGAF® Standard) maps exactly onto the agile backlogs:

Product – Vision
Minimum Viable Product – Vision Version
Delivery Sprint – Work Package
Transition Architecture – Release Plan

And they both include:

Continual review of changing requirements
Very strong emphasis on stakeholder, needs, priorities and decisions
Emphasis of delivery of units of value
Encompass multiple iterative units of change at all levels of detail
Refactoring / backtracking and continuous learning and improvement.

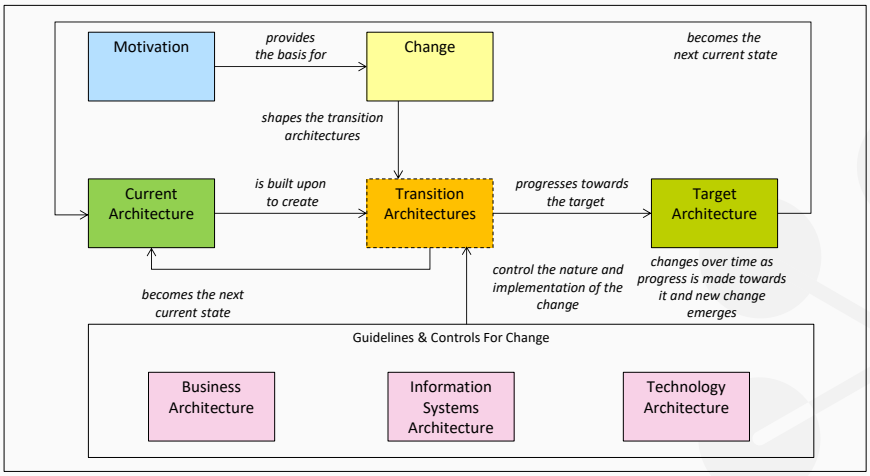
https://pubs.opengroup.org/togaf-standard/guides/enabling-enterprise-agility/index.html

Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. Varieties of the Software Development Lifecycle (SDLC).
3. **The design of a target solution architecture.**
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. Gap Analysis.
8. A business case for presentation.

QA

Target Solution Architecture

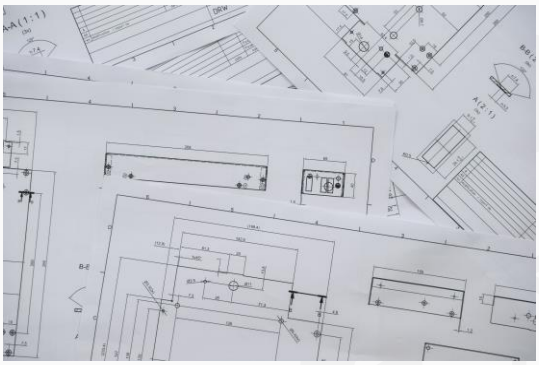


A target solution architecture is evolving through a series of change transitions in response the changing motivations of its stakeholder.

QA

Designing Target Solution Architecture

- Options evaluation
- Trade off analysis
- Modelling
- Design patterns



Consider

What approaches of evaluation/analysis are available? What have you used?
How might different projects influence your approach to evaluation/analysis?

You should be able to discuss the activities undertaken by the architect during design, including how to analyse the possible options and suitability of design patterns.

Options Evaluation

It's not about finding the 'perfect' solution, but the 'right' one that aligns with organisational needs and constraints.

Options Evaluation involves assessing different architectural solutions to determine the best fit.

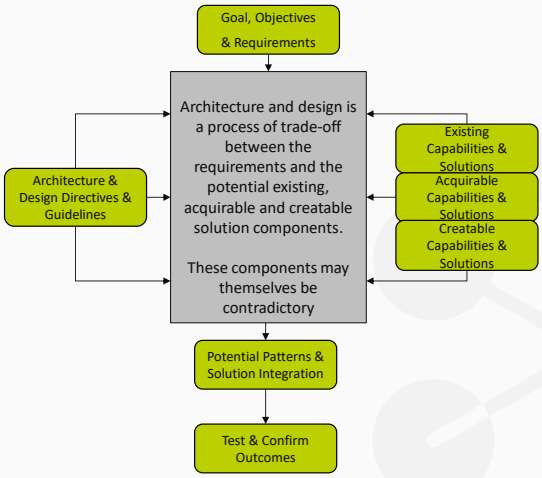
Use criteria such as:

- Cost
- Scalability
- Performance
- Integration
- Maintainability

Trade Off Analysis

As architecture and design progresses the differences and contradictions between the desired and available properties emerge.

Architecture and design is often a process of trading off between different requirements and the ability of different potential solutions to best meet those requirements.



Modelling

Modelling captures and communicates the essential features and functions of the architecture.

Models are more than just diagrams; they're tools for understanding, validation, and communication.

Models are abstractions that emphasize specific aspects .

Models may be:

- **Contextual**: provide an overview of a situation's motivation and scope.
- **Conceptual**: describe the main elements, their interactions. And their benefits.
- **Logical**: describe the detail of all elements, their interactions and outcomes.
- **Physical**: define how all elements and interactions will be specially implemented and their effect.

QA

Design Patterns

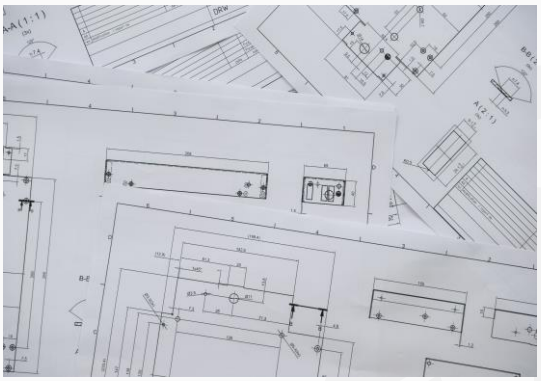
- Patterns describe how things are put together for similar things to achieve their targeted outcomes.
- Patterns aren't one-size-fits-all.
- They can be shapes that can often be applied and adapted to different specific needs.
- They can also be shapes that are imposed on different solutions to provide a degree of consistency and effectiveness between those solutions.

QA

Designing Target Solution Architecture

Design Patterns:

1. Data design
2. Enterprise integration
3. Business model
4. Application integration
5. Technology solution



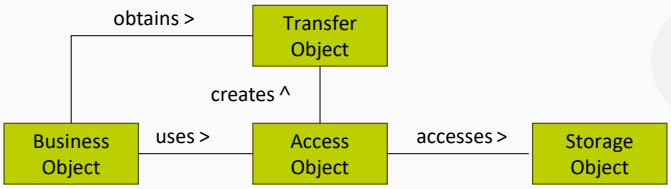
Consider

How might different projects/solutions influence your use of design patterns?
How might you choose between different design patterns and why.

You should be able to discuss the activities undertaken by the architect during design, including how to analyse the possible options and suitability of design patterns.

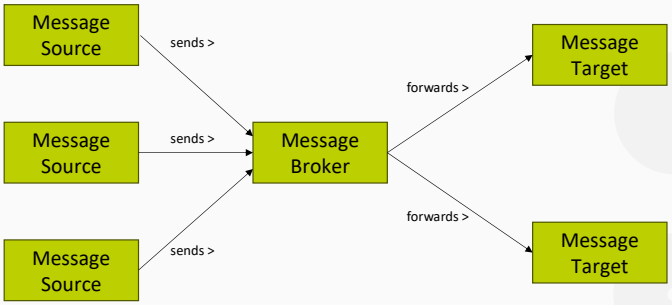
Data Design Patterns

Data Design Patterns focus on effective and efficient data storage, retrieval, and manipulation.
Example: Data Access Object Pattern



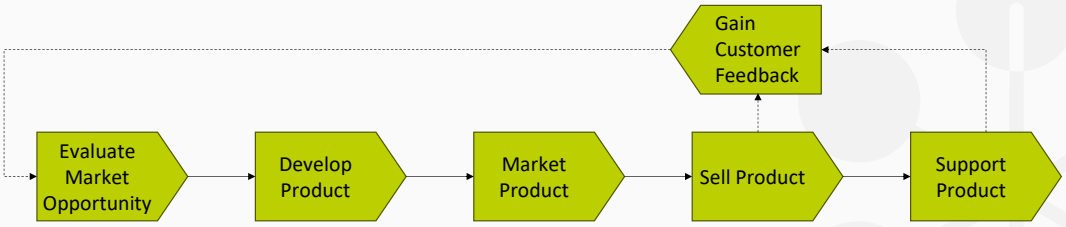
Enterprise Integration Patterns

Enterprise Integration Patterns provide at scale exchange of messages between systems.
Example: Message Broker Pattern



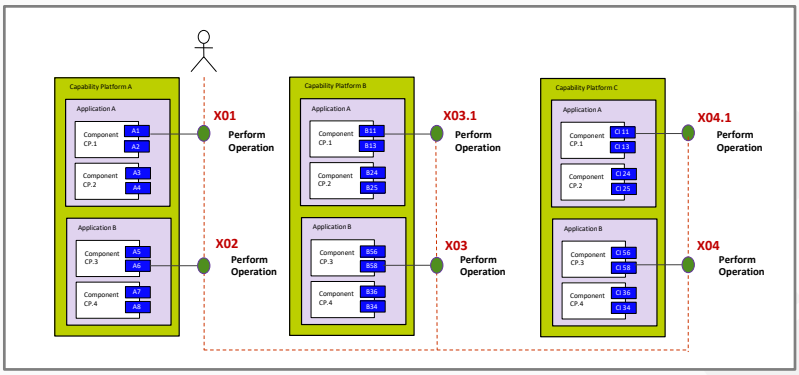
Business Model Patterns

Business model patterns show how business objects interact at various levels of detail and composition.
Example: Value Stream Pattern



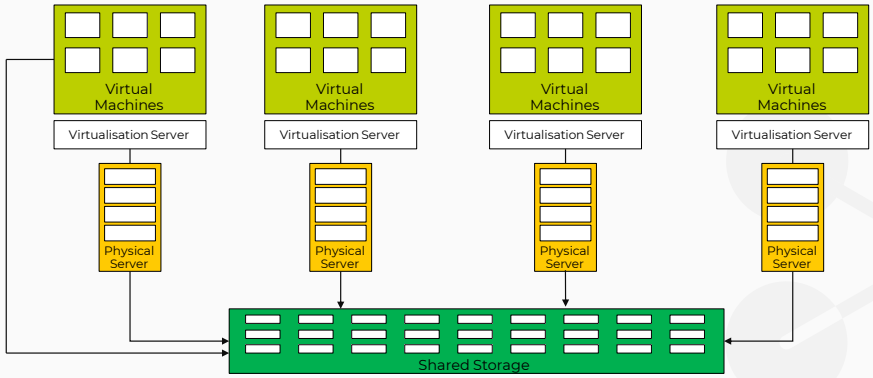
Application Integration Patterns

Application Integration Patterns show how application components interact to deliver business services.
Example: Generic Business Service Pattern



Technology Solution Patterns

Technology Solution Patterns show how infrastructure interacts to provide an effective execution and managed environment. Example: Virtualisation Pattern



Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. **The use of common architecture models.**
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. Gap Analysis.
8. A business case for presentation.

