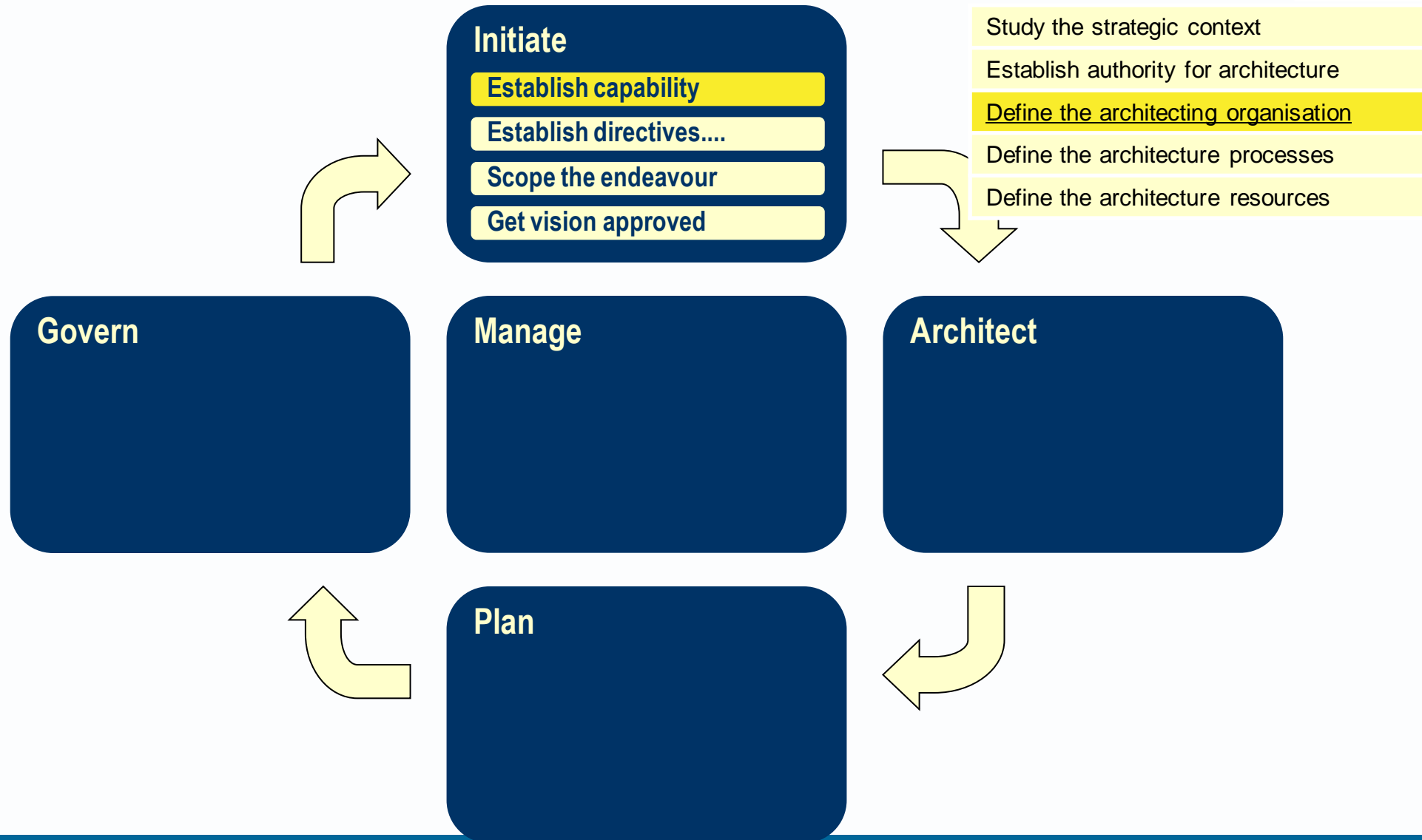


Avancier Methods (AM)

INITIATE

Define architect roles

It is illegal to copy, share or show this document
(or other document published at <http://avancier.co.uk>)
without the written permission of the copyright holder



The 4 primary architecture domains

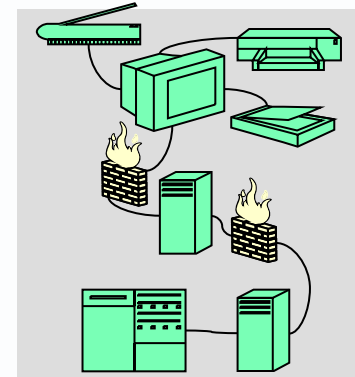
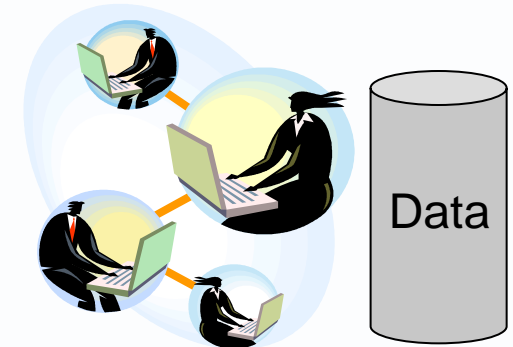
From Business to Technology

Architects support and enable a business by

- ▶ Focusing on business roles and processes that are
 - systemisable (repeatable and deterministic)
 - digitisable (create or use digitised data)

- ▶ Shaping and steering the portfolio of systems that
 - enable and support, monitor and direct
 - business roles and processes

- ▶ Ensuring a robust IT platform



The digitisation of business processes has enabled business to

- ▶ standardise and integrate business processes and data to a degree that was impossible before
- ▶ perform new information-related processes
- ▶ gather new kinds of business intelligence about entities and events of interest to business managers.



Timely and good quality information helps managers

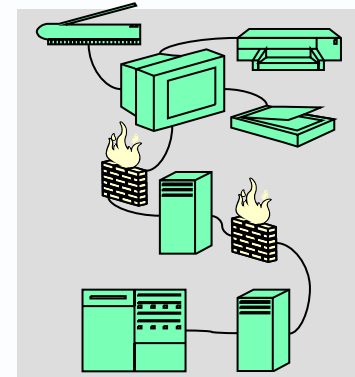
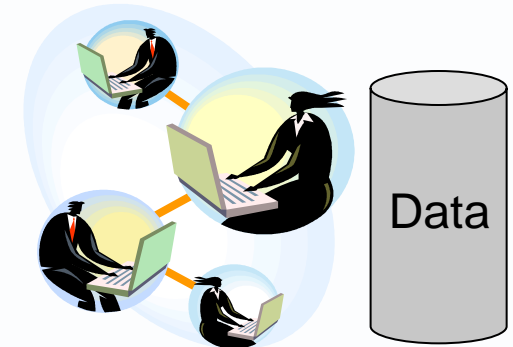
- ▶ Faster rate of change
 - Products and services change more frequently
 - Exponential growth in mobile devices and internet.

- ▶ Global competition and knowledge sharing
 - Workers available across the world - any time of the day.
 - Intellectual property is hard to protect
 - Cross-enterprise communities exchange information.



The four primary architecture domain/views

- ▶ The PRISM report of 1986 defined 4 views
 - Business (organisation and processes)
 - Data
 - Applications
 - Infrastructure (platform technology)
- ▶ These have appeared in countless frameworks
 - “EA Planning” (Stephen Spewak, 1993)
 - TOGAF
- ▶ The key elements in each view are defined later



Architecture roles by level

According to survey and standard

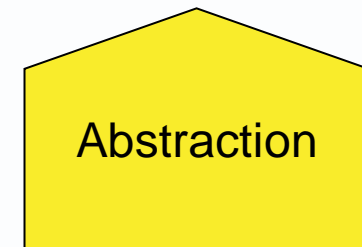
The top three architect job titles as a hierarchy

Architect job title in job advert	Survey 1	Survey 2
Technical Architect	960	141
Solution(s) Architect	177	433
Enterprise Architect	56	105

- Inverted in terms of scope and authority



Enterprise architect
Solution(s) architect
Technical architect (mostly specialists)



Seniority of architect roles

- ▶ Architect roles are seen as senior, directing, decision-making roles.



- ▶ But architectural decisions need to be understood at every level
- ▶ And architecture *description can be done by juniors*

- ▶ All solutions need architecting. The larger the enterprise, the more distinct architect roles emerge, and the more likely the solution architect will report to a central enterprise architecture team, and need to understand EA

“Skills Framework for the Information Age” (SFIA)

SFIA Role	Responsibility level						
Enterprise architecture					5	6	7
Solution architecture					5	6	
Project management				4	5	6	7
Business analysis			3	4	5	6	
Business modelling	2	3	4	5	6		
Requirements definition and management	2	3	4	5	6		
System design	2	3	4	5	6		
Database design	2	3	4	5	6		
Software development	2	3	4	5			
Database admin	2	3	4	5			

Q) When can I start out as an architect?