Architecture Models

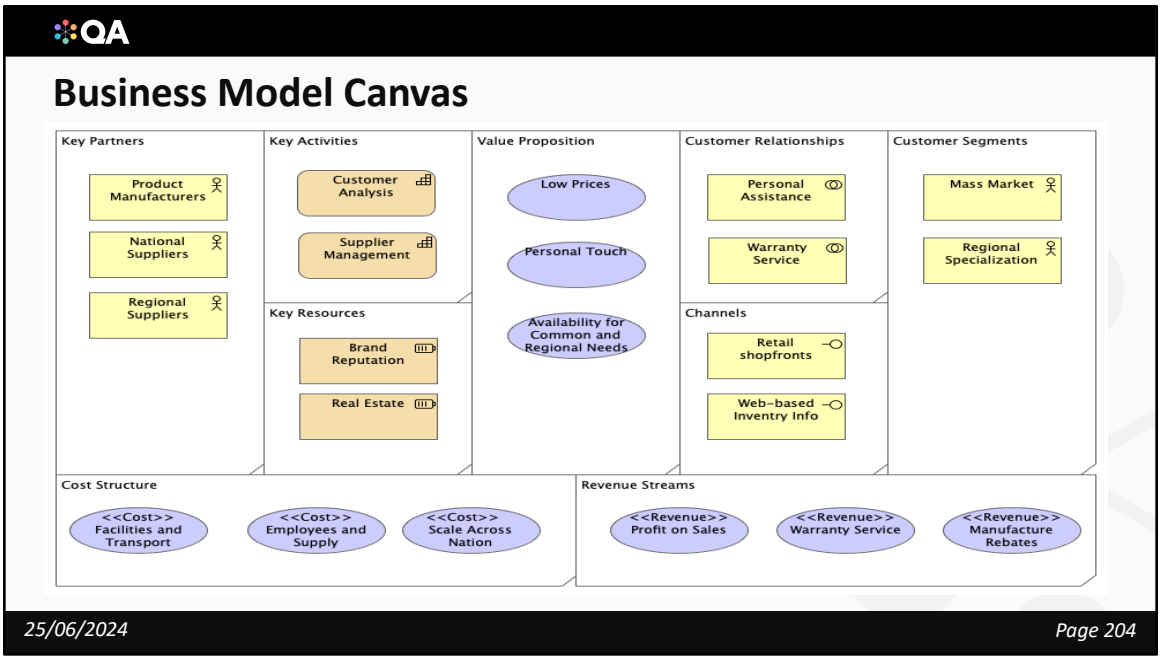
- Business motivation
- Capability map
- Value stream map
- Functional Decomposition
- Business process

25/06/2024 Page 203

Consider

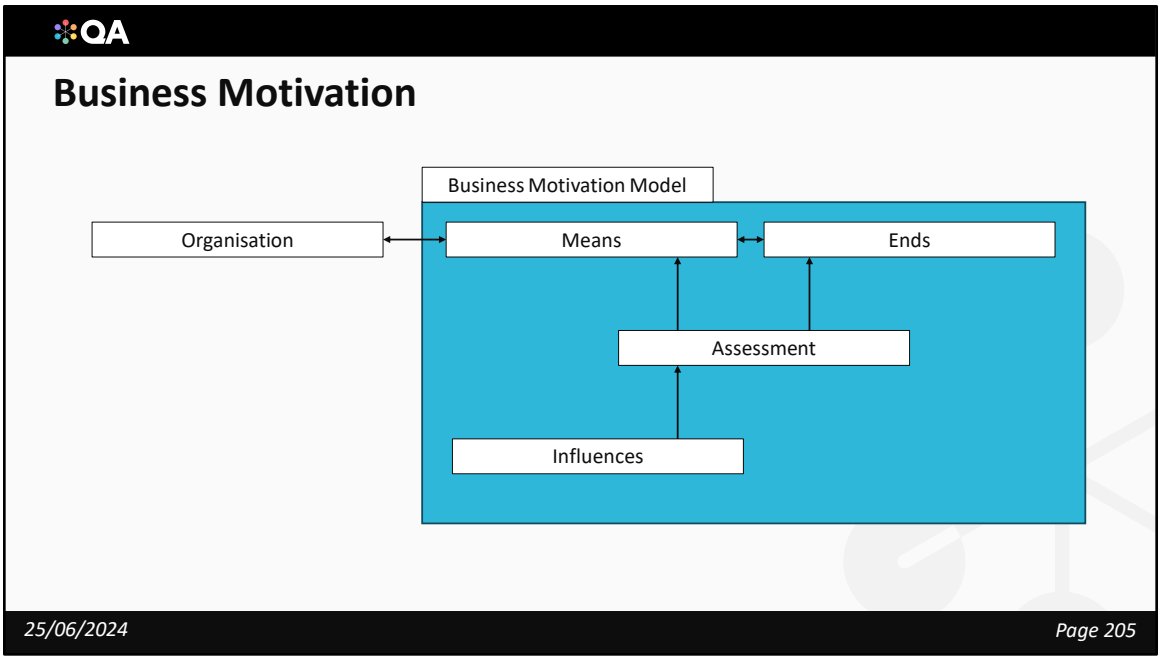
What is the purpose of each model?
What should be included?
A list of the purpose, scope, and information required for each model.

You should be able to discuss the purpose and application and desired output of the common architecture models listed, and their suitability for use in a given scenario.



A BMC is a visual template that illustrates various objects of a business model for an enterprise. Osterwalder's original canvas includes nine elements, that represent vital aspects of business survival.

https://en.wikipedia.org/wiki/Business_Model_Canvas



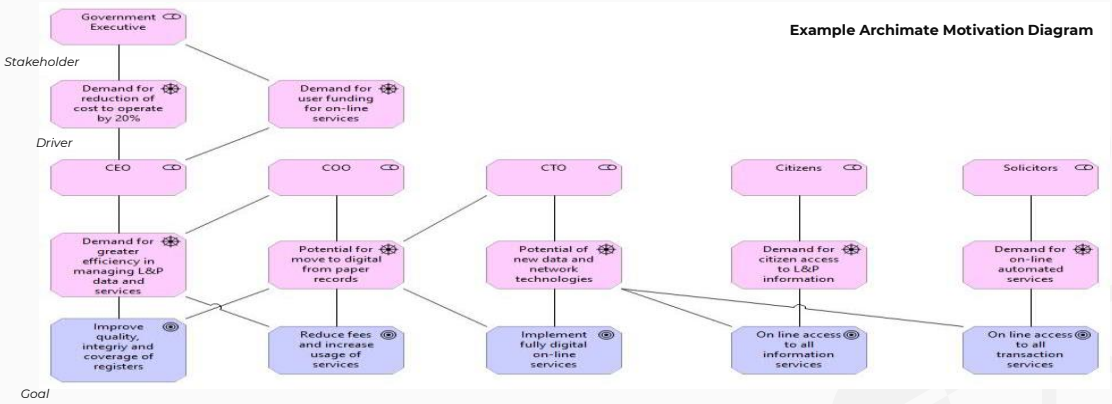
There are a number of business motivation models that capture the initial motivation of an enterprise and thread that through to the architecture landscape and its elements, including Archimate and The Business Motivation Model.

https://www.opengroup.org/archimate-forum/archimate-overview

https://en.wikipedia.org/wiki/Business_Motivation_Model

QA

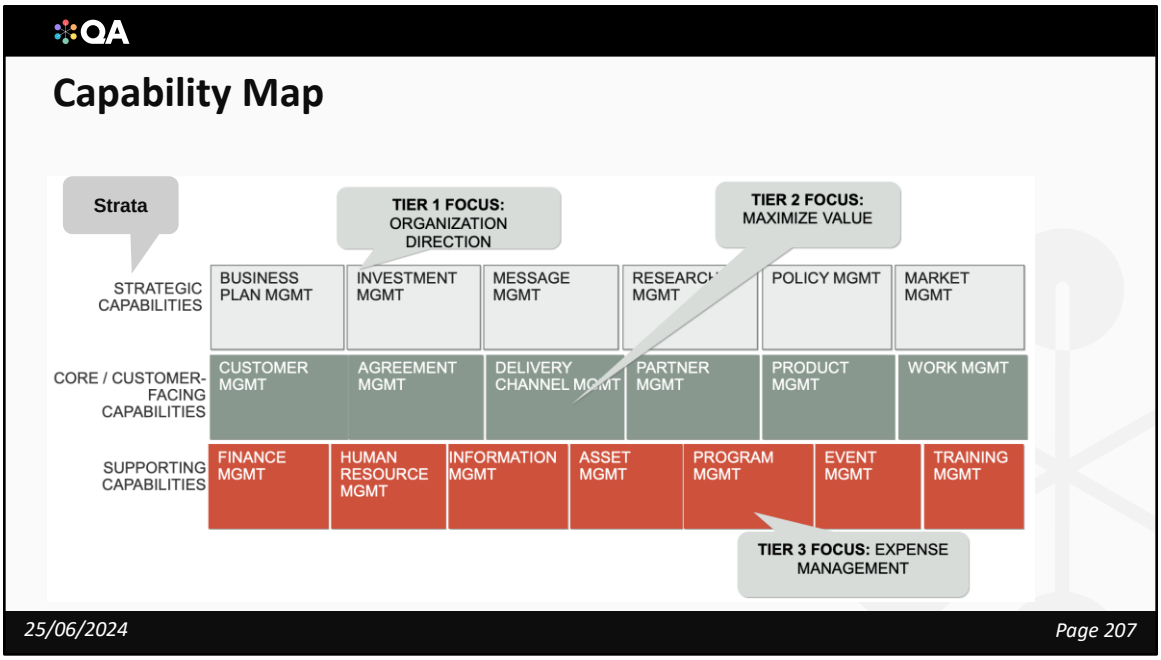
Business Motivation - Archimate



There are a number of business motivation models that capture the initial motivation of an enterprise and thread that through t0 the architecture landscape and its elements, including Archimate and The Business Motivation Model.