- ▶ In building architecture, “architect” is protected in law.
- ▶ You cannot put the term “architect” on your business card until you have qualified after an intensive 7 years.
 - academic education
 - practical work done under supervision.
- ▶ So, should an enterprise or solution architect should have 7 years experience of relevant projects?

Architecture domains and backgrounds

From Business to Technology

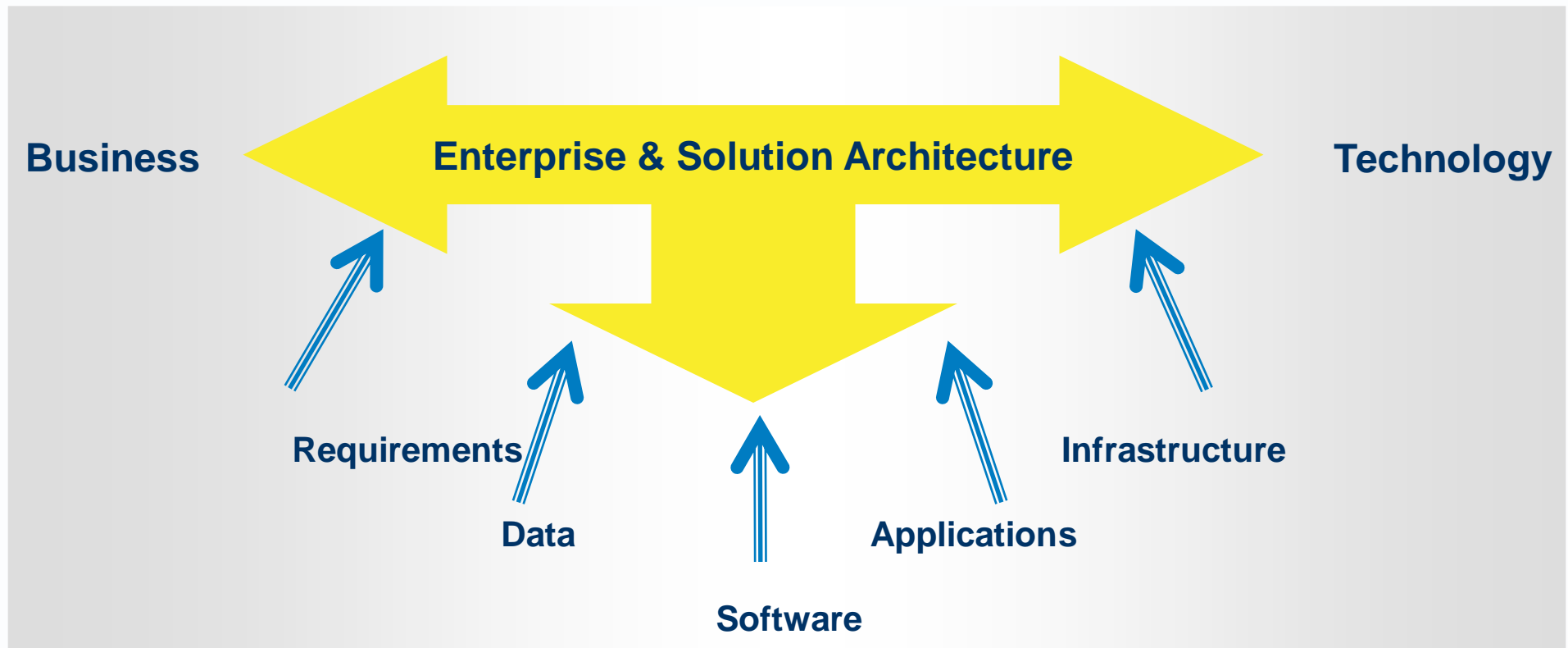
- ▶ There is no industry standard!
- ▶ An EA team usually divides roles by **level** and/or by **domain**

Architecture Work Space				
Domain/view	Business	Information Data	Applications	Infrastructure Platform Technology
Level				
Enterprise				
Solution				
Software/ Technical				
Operations				

- ▶ The power and the politics vary widely

Q) What experience do I need?

- ▶ Experience of detailed design and low-level architecture descriptions - in junior roles and narrower specialist domains.



Solution Architects

Solution Architect goals

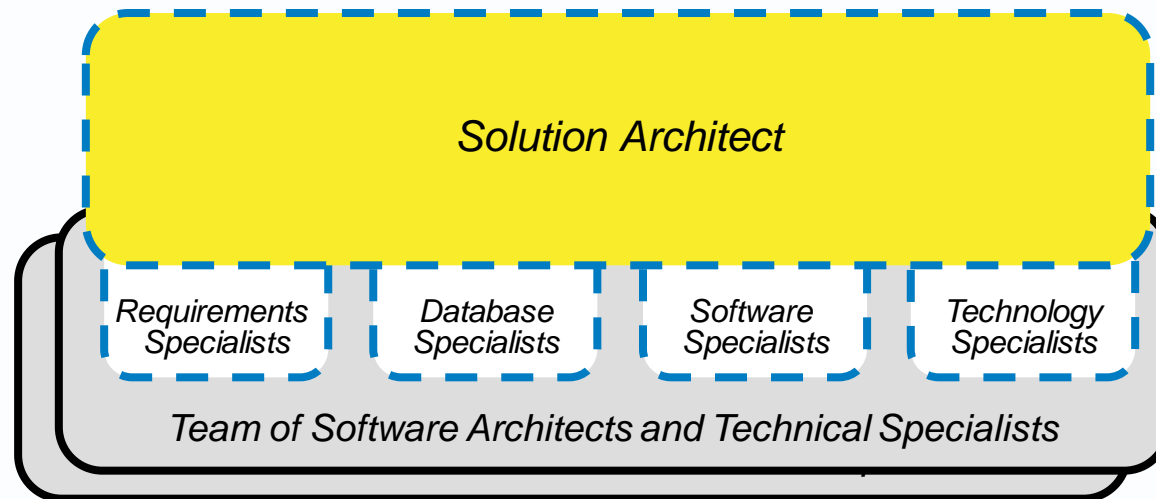
- ▶ Focus on success in solution delivery
- ▶ Design and deliver an effective and efficient solution
- ▶ Identify and manage technical risks along the way.

- ▶ Address sponsors and stakeholders who
 - have problems and requirements, and
 - want systems to support business roles and processes.
- ▶ Work closely with
 - project managers, EA and solution directors
 - business analysts and business change specialists
- ▶ Lead others by
 - shaping and direct solutions
- ▶ Attend early to
 - critical non-functional requirements and
 - physical design matters
- ▶ Govern delivery
 - may double as a project-level technical/software lead.

▶ SFIA says:

- leads architectures for complex systems, manages the target design
- co-ordinates design activities, promoting the discipline to ensure consistency.
- ensures relevant technical strategies, policies, standards and practices are applied.

- ▶ The right hand (wo)man of the programme/project manager.
- ▶ An experienced generalist who **joins up specialists** to deliver the solution
- ▶ **Smells out costs and risks, and ensures they are addressed.**



Without EA, silo (or point) solutions proliferate

- ▶ A silo is an organisation unit or application that:
 - ▶ is not standardised
 - does not follow the same rules or processes as another doing the same thing
 - ▶ is not joined up
 - does not share information with another doing something different
 - ▶ does not share/reuse common services
 - at the business or technology level.
- ▶ Silos are the result of architects being given only narrow project-specific objectives.
- ▶ Where to find the motivation and ability to avoid or reduce silo solutions?

Enterprise architect

EA is more *strategic* than SA

- ▶ SFIA defines EA development in 16 sentences in which
 - “**strategy**”, “**strategies**” and “**strategic**” appear 18 times.
 - “**setting strategies, policies, standards and practices**” appears twice

- SA is more
 - Tactical, Local, Concrete
- ▶ EA is more
 - Strategic, Cross-organisational, Abstract

- ▶ EA aims for integration and reuse of business systems
 - shared processes
 - shared data
 - shared services

- ▶ Prompts EAs to position an enterprise’s “operating model”
- ▶ in a quadrant of a standardisation/integration grid.