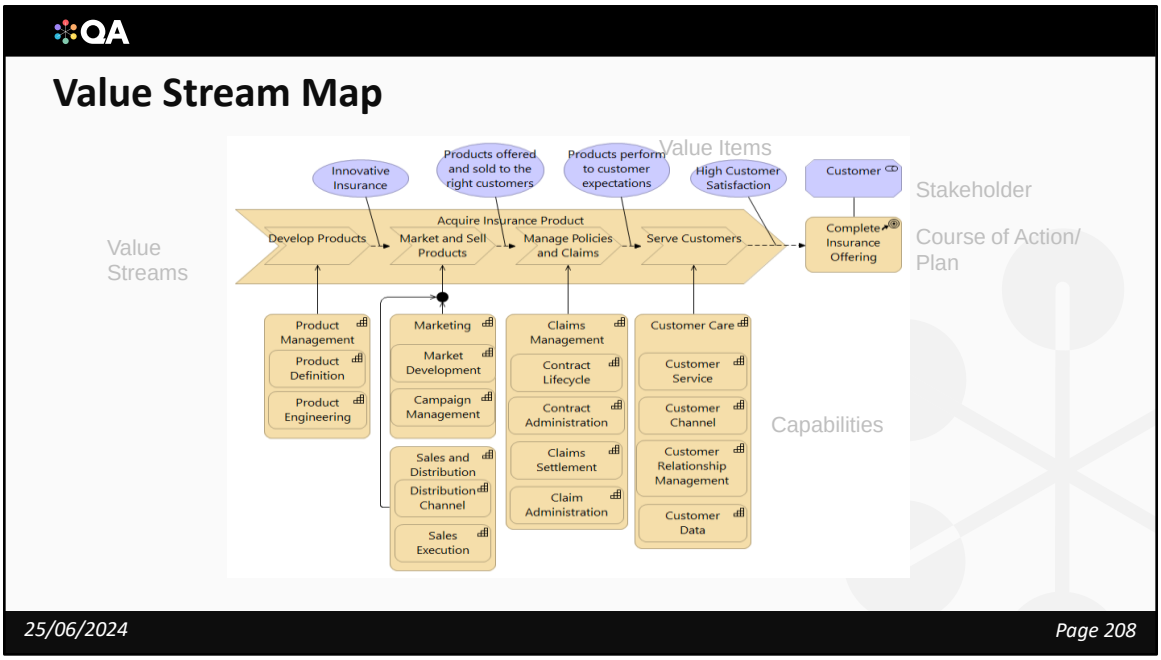
https://www.opengroup.org/archimate-forum/archimate-overview

https://en.wikipedia.org/wiki/Business_Motivation_Model



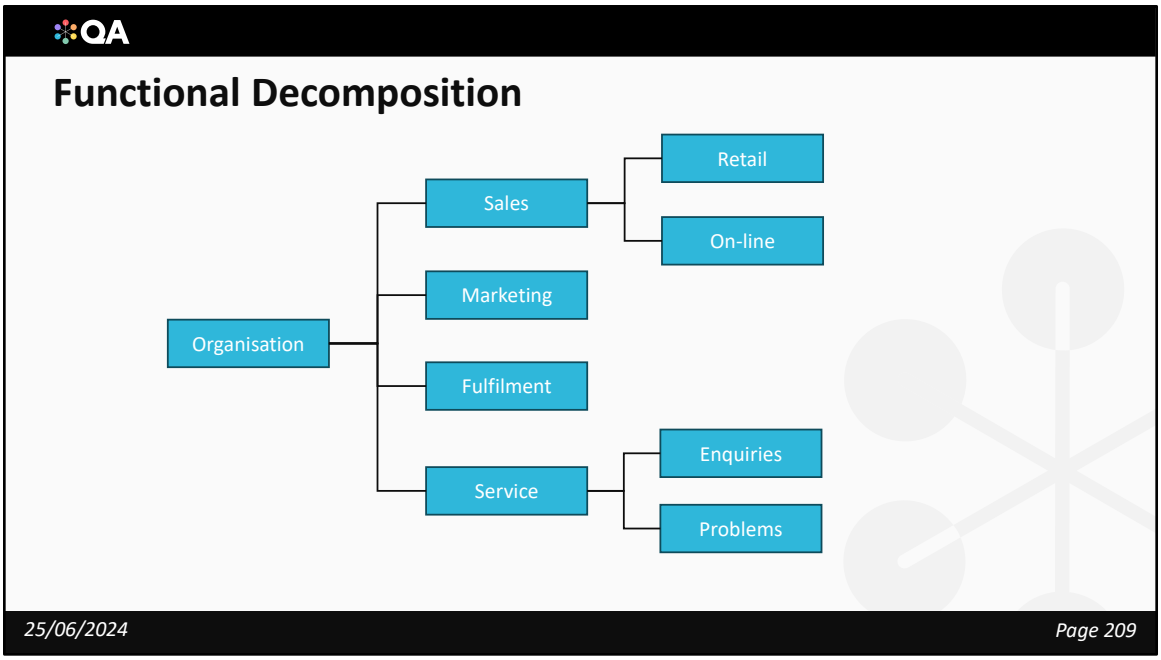
Capability map:

A visual representation of the abilities an organisation needs to achieve its objectives
Capabilities are the building blocks of an organisation, highlighting what it can do irrespective of how it's done



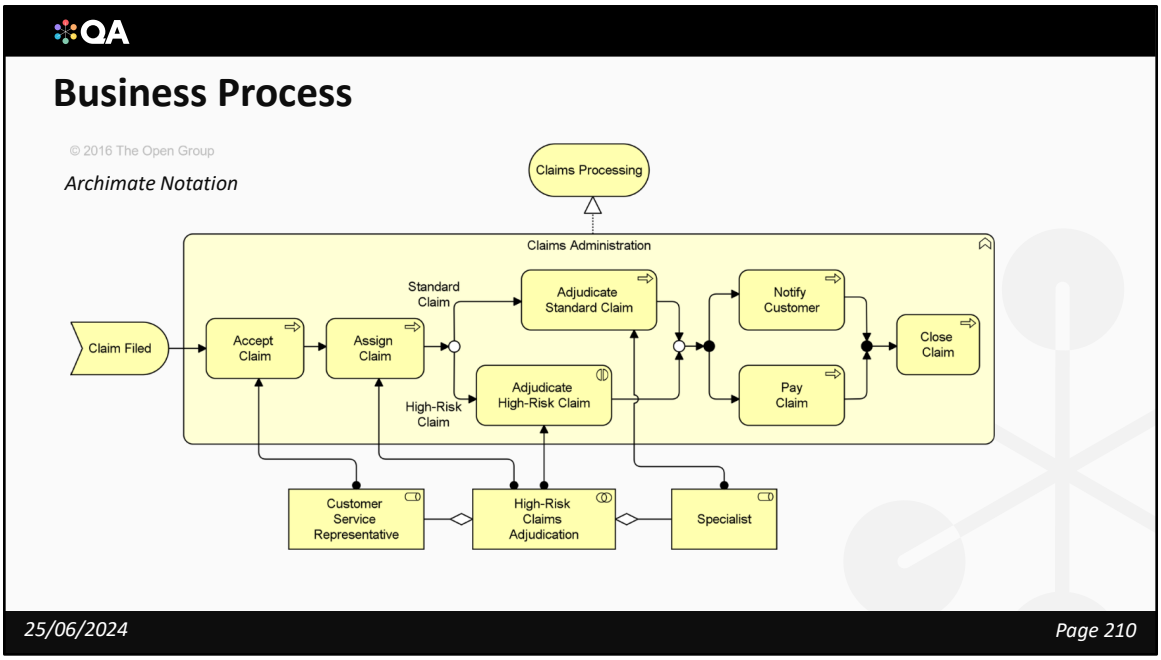
Value stream map:

A tool that visualises a processes in terms of value creating stages necessary to bring overall value to a customer
It's about optimising the flow and ensuring value delivery at every step.



Functional decomposition:

Breaks down complex functions or processes into simpler, more manageable parts
Simplifies complexity and makes it easier to understand, manage, and optimise



Business process models

Graphical representation of business processes, showcasing activities, flow, and interactions
It's about understanding the sequence, dependencies, and stakeholders involved in a process

Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. The use of common architecture models.
5. **Common information systems modelling techniques.**
6. Different types of business requirements.
7. Gap Analysis.
8. A business case for presentation.