Positioning the “Operating model” for core business processes		
High integration	Coordinated	Unified
Low integration	Diversified	Replicated
	Low standardisation	High standardisation

- ▶ EA aims for integration and reuse of business systems
 - shared processes
 - shared data
 - shared services

EA is more *abstract than SA*

- ▶ EA works at the highest level of abstraction with
 - coarse-grained descriptions,
 - generic components
 - idealised/conceptual models, and
 - strategic road maps.

“The Enterprise Architect

- ▶ has the responsibility for architectural design and documentation at a landscape and technical reference model level.”
- ▶ often leads a group of the Segment Architects and/or Solution Architects related to a given program.”
- ▶ “elements in an enterprise architecture may still be **considerably abstracted from Solution Architecture, design, or implementation views.**”

TOGAF

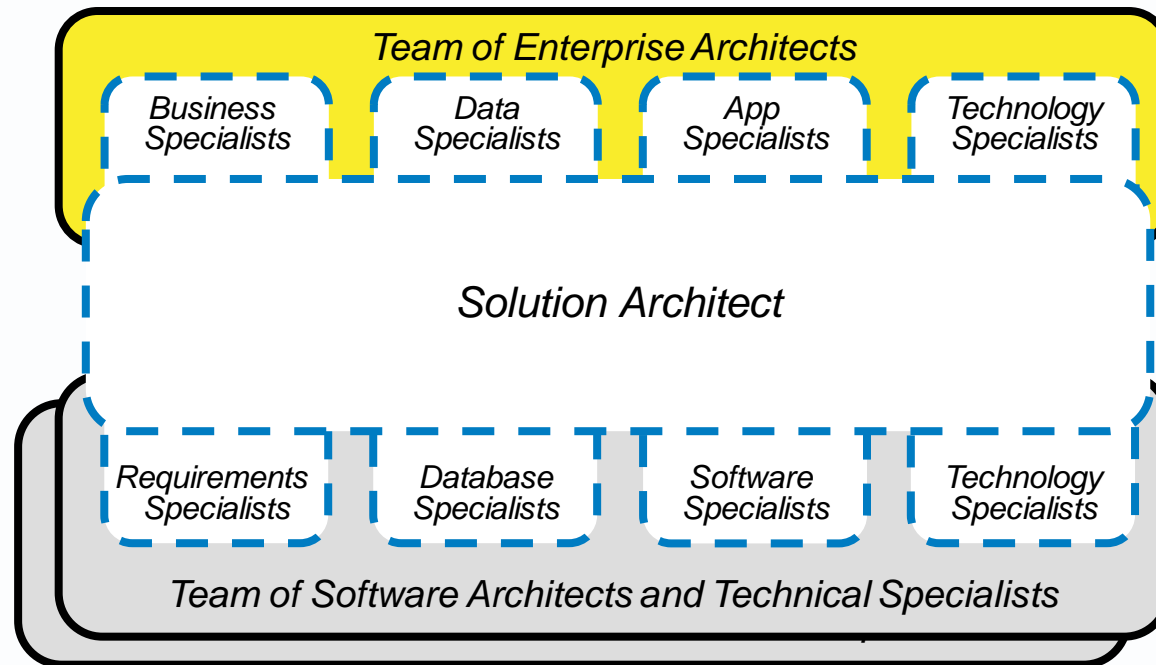
- ▶ Optimise an enterprise's many services and systems
 - Remove redundancy
 - Standardise services and systems
 - Improve the efficiency and effectiveness of the whole enterprise

- ▶ Has to
 - Understand the enterprise's estate,
 - Deliver cross-organisational road maps and EA collateral, and
 - Govern Solution Architects to ensure that solutions are
 - de-duplicated,
 - standardised,
 - interoperable and/or integrated.

Enterprise and solution architects together

An enterprise architecture team

- ▶ The EA team is often composed of domain/specialists who
 - take the cross-organisational & strategic view of their domain.
 - define a road map for their own domain
 - (which may cut across other domain road maps and business change plans)
 - not dedicated to any particular solution delivery.



AM classification by Architecture Domain and Architect Level

Domain Level	Business	Data/Information	Applications	Infrastructure technology
Enterprise level	Enterprise/Business Standardisation & integration of business roles & processes Business function/capability hierarchy Business products & services catalogue Business processes and roles Etc.	Enterprise/Data Data standardisation & integration Data store & data flow catalogues Maps data to business functions Business data model & views of it Canonical data model(s) Core business data entity life cycles Etc.	Enterprise/Apps Business app standardisation & integration Business app portfolio/catalogue Maps business apps to business functions Business app life cycles and road maps Etc.	Enterprise/Platform Platform standardisation & integration Platform technology portfolio/catalogue Platform services portfolio/catalogue (TRM) Platform technology life cycles and road maps Etc.
Solution level	Solution/Business For a required system/solution: Business services Business processes and roles Mappings to goals & locations Requirements catalogues Use case diagrams and definitions Outline UI (or other I/O) designs Etc.	Solution/Data For a required system/solution: Maps data to processes and roles Logical data models CIA requirements Data qualities/meta data Etc.	Solution/Apps For a required system/solution: Maps use cases to processes and roles Maps business apps to use cases Design for NFRs Coarse-grained app components Coarse-grained sequence diagrams Etc.	Solution/Platform For a required system/solution: Maps platform to business apps Platform technology definitions Client & server node definitions Design for NFRs Outline deployment diagrams Outline network diagrams Etc.
Software & technical level	Software/Business Detailed use case definitions Detailed UI designs Governs UI implementation Etc.	Software/Data Detailed database design Detailed message design Governs database administration Etc.	Software/Apps Detailed (fine-grained) software design Governs software development Etc.	Software/Platform Detailed deployment diagrams Detailed network diagrams. Governs platform and network configuration Etc.

Architect as designer and governor

one who designs buildings and superintends their
construction

People called 'architect' sometimes

- ▶ Play roles as
 - ▶ business analyst
 - ▶ manager
- ▶ But that is not our focus here

People called 'architect' sometimes

- ▶ Install systems
 - ▶ Manage operational systems
 - ▶ Monitor operational systems and diagnose faults
 - ▶ Mend a system when issues arise
 - ▶ Document a (baseline) system after it is built
- ▶ But really, that is engineering, operations, fire fighting, documentation

Our focus is on architecting

- ▶ An architect must understand and address the form and functions of system - ever mindful of costs and risks

“Architecture: The art or science of building. In a specific sense, one of the fine arts”

After Chambers 20th century dictionary

“Architect: Master builder [from the Greek].

One who designs buildings and
superintends their construction.

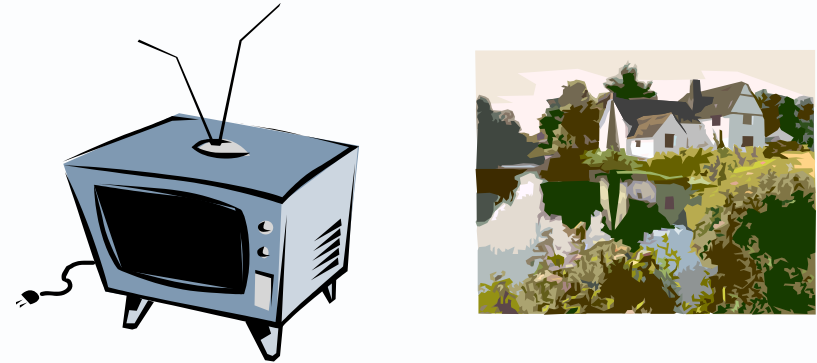
Any maker; a contriver.”

After Chambers 20th century dictionary.

- ▶ An architecture describes



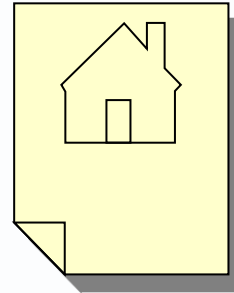
- ▶ A system is operational



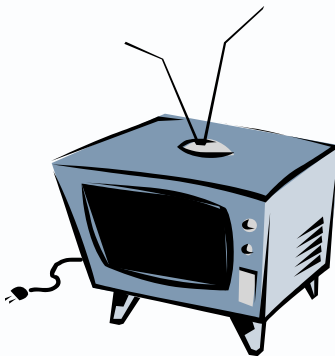
- ▶ specifies the structure and behaviour of a system;
 - ▶ can exist before the system is built, after the system is changed and after the system is destroyed.
- ▶ a collection of interacting subsystems
 - ▶ an encapsulated set of processes that transform input into outputs.