Information System Modelling Techniques

- Process model
- Data model
- Context diagram
- Use case diagram
- Data flow diagram
- Interaction/sequence diagram

Consider:

What is the purpose of each item?
What should be included?
What is the purpose, scope, and information required for each model.
How might the project/solution type influence the use of the listed models?

You should be able to discuss the purpose and use of modelling techniques as listed, and consider the applicability of each technique to a given scenario.

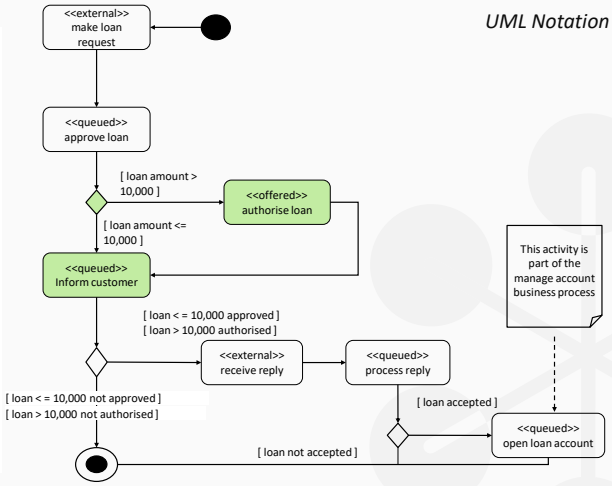
Process Model

Process flow / activity diagrams are used to display a sequence of activities from a start point to a finish point.

They can used to model different aspects of sequences including business processes and software execution paths.

The example shown is of a business process model for a loan request.

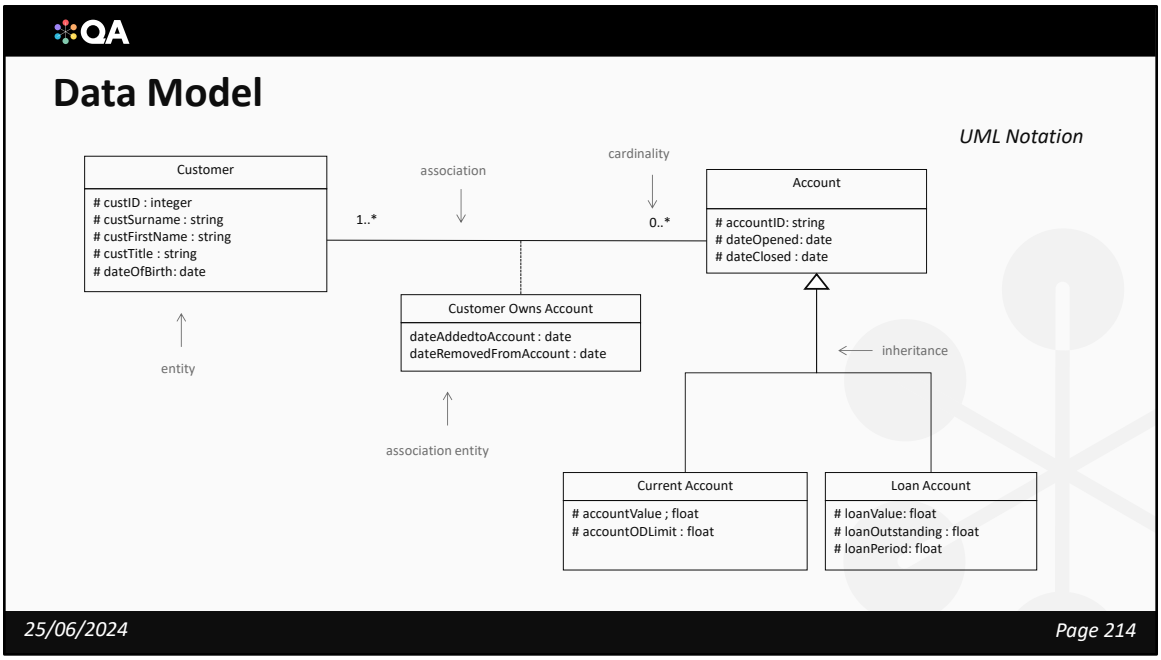
They can be partitioned and nested and they can include swimlanes grouping activities within role or organisation boundaries.



Process models:

Represents a sequence of activities in a process, focusing on the flow and interactions
Process models provide clarity on 'how' things are done, step by step

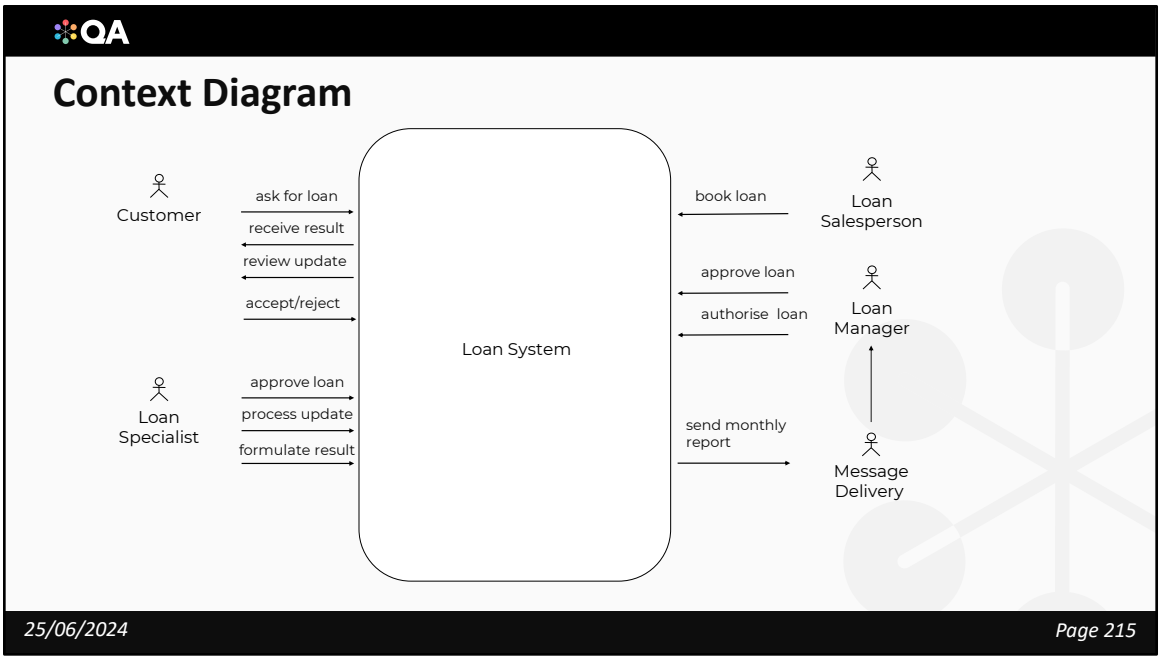
https://www.omg.org/spec/UML/#docs-normative-supporting



Data models:

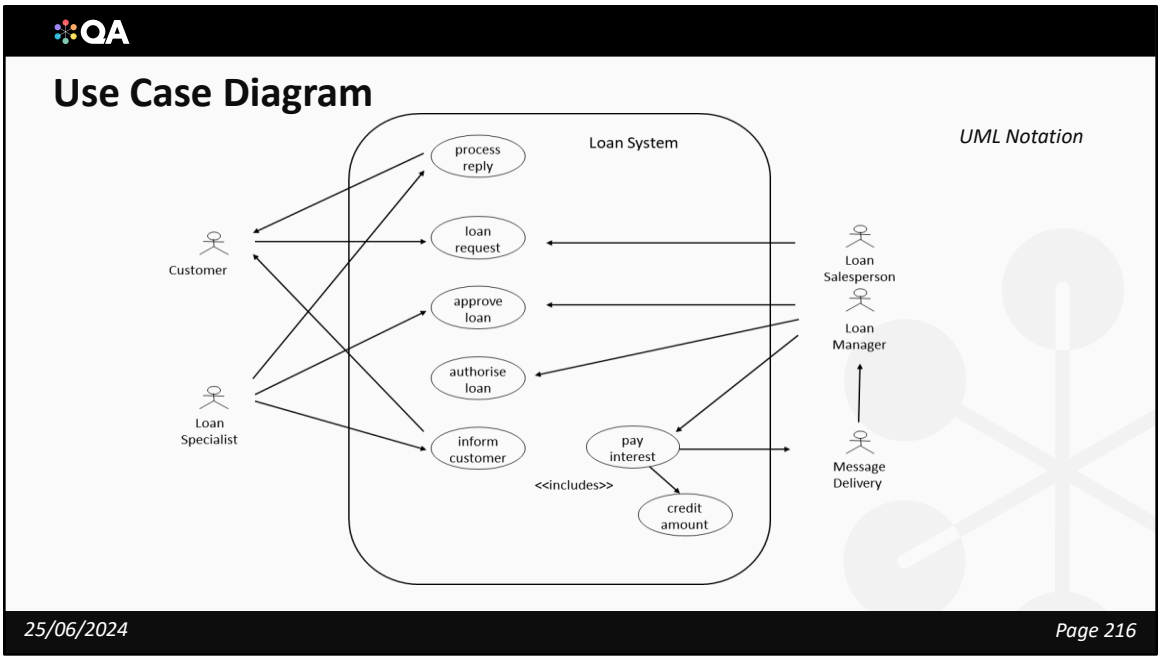
Outline the structure, relationships, and constraints of data within a system.
Notes: "Data is the backbone of systems. Understanding its structure and relations is crucial.

https://www.omg.org/spec/UML/#docs-normative-supporting



Context diagrams:

A high-level view of a system, showcasing its boundaries, external entities, and interactions
Sets the stage, defining what's inside and outside the system's scope