Architects learn to describe - as builders learn to build

- ▶ Architects have to learn how to produce plans containing architecture descriptions,

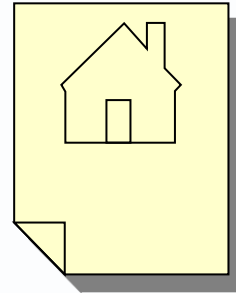


- ▶ Just as builders must learn how to build to those specifications.

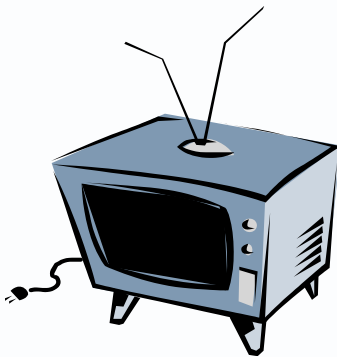


The architect as chief designer

- ▶ Given a customer's requirements for an operational system
- ▶ The architect must set out the form and functions of that system.

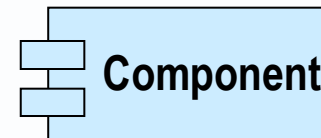
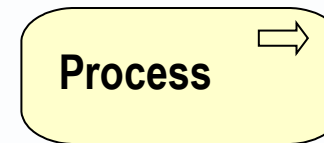
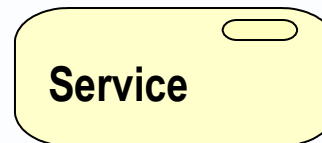


- ▶ Direct others in the detailed design and building of the system



- ▶ “Architect: Master builder [from the Greek].
- ▶ One who designs buildings and superintends their construction.”

- ▶ Given overarching precursors
 - mission, drivers and vision
- ▶ Clarify aims
 - goals, objectives and requirements
- ▶ Describe a systems’ behavioural properties
 - services and
 - processes
- ▶ And structural properties
 - interfaces and
 - components



Architects have to

- ▶ Understand business and technical contexts
- ▶ Understand design patterns and trade offs,
- ▶ Understand the strengths and weakness of materials
- ▶ Create and evaluate different options
- ▶ Make decisions
- ▶ Design and describe new (target) systems
 - To an acceptable level of detail

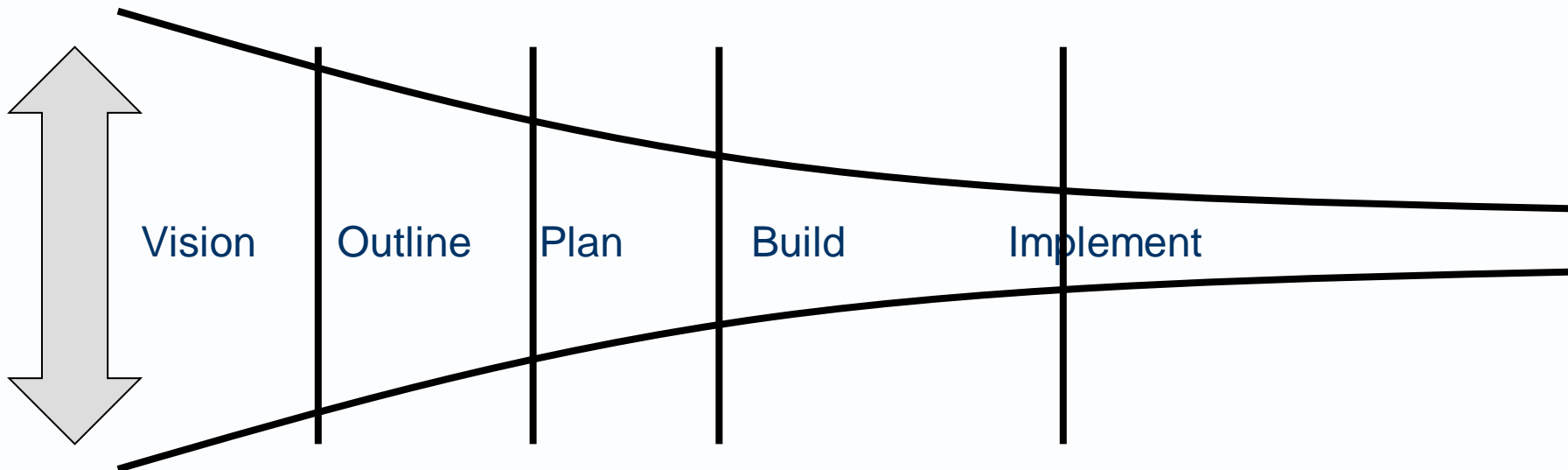
How far *can* an architecture description be refined?

- ▶ The level of detail depends on
- ▶ the breadth of the system or endeavour
- ▶ the constraints on the available time, money and resources.

Three dimensions of scope		
Breadth	Constraints	Depth
Size & complexity of system or project Large / Medium / Small	Time & resources to describe the system or project Little / Moderate / Lots	Level of detail reachable in descriptions or plans
Large	Little	Vacuous
Medium	Little	Sketchy
Large	Moderate	Sketchy
Medium	Moderate	Elaborate
Small	Little	Elaborate
Large	Lots	Elaborate
Small	Moderate	Fulsome
Medium	Lots	Fulsome
Small	Lots	Complete

How far *should* an architecture description be refined?

- ▶ Until the cone of uncertainty has narrowed sufficiently that
 - stakeholders understand the benefits, costs and risks
 - a decision to invest in the next stage can be made.



- ▶ Focus early on costs and risks associated with NFRs.
- ▶ Analysts complete functional requirements incrementally

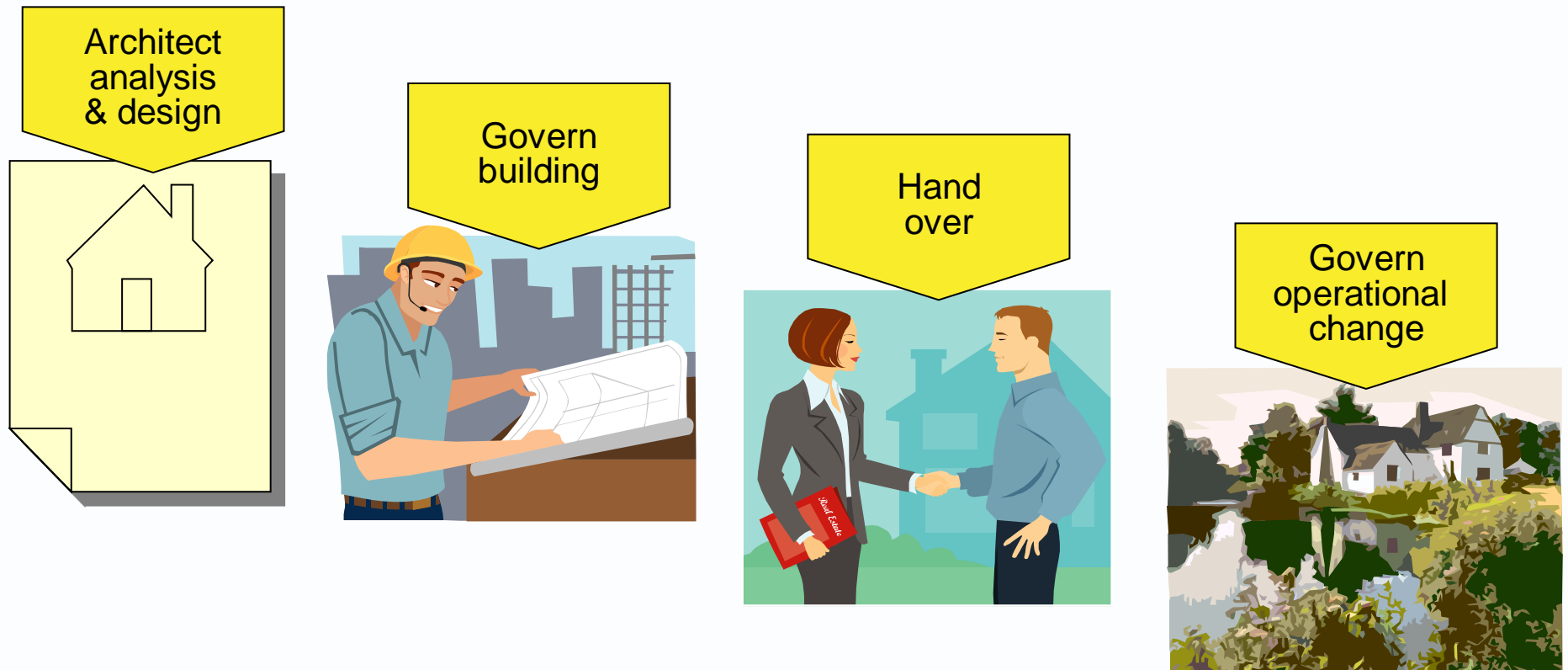
Architects should

- ▶ Identify fire risks and
 - ▶ design to prevent fires breaking out,
 - ▶ rather than fight them later.
-
- ▶ Which is a thankless task!