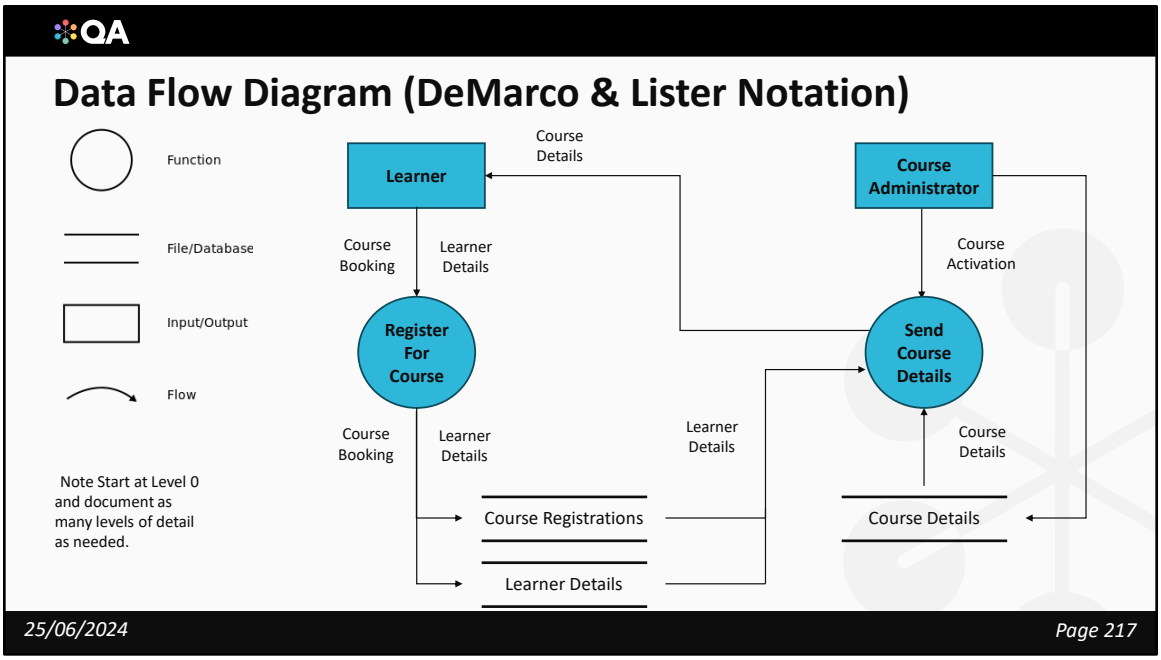


Use case diagrams:

Visualises the interactions between a system and its external entities (actors
About understanding events that have an information exchange across the system boundary.
What actions are expected from the system

A use case is a piece of functionality, a sequence of actions, in the system that gives the users a result of value.
They drive the requirements model that is transformed into the design model and realised into the solution implementation.

https://www.omg.org/spec/UML/#docs-normative-supporting



Data flow diagram:

Showcases the flow of data within a system, highlighting sources, destinations, and transformations. Data doesn't stay static; understanding its movement and transformation is key.

Note: You could also consider mapping the data lifecycle and different states of data throughout it.

QA

Interaction / Sequence Diagram

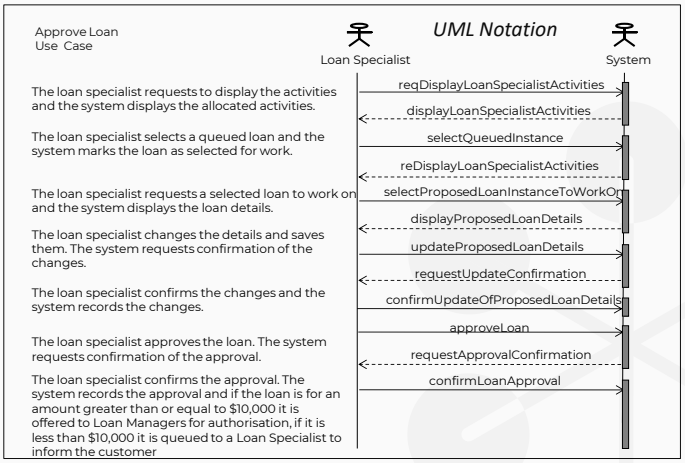
A scenario diagram is a specific type of sequence diagram that describes the system when a use case is run.

Each specific event is identified and the exchange that occurs during the event.

Multiple scenarios may be modelled for the "happy path", "alternate paths" and "failure paths".

The one shown is the "happy path" for the approve loan use case.

The scenario can be described by using the defining sentences and / or the specific operation names depending on the level of detail required, the audience, and the need to connect the logical operation to their specific implementations.



Interaction/sequence diagrams:

Detail the interactions between objects in a system over time.
A timeline of interactions, providing a dynamic view of the system's behaviour.

https://www.omg.org/spec/UML/#docs-normative-supporting

Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. **Different types of business requirements.**
7. Gap Analysis.
8. A business case for presentation.

Business requirements are clear and specific statements that outline the needs and expectations of a business, acting as a guide for designing and developing technology solutions that fulfil those needs.

Business Requirements

- Types of requirements
- Quality and characteristics of requirements

Consider

How to create a list of requirements and categorise them
How to capture non-functional, quality and performance attributes.

You should be able to discuss the different types of requirements, their origins and how to ensure quality.

Types Of Requirements

Types of Requirements:

- Business and IT Strategy: High-level goals and objectives and specific business activities, entities and outcomes
- General Directives: Constraints derived from the organisation's architectural framework.
- General Requirements: Broad business needs across all domains
- Specific Domain Requirements: (Business, Application, Data, Technical and Security)

- **Covering:**

  Functional Aspects: Describe what the system should do.
- Non-functional Aspects: Describe how the system should behave (e.g., performance, security).
- Specific Constraints: Limitations or restrictions.

Quality & Characteristics Of Requirements

Quality requirements reduce ambiguities and ensure a smoother development process.

- Clear and concise
- Complete and comprehensive
- Consistent with other requirements
- Relevant to the project
- Verifiable and achievable

Business Requirements - INVEST

- Independent
- Negotiable
- Valuable
- Estimatable
- Small
- Testable

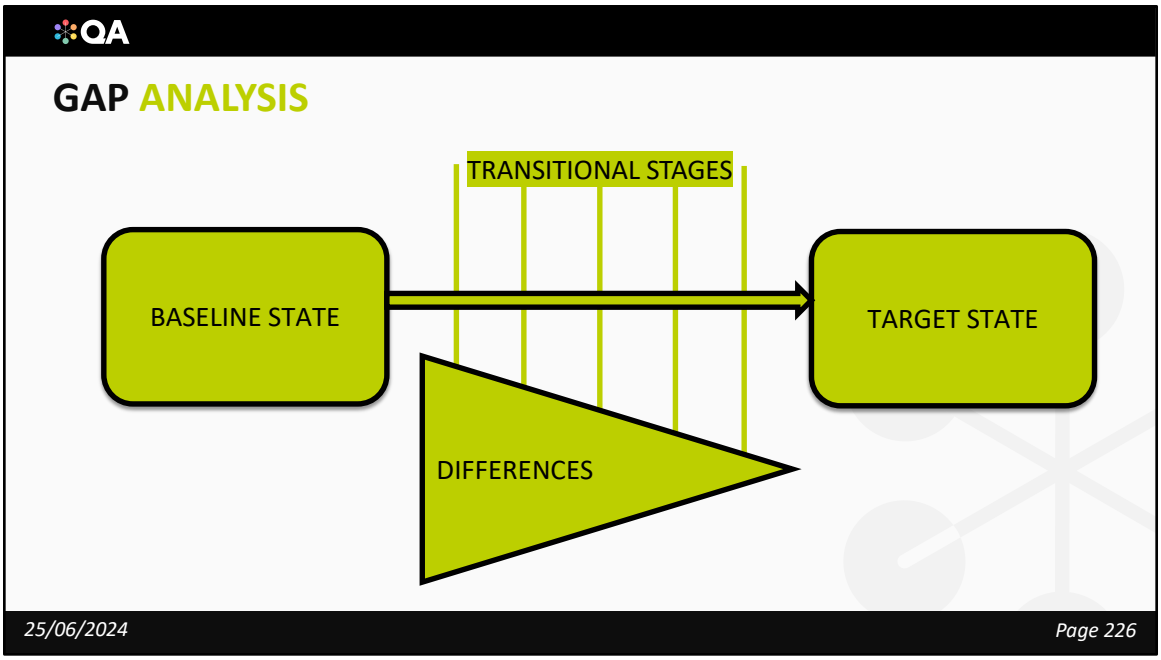
The INVEST criteria ensure that requirements are well-defined & actionable.

Independent: Each requirement should stand alone
Negotiable: Open for discussion and not set in stone
Valuable: Brings value to the business or users
Estimateable: Can be estimated in terms of time, cost, & resources
Small: Not too broad; should be specific
Testable: Can be verified through testing

You should be able to discuss the different types of requirements, their origins and how to ensure quality.

Architecture Process Tools and Vision

1. The practical application of the solution architecture framework.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. **Gap Analysis.**
8. A business case for presentation.



Gap analysis is one of the tools which architects can use to help them identify what is required of a solution. This works by comparing the existing situation with the desired one to find gaps, this comparison can be done by comparing two architecture descriptions. As the solution is implemented and deployed, these differences and gaps should close. This may occur over time depending on the deployment style or in instances of major change being broken down into multiple smaller changes.

Consider:

Instances in which gap analysis would be useful.
The need for transitional stages.

Benefits of a gap analysis

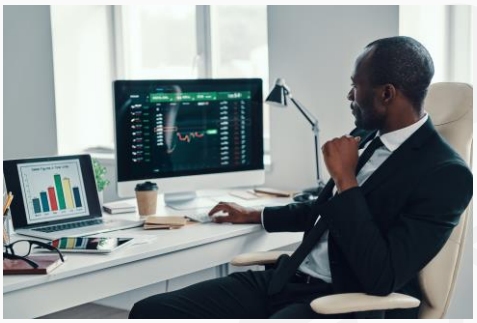
Identifies areas of improvement
Understand current capabilities
Set clear objectives for future growth
Prioritise actions based on the identified gaps

By understanding the gaps, organisations can make informed decisions and allocate resources effectively.