OK, but how much architecting does an architect do?

- ▶ "Architecting" is the (high level) design of the structure and behaviour of systems.
- ▶ "Architects" are accountable for this - even if they don't actually do it themselves.
- ▶ An architect may spend a minority of time on architecting - but they need to understand it deeply - to be accountable for it.
- ▶ Where architects are called in after major design decisions have been made by non-architects, the architect is left with a fire-fighting role, for which they also need to understand architecting.
- ▶ From a LinkedIn discussion

- ▶ “Architect: Master builder [from the Greek].
- ▶ One who designs buildings and
- ▶ superintends their construction.” (Chambers dictionary)



Architecture as abstraction

from lower level design

- ▶ *“Enterprise architecture is considerably abstracted from Solution Architecture, design, or implementation views.” (TOGAF 9).*
- ▶ An EA-level description might
 - list applications, but
 - *not* list the services (use cases) each application offers
- ▶ An SA-level description might
 - name application use cases in use case diagrams, but
 - *not* detail use case definitions.
- ▶ A software architecture description might
 - Detail use case definitions

Enterprise architecture compared with solution architecture

EA tends to	Design dimension	SA may (without an overarching EA)
Be relatively strategic (typically, 2 to 10 years)	Longer time <-duration-> Shorter time	Be relatively tactical (typically, 6 months to 2 years)
Look to integrate systems across the enterprise	Composition <-granularity-> Decomposition	Deliver a solution for a local function or unit
Standardise process and data definitions	Generalisation <-commonality-> Specialisation	Use parochial process and data definitions
Produces relatively abstract models and plans	Idealisation <-logicality-> Realisation	Produce relatively concrete models and plans

Enterprise architecture as strategic, holistic and abstract

- ▶ longer-term – more *strategic*
- ▶ wider - treats the *whole enterprise* as a system
- ▶ higher - *more abstract*

Architecture

Enterprise
Architecture

Solution
Architecture

Software
Architecture

1,000 days

Business
organisation

Business
process

Architect level	Strategic	Holistic	Abstract	Target
Enterprise Architect	Distant	Wide	High-level	Soft target
Solution Architect	Moderate	Moderate	Mid-level	Flexible target
Software Architect	Near	Narrow	Low-level	Hard target

30 days

Software
module

Executable
instruction

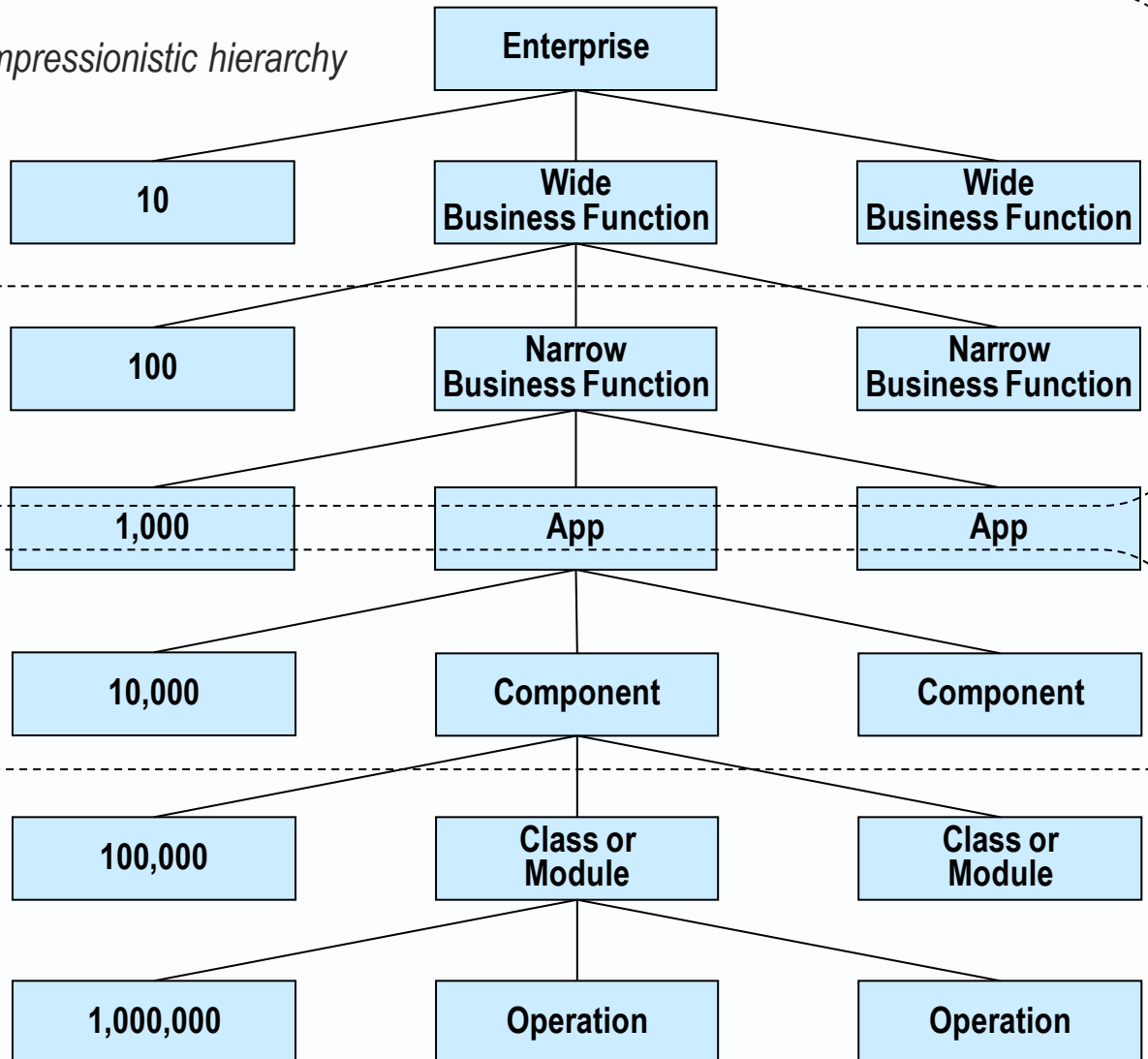
Abstraction of architecture description by *composition*

Enterprise Architect

Impressionistic hierarchy

Solution architect

Software architect



There should be more solution architects

- ▶ For every enterprise architect with a strategic 5 year plan:

Year	2019	2020	2021	2022
Enterprise architecture road map				
SOA programme				
Build enterprise service catalogue				
Identify integrations needing ESB				
Select ESB				
CRM				
Select CRM				
Implement CRM			Sales	Service

Elaboration

- ▶ Several solution architects are helping to shape and steer more tactical plans:

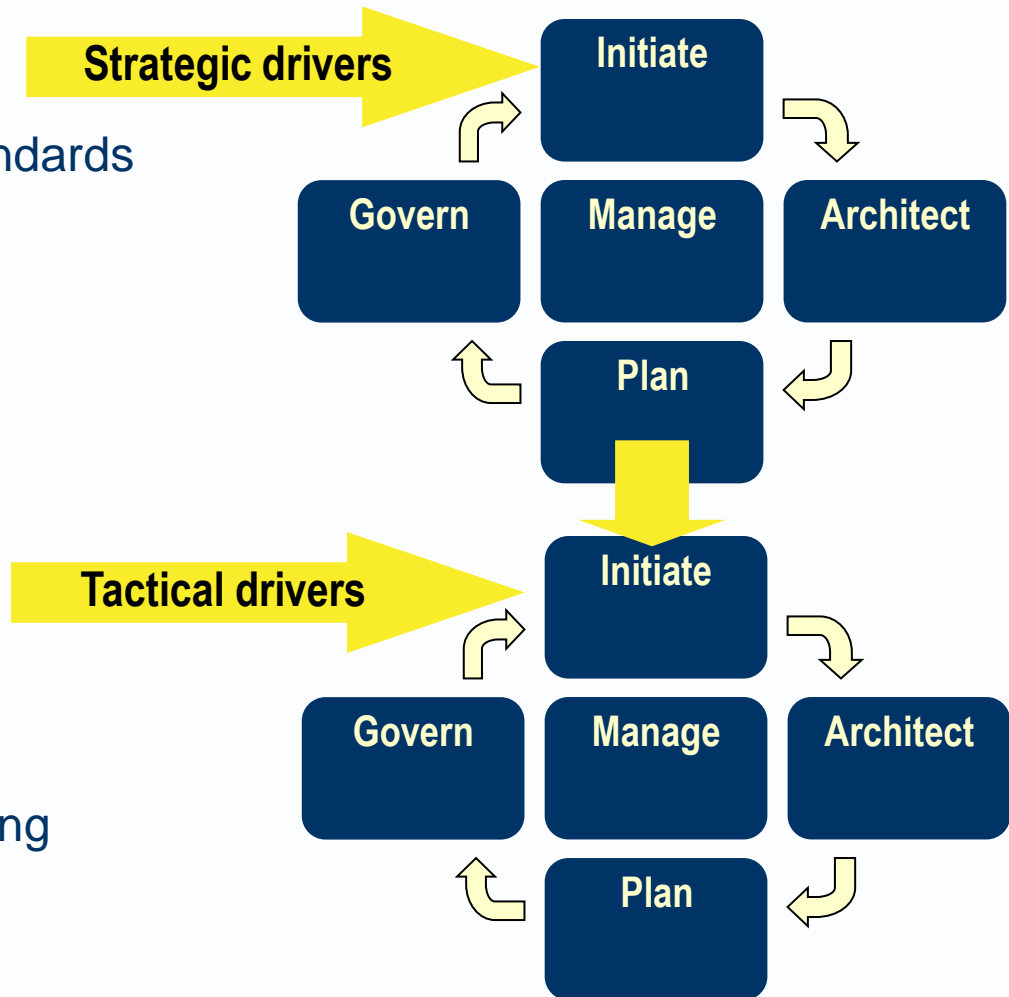
Quarter	2021 Q1	2021 Q2	2021 Q3	2021 Q4
Implement CRM				

► Enterprise architecture

- promotes and polices, common standards and principles
- promotes rationalisation
- produces high-level road map(s)
- governs solution architecture

► Solution architecture

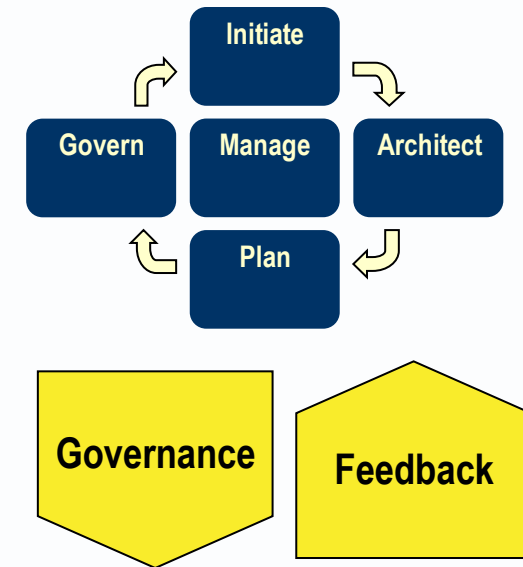
- specific problems
- solution delivery
- wrestles with the realities of designing and developing specific solutions



Two levels of goals, sometimes in conflict

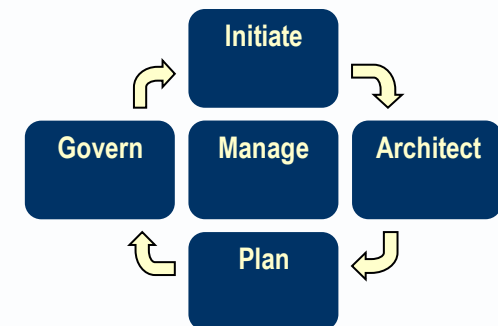
EA

- ▶ Rationalisation and refactoring of the estate
 - Portfolio management - prioritisation of changes
 - Consolidation and integration
- ▶ Business and technical agility
 - ▶ Faster and cheaper change
 - ▶ Imposition of standards and commonality across IS/IT developments.
- ▶ Enterprise-wide transformation
 - ▶ An overarching plan for driving towards a target,
 - ▶ Scoping of specific solutions.



SA

- ▶ Solve a problem – make a project-level change
- ▶ Assurance of functional and non-functional qualities
- ▶ Shape a solution that will work
- ▶ Minimize the overspend and manage the risks typical of IS/IT projects



Architecture tends to the left in some or all ways below

Higher level design	Design dimension	Lower level design
Strategies and road maps	Longer time <--> Shorter time	Shorter term sprints and deadlines
Broader goals, longer processes and coarser-grained subsystems	Composition <--> Decomposition	Narrower requirements, shorter process steps and finer-grained components
Generic standards, principles, patterns and reference models	Generalisation <--> Specialisation	Applications of standards, principles, patterns and reference models
Business needs and idealised system descriptions	Idealisation <--> Realisation	Physical technology solutions
External: services presented via interfaces	Encapsulation < --> Realisation	Internal: processes and components
Required behaviour: transient services and processes	Activities < --> Actors	Designed structure: persistent components, roles and interfaces

Architect roles in practice

How to divide architecture roles?

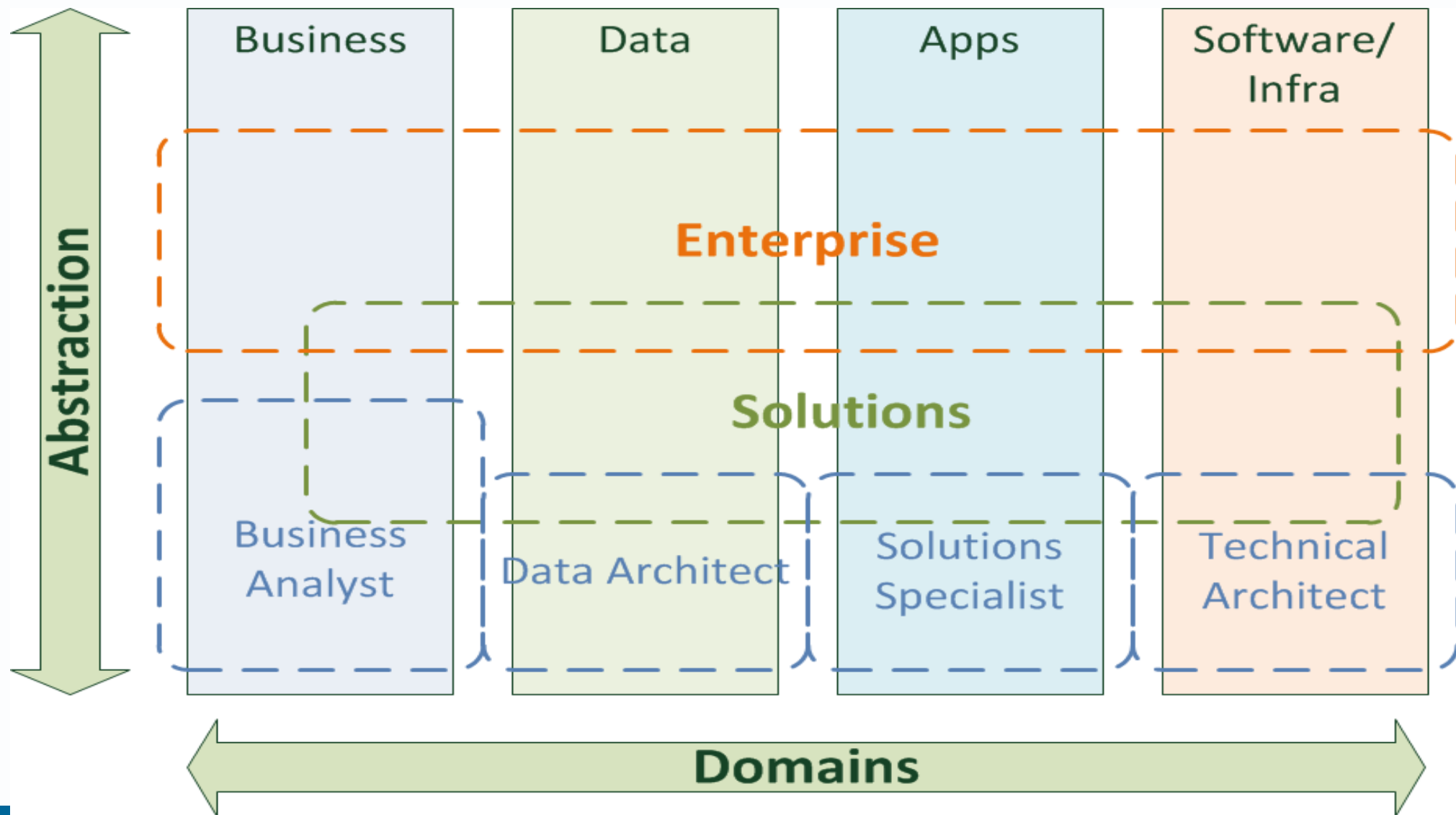
- ▶ Any real architect role may span rows and/or columns