You should be able to use a gap analysis to identify differences, required improvements and planned growth between a current and desired state, or between two or more architecture descriptions.

Preparing A Gap analysis

- Grid method
- Table method
- Graph method
- Gap report
- Gap models

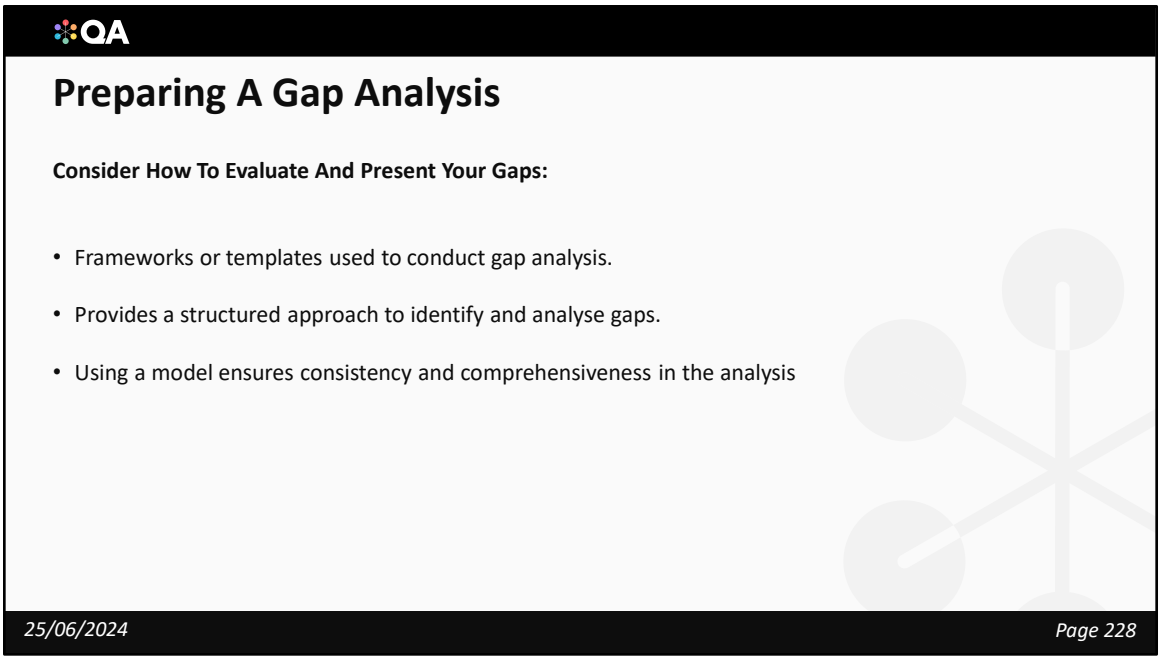


Consider:

What is the purpose of gap analysis?
What the advantages and drawbacks of the items listed?
How does the type of project/solution influence the type of method used?
How does the target audience influence the type of method used and the information included?

You should be able to discuss how to prepare a gap analysis, through the use of various tools and techniques.

Grid method
Table method
Graph method
Gap report



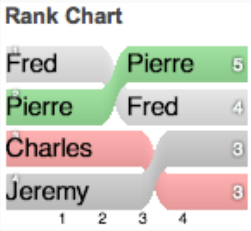
More extensive, including multiple levels that are required to perform the analysis in sufficient detail. Gap models can be highly valuable stakeholder communication tools as they are able to focus on a specific area of interest and benefit from their resemblance to existing familiar models that were designed to present information clearly and concisely. They are essentially a combination of the ‘before’ and ‘after’ representations designed to show the transition between the two.

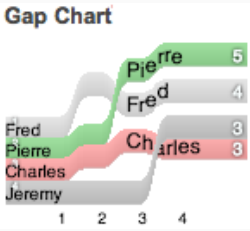
Gap Analysis & Grid Presentation

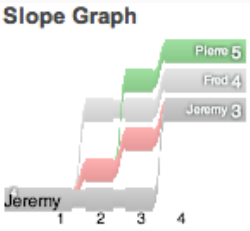
- The technique known as **gap analysis** is widely used in the ADM to validate an architecture that is being developed.
- It highlights any shortfall between the Baseline and Target architecture, i.e., items deliberately or accidentally omitted, or not yet defined.
- A key validation step is to consider what may have been forgotten or overlooked (Eliminated services).
- The most critical source of gaps are those relating to stakeholder concerns that have not been addressed in prior architectural work.

Target Architecture → / Baseline Architecture ↓	Video Conferencing Services	Enhanced Telephony Services	Mailing List Services	Eliminated Services ↓
Broadcast Services				Intentionally eliminated
Video Conferencing Services	Included			
Enhanced Telephony Services		Potential match		
Shared Screen Services				Unintentionally excluded - a gap in Target Architecture
New →		Gap: Enhanced services to be developed or produced	Gap: To be developed or produced	

Gap Analysis Graph & Table Presentation





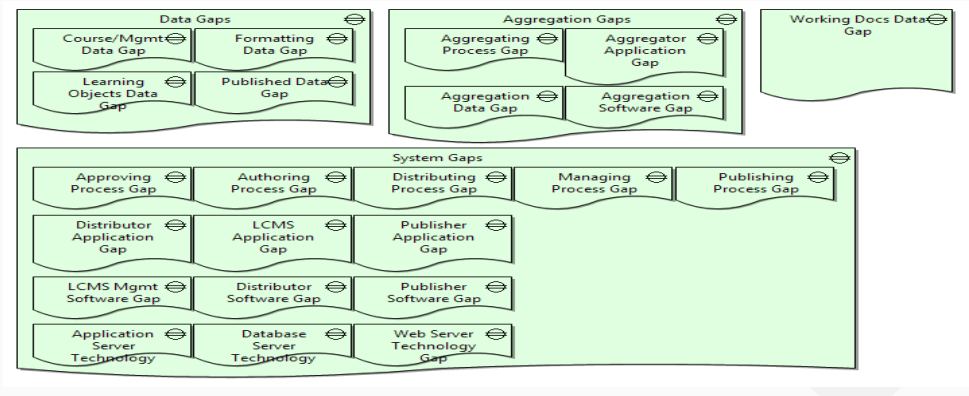


Raw Table

\	D1	D2	D3	D4
Fred	0	3	3	4
Pierre	0	1	4	5
Charles	0	1	2	3
Jeremy	0	0	0	3

Gap Report

Gap report: the result of a gap analysis that compares the current and future state models of a solution and itemises the changes required to move to the future state. Usually includes a breakdown of the costs, resources and time that will need to be invested to achieve the change.



A gap report contains a larger, containing more components and interfaces with more detailed information about the changes required, and exposing a lower level of detail, where relevant, such as subcomponents and configuration options.

Architecture Process Tools and Vision

1. **The practical application of the solution architecture framework**.
2. Varieties of the Software Development Lifecycle (SDLC).
3. The design of a target solution architecture.
4. The use of common architecture models.
5. Common information systems modelling techniques.
6. Different types of business requirements.
7. Gap Analysis.
8. **A business case for presentation.**

QA

Presenting A Business Case

- Contents of a business case
- Cost-benefit analysis
- ROI
- Risk assessment
- Gap analysis (see slide 50)
- Non-technical stakeholders

25/06/2024 Page 233

Consider:

What are the main elements included in a business case?
Which elements are compulsory, and which are optional?
What is the most effective method of business case delivery?
How does the target audience influence the information included in a business case?

You should be able to discuss the role of the architect in preparing the contents of a business case, including the artefacts and information required and the most suitable methods for presentation.

Business Case Contents

- Executive Summary
- Business Objective
- Current Situation & Problem/Opportunity Statement
- Solution Options & Recommendations
- Cost-Benefit Analysis
- Risk Assessment
- Gap Analysis
- Implementation Timeline
- ROI (Return on Investment) Projection
- Each section of the business case serves a specific purpose and is crucial for gaining approval and funding for the project.

A business case provides the justification for a proposed business change or plan, and typically outlines the allocation of capital and resources required to implement the proposed business case.

A business case is a critical document that outlines the rationale for initiating a project or task. It is the way by which an organisation presents the benefits of an undertaking and compares it with the potential risks and costs associated with it.

Cost Benefit Analysis and ROI

Cost-Benefit Analysis (CBA):

A process used to measure the benefits of a decision or taking action minus the costs associated with taking that action. A CBA helps in determining whether a project is financially viable and what the potential benefits will be

Return on Investment (ROI):

A performance measure used to evaluate the efficiency of an investment or compare the efficiency of several different investments. An ROI provides a direct measure of the return on a particular investment, relative to the investment's cost.

QA

Risk Assessment and Risk Model (after TOGAF®)

- Risk classification
- Risk identification
- Initial risk assessment
- Risk mitigation and residual risk assessment
- Risk monitoring

Effect	Frequency				
	Frequent	Likely	Occasional	Seldom	Unlikely
Catastrophic	E	E	H	H	M
Critical	E	H	H	M	L
Marginal	H	M	M	L	L
Negligible	M	L	L	L	L

Risks are defined here as anything that can affect delivery of critical stakeholder values, including quality requirements, or which can negatively affect resource consumption in the short term or long term. Risk management needs to permeate all aspects of IT projects and operations.
Risks are everybody's responsibility, at a very detailed level of planning requirements, contracts, architecture and implementation.

Determining the risks that are relevant to an asset or object.
Qualitative risk assessment classes risk on a high-medium-low scale.
Quantitative risk assessment classes risk on a numeric scale.
The main deliverable of a risk assessment is the Business Risk Model.