The architects' working space				
Domain	Business Architecture	Data Architecture	Applications Architecture	Technology Architecture
Level				
Enterprise Architecture				
Solution Architecture				
Software Architecture & Technical Specialisms				

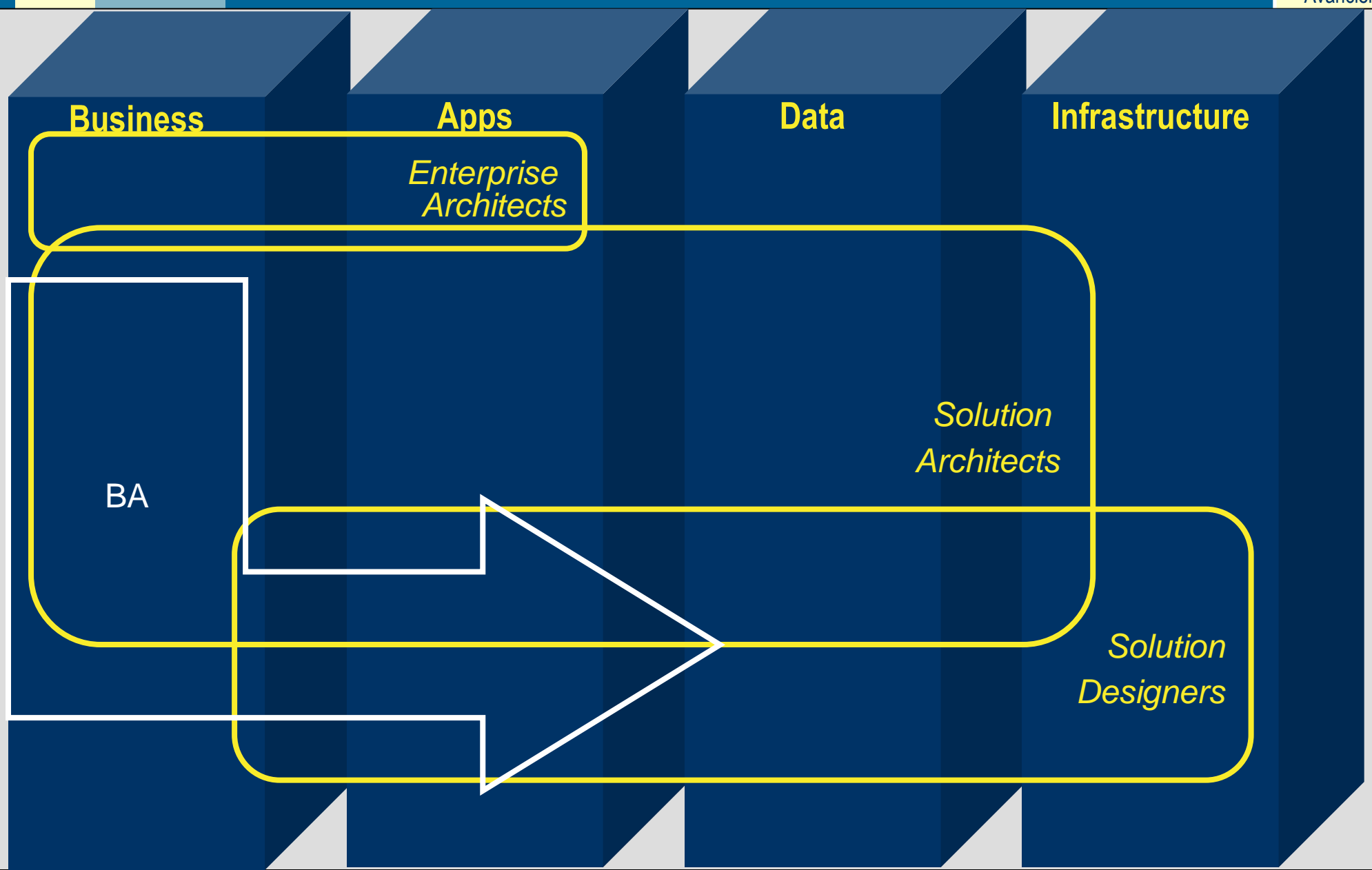
- ▶ Roles by level

- 
- ▶ Roles by domain

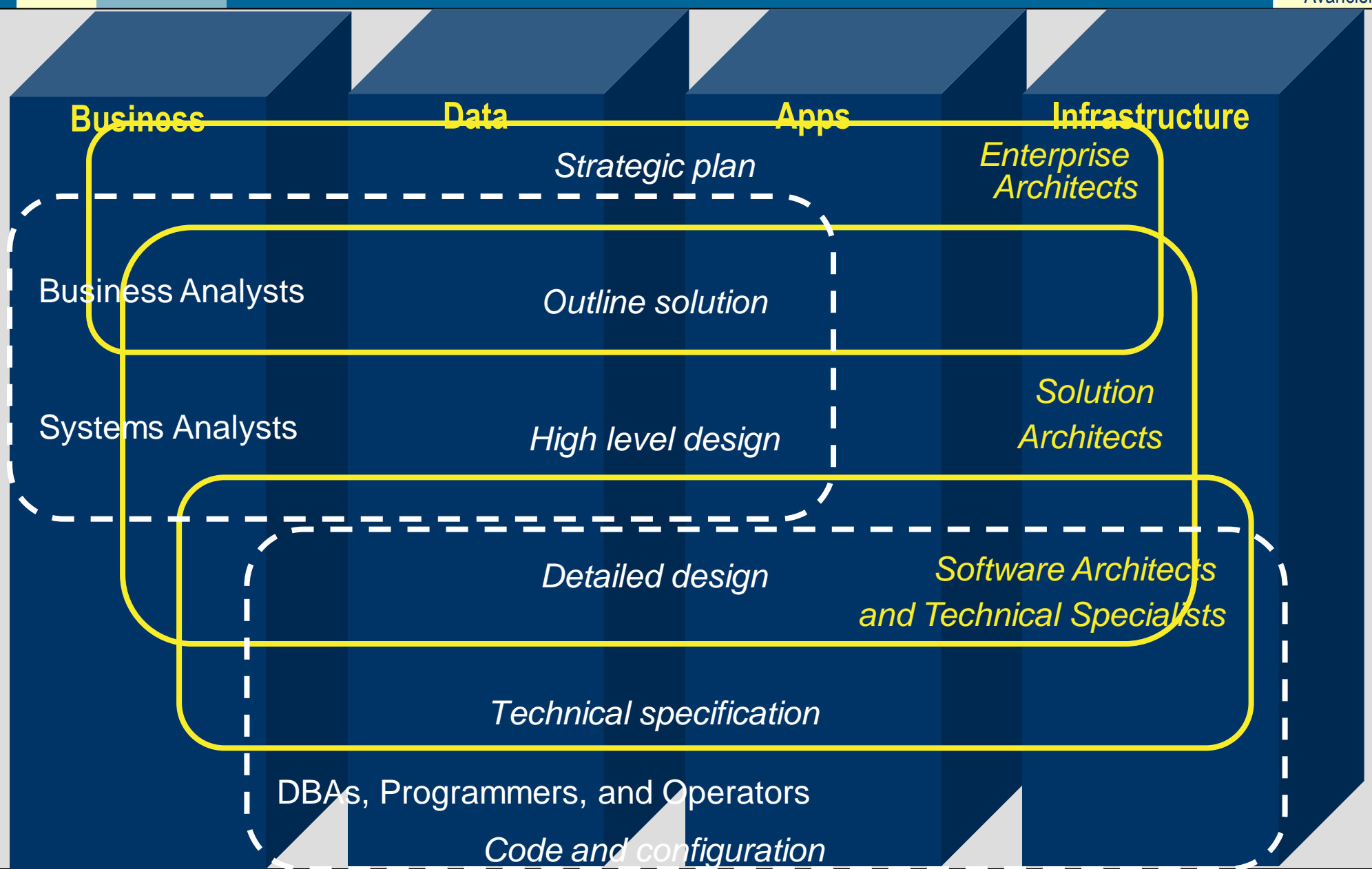
One organisation

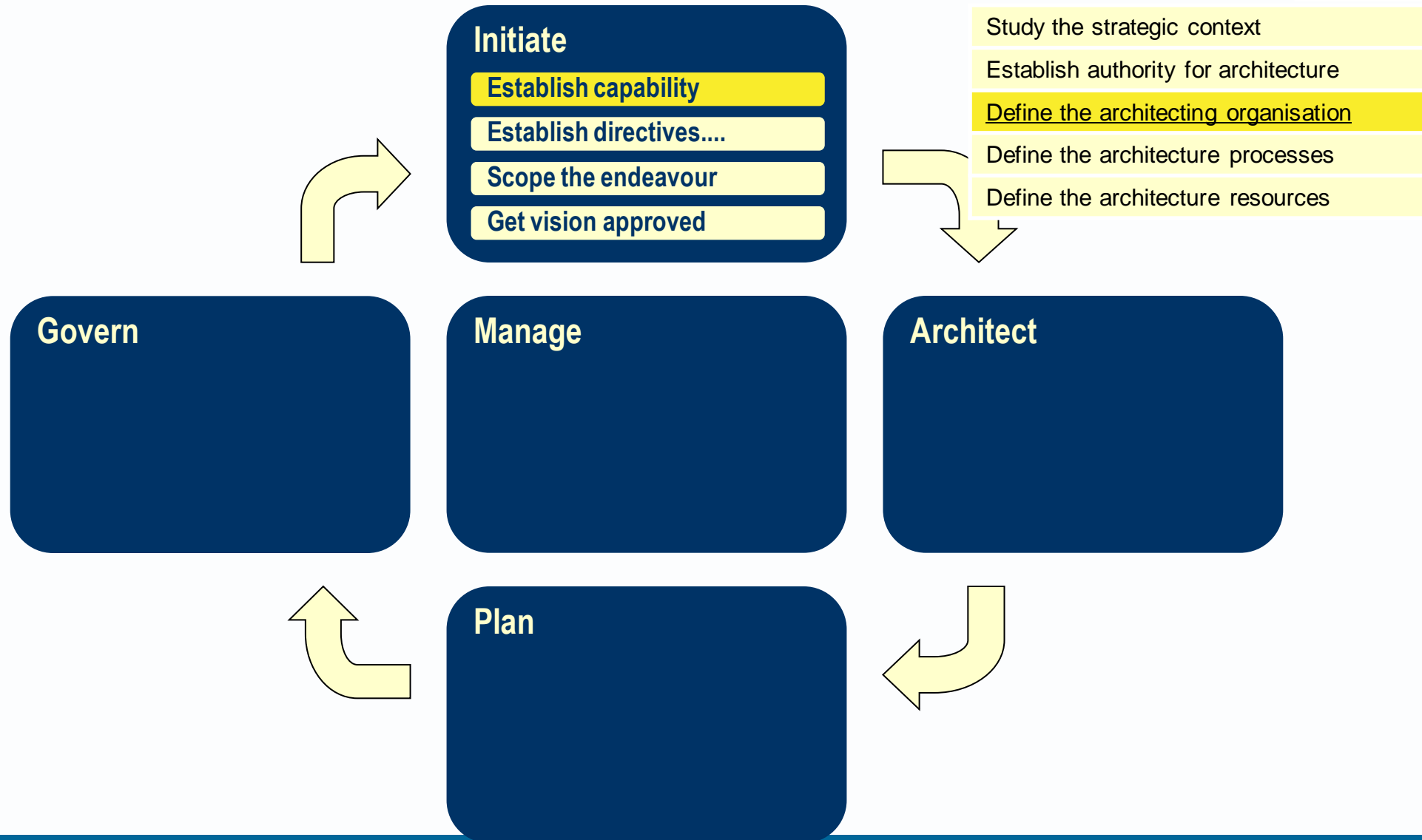


Another organisation



Mapping architecture domains, levels and roles to a traditional process with milestones and specification levels





Enhancing TOGAF with Avancier Methods

TOGAF's ADM is a change management framework that promotes the role of architects

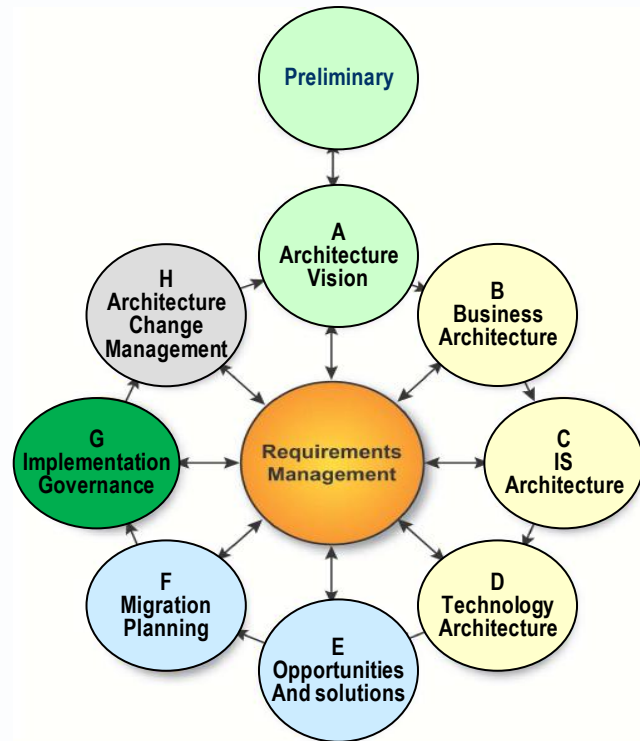
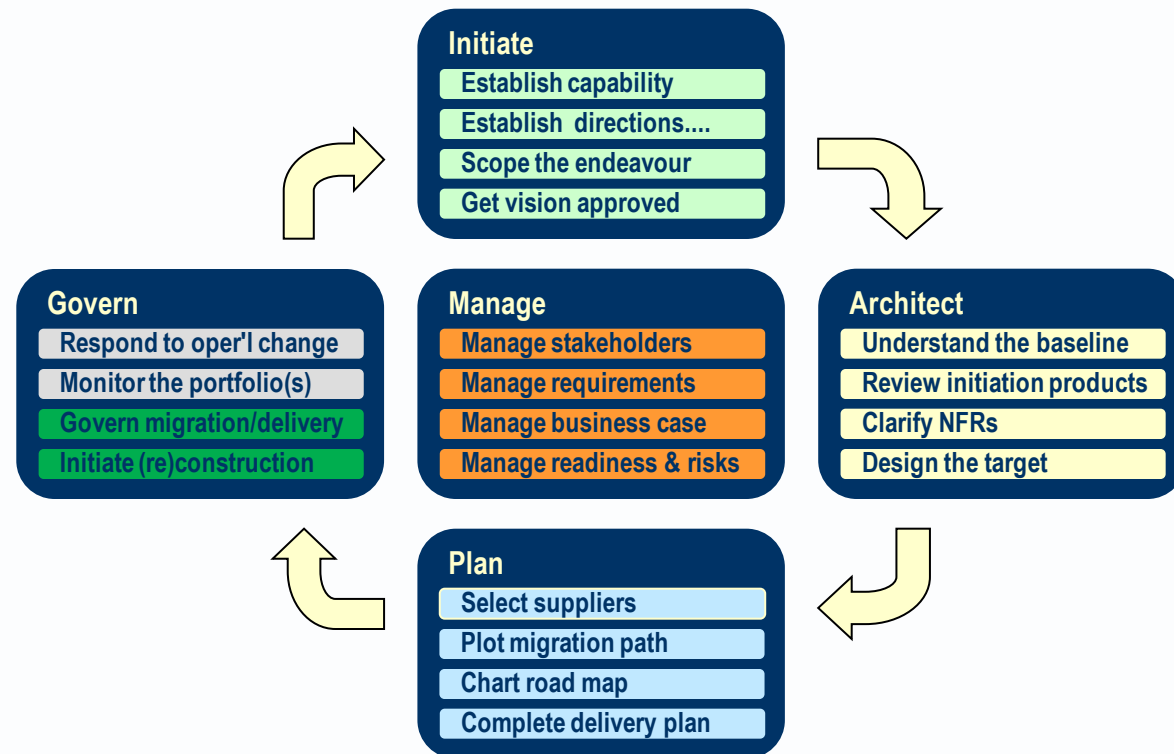


Figure 5-1 Architecture Development Cycle

AM gives architects more specific processes and documentation artefacts



- ▶ **Avancier Methods**
are useful with all
architecture
frameworks that share
similar ends and
means
- ▶ <http://avancier.co.uk>