Non-Technical Stakeholders

Presenting information to non-technical stakeholders:

When presenting to non-technical stakeholders, it's crucial to make the information accessible and relatable. The goal is to convey the value and importance of the architecture without overwhelming them with technical details

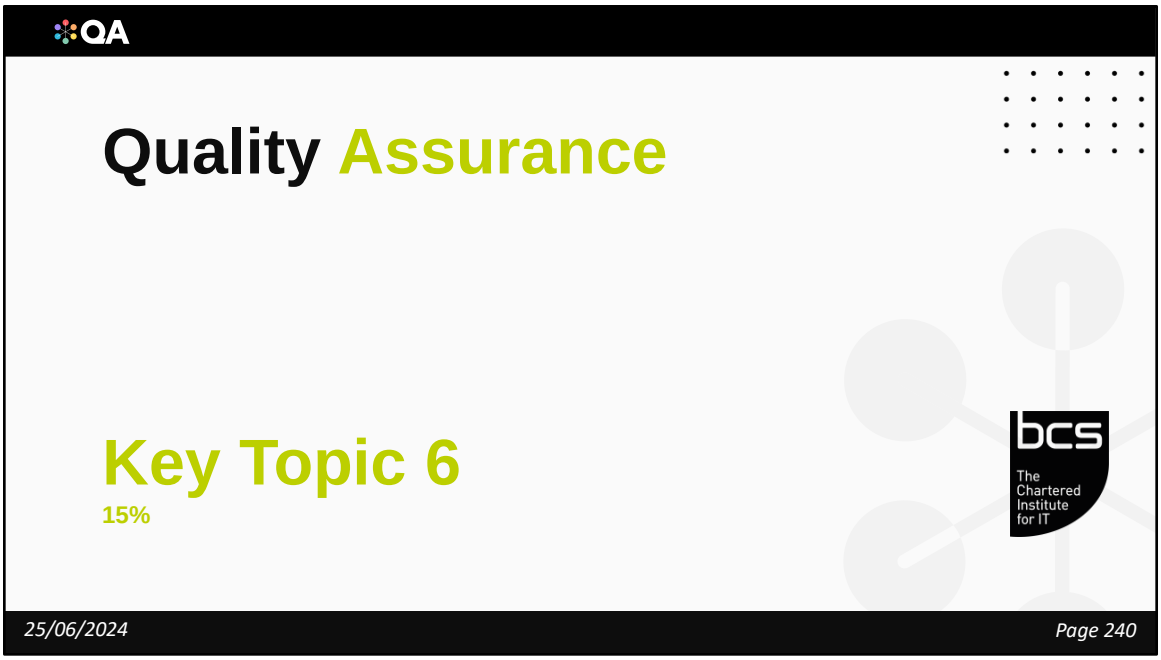
Use Simple Language: Avoid jargon and technical terms:

- Visual Aids: Graphs, charts, and infographics.
- Relate to Business Goals: Connect technical aspects to business objectives.
- Practice & Prepare: Anticipate questions and practice your delivery.

Please open your workbooks to topic 5

Section 06

Quality Assurance



Elements to be addressed:

Quality Assurance

The use of a range of quality assurance tools or techniques.
The principle of Separation of Concerns (SoC).
Change management tools and processes.
Opportunities for continuous improvement.

The Course Schedule

DAY 3 09:00 - 15:30

6.1 Refresh On Quality Assurance

6.2 Practice Exercises

7.1 General Practice Exercise

7.2 Questions and Wrap Up

Quality assurance

1. **Use a range of quality assurance tools or techniques.**
2. Apply the principle of Separation of Concerns (SoC).
3. Use change management tools and processes.
4. Continuous improvement.

Quality assurance involves systematically assessing and validating the design and implementation systems to ensure they align with best practices, meet functional and non-functional requirements, and contribute to the organisation's objectives.

QA Tool And Techniques

- Delivery roadmap
- Comparison to Service Level Agreement (SLA)
- Comparison against requirements
- Technical walkthroughs

25/06/2024 Page 244

Consider:

Which stakeholders are usually contributors to the items listed?
How would the scope/budget/aims of a solution influence the use of the tools and techniques listed?

You should be able to discuss the use of the tools listed as means of assuring the quality of a solution. You should be able to discuss how requirements, SLA and delivery roadmap can be used to compare the solution being delivered to the needs and expectations of stakeholders, and technical walkthroughs can identify any quality issues.

QA Tools & Techniques

Quality Assurance (QA):

- A systematic process that ensures product and service excellence.
- Preventing defects in products and ensuring that requirements, standards, and criteria are met consistently.

Delivery Roadmap as a QA Tool:

- A delivery roadmap ensures that everyone is on the same page about what's being delivered and when. It acts as a reference point for QA to ensure timely and quality delivery.
- It is a visual representation that outlines the path or timeline of product delivery.
- Benefits: Clear milestones, Predictable delivery, Stakeholder alignment.

Comparison To Service Level Agreements

A SLA defines the level of service expected by a customer from its provider / supplier.

- Quality assurance tests conformance to specification and satisfaction levels for all relevant solution elements including service levels.
- Comparing the actual service delivery against the SLA ensures that the service meets the quality and performance metrics agreed upon with the client.

Comparison Against Requirements

Requirements describe all the defined expected change and performance levels for a solution.

- Comparing the solution against initial requirements ensures that the product meets the needs and expectations of stakeholders.
- This is often checked with a Requirement Traceability Matrix (RTM).

Requirements Test Count: Tests by Task

Task	Total # QA Tests	Functional Requirement ID: Camp_01	Camp_04	ClinOps_01	ClinOps_02	ClinOps_03	Event_01	Event_03	Event_04	Event_05	Form_02	Form_03	Form_04	Form_07	Loc_01	Loc_02	Loc_03	Medication_02	NYSIIS_04	PreReg_05	Provider_02
Add Patient Records	104			1	79	24															
Associate Medications	16								16												
Associate Providers	13									13											
Barcode Generation	14																	7			7
Cancel Reservation	27																			27	
Correct Invalid NDC Codes	6																	6			
Create a New Form	33										31	2									
Create Event	33						31	2													
Create Location	15														12	3					
Create New Campaign	26	25	1																		
Edit Archived Form	1												1								
Edit Form	18										5	13									
Edit Locations	4																4				
Edit Published Form	2											1	1								
Edit Unarchived Form	5												5								
Edit Unpublished Form	3												2	1							
Edit Copy Existing Form	9										2	7									
Export Patient Records	1				1																
Generate Paper Form	2				1						1										
Total Tests per Functionality:	**332**	**25**	**1**	**1**	**81**	**24**	**31**	**2**	**16**	**13**	**39**	**23**	**9**	**1**	**12**	**3**	**4**	**7**	**6**	**27**	**7**

Technical Walkthroughs

A review process where specialist and developers present their work to peers and stakeholders.

- They should be collaborative and help in identifying quality issues early, ensuring that the final product is of high quality.
- Benefits: Early defect identification, effective knowledge sharing and promoting continuous improvement.

Quality assurance

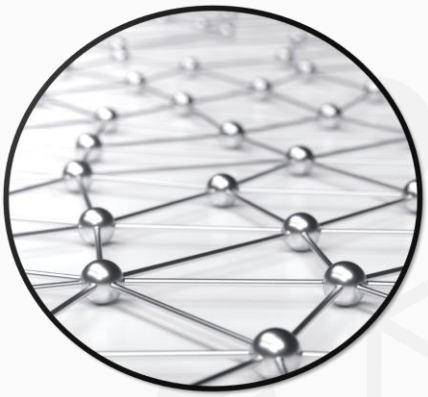
1. Use a range of quality assurance tools or techniques.
2. **Apply the principle of Separation of Concerns (SoC).**
3. Use change management tools and processes.
4. Continuous improvement.

QA

Separation Of Concerns (SoC)

Separation by:

- Resource access layer
- Business layer
- Presentation layer
- Vertical separation into client-server layers



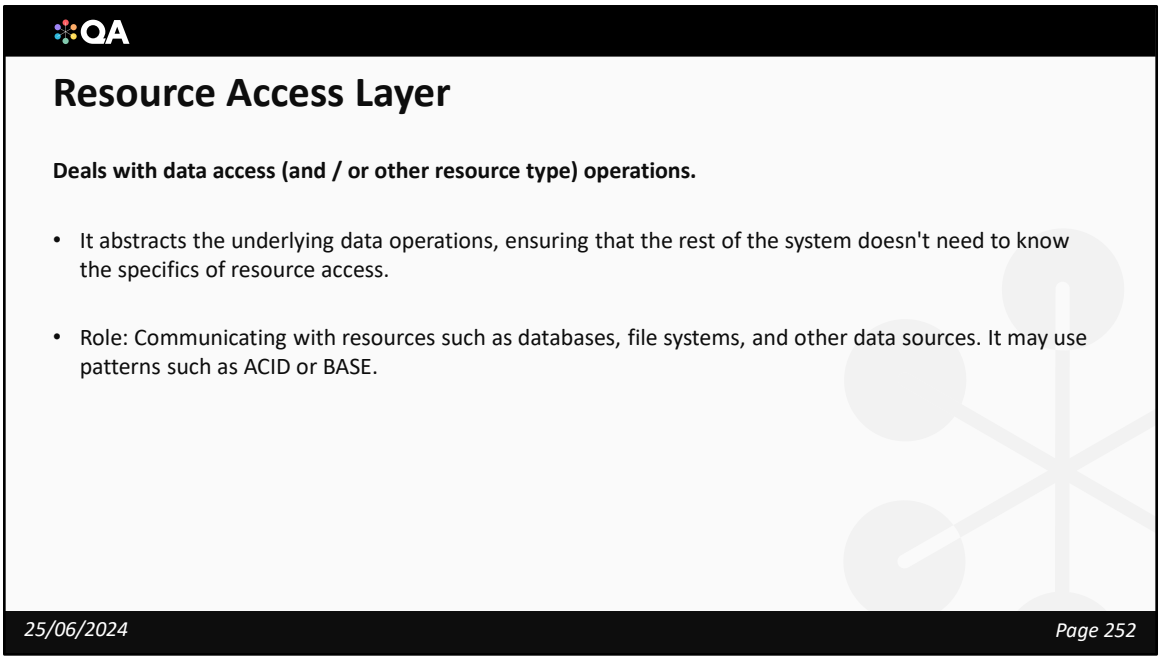
Consider:

How separation of concerns can be achieved through layers (E.g. presentation, business, data, etc.)
What are the benefits of isolating different aspects of an application?

You should be able to discuss the use of separation of concerns - either by layer or by feature – and explain how it uses the software development principles of “single responsibility” and “cohesive responsibilities” to ensure the intended solution is both logical and easy to maintain.

Separation of Concerns (SoC)

- SoC is a design principle for separating a elements program into distinct cohesive sections, ensuring that each section addresses a separate concern and does so well.
- SoC promotes modularity, makes the software more maintainable, scalable and understandable; and promotes appropriate loose coupling and high cohesion.



ACID guarantees a safe environment in which to process your data. The ACID acronym stands for:

Atomic: All operations in a transaction succeed, or every operation is rolled back.
Consistent: On the completion or rollback of a transaction, the database is in a consistent state; there are no partial updates or logical corruptions.
Isolated: Transactions do not contend with one another. Contentious access to data is moderated by the database so that transactions appear to run sequentially.
Durable: The results of applying a transaction are permanent, even in the presence of failures

BASE data store values availability (since that's important for scale), but it doesn't offer guaranteed consistency of replicated data at write time. Overall, the BASE consistency model provides a less strict assurance than ACID:
Here's how the BASE stands for:

Basic Availability: The database appears to work most of the time.
Soft-state: Stores don't have to be write-consistent, nor do different replicas have to be mutually consistent all the time.
Eventual consistency: Stores exhibit consistency at some later point (e.g., lazily at read time).

Navigating ACID vs. BASE Trade-Offs

There's no right answer to whether your application needs an ACID versus BASE consistency model. Developers and data architects should select their data consistency trade-offs on a case-by-case basis – not based just on what's trending or what model was used previously. Given BASE's loose consistency, developers need to be more knowledgeable and rigorous about consistent data if they choose a BASE store for their application. It's essential to be familiar with the BASE behaviour of your chosen aggregate store and work within those constraints. On the other hand, planning around BASE limitations can sometimes be a major disadvantage when compared to the simplicity of ACID transactions. A fully ACID database is the perfect fit for use cases where data reliability and consistency are essential (banking, anyone?). If you

decide on using an ACID-compliant database, you might want to choose a graph database. These types of databases are great for model

Business Layer

- The Business Layer ensures that data is processed according to business rules before it's presented or stored.
- It contains business logic, calculations, and rules.

Role: Acts as an intermediary between the presentation layer and the resource access layer.

Presentation Layer

- The Presentation Layer is what the end-users interact with. It should be kept separate to ensure changes in the UI don't affect business logic.
- It is the user interface of the application.

Role: Display data to the user and interpret user commands.

Vertical Separation Into Client Service Layers

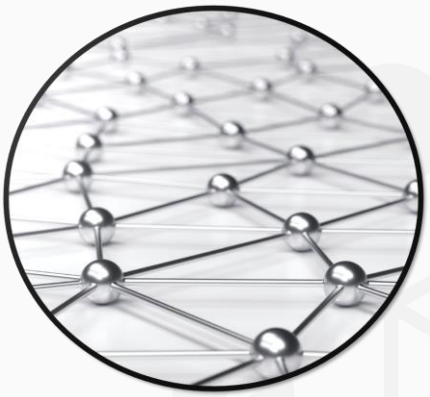
- Vertical separation allows for independent scaling and development of client and server parts.
- It divides the application into client (front-end) and server (back-end) parts.

Benefits: Scalability, Security, Maintainability.

QA

Separation Of Concerns (SoC)

- Horizontal separation of loosely-couple features



Consider:

What are the benefits of separating through features?
How does separation through features/modules differ from separation by layers?

You should be able to discuss the use of separation of concerns - either by layer or by feature – and explain how it uses the software development principles of “single responsibility” and “cohesive responsibilities” to ensure the intended solution is both logical and easy to maintain.

Horizontal Separation

Loosely-Coupled Features:

- By separating based on features, each module can evolve independently, reducing the risk of changes in one module affecting others.
- It divides the application based on features or modules.

Benefits: Modularity, Easier debugging, Independent deployments

Quality assurance

1. Use a range of quality assurance tools or techniques.
2. Apply the principle of Separation of Concerns (SoC).
3. **Use change management tools and processes.**
4. Continuous improvement.

QA

Change Management Tools And Processes

- Methods of control
- Requests for change
- Analysing change
- Risks

25/06/2024 Page 259

Consider:

Who is responsible for change management?
What artefacts can be used to reduce non-compliance?
How can artefacts be used during compliance procedures?
What are the potential risks of non-compliance?

You should be able to discuss the purpose and practical application of configuration and version, change control and management. You should be able to the need for change and how this is requested, implemented and documented should be explained, as well as the risks of non-compliance or the absence of control measures.

Change Control

Definition: Ensuring all changes are made in a systematic and coordinated manner.

- Change Control is about governance. It's the mechanism that approves or rejects changes.

Example: An architecture (change) control or governance board could be responsible for this.

Change Management

Definition: The process responsible for controlling the lifecycle of all changes.

- Change Management is the overarching process that ensures all changes are made systematically.

Example: Having a formal approval process before any software updates are implemented

Configuration Item

Definition: An aggregation of hardware, software, or both, designated for configuration management.

- Configuration Items are the individual components that you manage within a system.

Example: A database schema could be a configuration item for which you track changes.

Requests For Change

Definition: Formal proposal for a change to be made.

- An RFC is essentially a formal request to make a change.

Example: If developers submit an RFC to update a software library, they would submit an RFC detailing why the change is necessary, what the impact would be, etc

Analysing Change - Impact Analysis

Definition: Assessment of the effect of a change.

- Impact Analysis is about understanding the ripple effects of a proposed change.

Example: Evaluating how a software update will affect other systems. Before updating a software library, an impact analysis would be conducted to see how this change would affect other interconnected systems.

QA

Risks

- Risk classification
- Risk identification
- Initial risk assessment
- Risk mitigation and residual risk assessment
- Risk monitoring

Effect	Frequency				
	Frequent	Likely	Occasional	Seldom	Unlikely
Catastrophic	E	E	H	H	M
Critical	E	H	H	M	L
Marginal	H	M	M	L	L
Negligible	M	L	L	L	L

Risks are defined here as anything that can affect delivery of critical stakeholder values, including quality requirements, or which can negatively affect resource consumption in the short term or long term.

Risk management needs to permeate all aspects of IT projects and operations.
Risks are everybody's responsibility, at a very detailed level of planning requirements, contracts, architecture and implementation.

Determining the risks that are relevant to an asset or object.
Qualitative risk assessment classes risk on a high-medium-low scale.
Quantitative risk assessment classes risk on a numeric scale.
The main deliverable of a risk assessment is the Business Risk Model.

Version Control

Definition: Management of changes to documents, programs, and other information.

- Version Control is all about keeping track of changes made to files over time.

Example: Using Git Hub for source code management. Git Hub, is a popular version control system used in software development.

Configuration Management

Definition: Identifying and defining the configuration items in a system.

- Configuration Management is the practice of systematically managing and documenting changes to configuration items.

Example: Using a tool like Ansible can be used to automate this process.

Quality assurance

1. Use a range of quality assurance tools or techniques.
2. Apply the principle of Separation of Concerns (SoC).
3. Use change management tools and processes.
4. **Continuous improvement.**

Continuous Improvement

- Plan, do, check, act
- Improving efficiency

Consider:

What is the benefit of continuous improvement (E.g. process, skills, etc.)
Have you used the process “Plan, do, check, act”? How did it affect the project/architecture?

You should be able to explain the purpose and concept of continuous improvement and the practical steps that are taken to implement these ideas in architecture.

QA

Plan Do Check Act

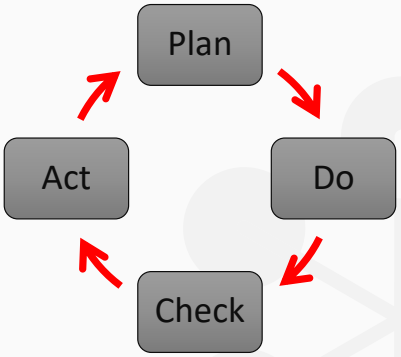
Plan, do, check, act - The Four-Step Cycle

(Shewhart / Deming Cycle)

Description:

A systematic series of steps for gaining valuable learning and knowledge for continuous improvement.

Note: ISO standards, especially ISO 9001, emphasise the PDCA cycle for quality management.



What is the ISO 9000 family of standards on quality management?

The ISO 9000 family consists of the world's best known standard for quality management systems (QMS), ISO 9001, along with a set of supporting standards on quality management, all published by ISO/TC 176 and its subcommittees.

ISO's seven quality management principles

The ISO 9000 standard outlining the fundamental concepts and vocabulary of quality management defines seven principles that all other quality management standards in this family are based on.
These principles include a strong customer focus, the active involvement and buy-in of top management, a process-oriented approach, and a commitment to continuous improvement. Further details can be found in ISO's brochure on quality management principles.

https://www.iso.org/standards/popular/iso-9000-family

Continuous Improvement

Continuous Improvement is an ongoing effort to improve products, services, or processes

- It aims to enhance efficiency, quality, and user satisfaction
- Provides continuous Improvement in Architecture

Improving Efficiency

Definition: Doing things right with minimal waste.

Practical Steps: in Architecture

- **Regular Reviews:** Periodically assess architectural designs.
- **Feedback Loops:** Incorporate feedback from stakeholders.
- **Training:** Keep the team updated with the latest best practices

Please open your workbooks to topic 6