STYLES OF ARCHITECTING

A SMARTER APPROACH TO ARCHITECTING

A successful architecture effort starts with a meaningful dialogue between the customer of the architecture output and the architect. Architecting approaches tend to fall into a range of categories. If the architect and the customer are able to understand the category of architecting required for a particular problem then this gives strong guidance on how the architecture should be developed and governed. These categories of architecting are called Architecting Styles and provide a place to start an architecture effort; they provide guidance and help to set expectations with the sponsors and stakeholders.
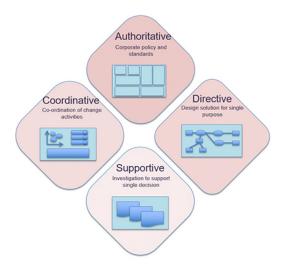

Figure 1 - The four styles of architecting

The Styles are a valuable tool in helping to overcome some of the issues that affect architecting which can result in architecture project failure. This can be caused by misplaced expectations as to the outcome and effort required (including maintenance) for the architecture, and consequently they are of no immediate value to decision makers. Key to this is that many approaches and Frameworks exist; however, there is not a good understanding as to what each one will deliver and that one size will not fit all. Some architectures will have longevity and require strong and sustained governance, others are only required to support shorter term objectives, such as to investigate a decision, or support a change activity. The architectural approach should take these differing requirements into consideration.

To gain a shared understanding of both the problem area and the approach, the Styles consider the relationships between key elements. These are the architecture Principles.

The **principles** of successful architecture will define the problem and approach, and guide you to the correct style:

- **VALUE**. The *Purpose* of the architecture will directly drive the shape and form of the *Outputs And Outcomes*;

- **SCOPE**. The *Area Of Concern* will dictate what *Reference Models* or standards need to be considered and (re)used;

- **CONTROL**. The *Level of Change* will identify the type of *Governance* that needs to be applied and by whom;

- **DELIVERY**. There is a strong synergy between the *Development Method* and the *Enablers*. This relationship will be influenced by each of the other components in turn as reflects the constraints of the approach, eg use of a corporate tool or framework.

Use these Principles to understand the problem and inform the approach.

The principles are shown below along with some of their attributes. These should be considered during dialog between the architect and sponsor to fully understand the **context** of the problem area and from this the **approach** that is best suited to achieve the desired benefit from the architecture effort.
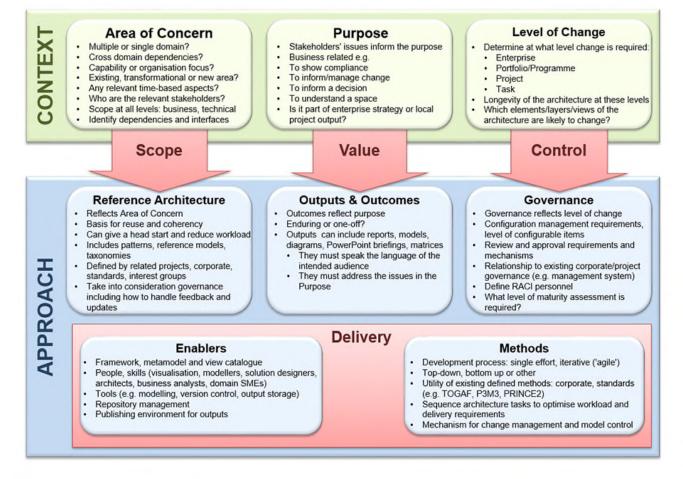

Figure 2 - The principles of architecture.

Once the sponsor and architect have considered the Principles and fully understood the context, the architecture approach should be developed using the characteristics of the four Architecting Styles as a guide.

The figure below summarises the characteristics that are detailed in more depth in the tables below.

Style	Formality	Persistence	Granularity (detail)	Utility	Effort
Authoritative	Strong governance - aligned to enterprise objectives	Enduring - similar to enterprise development	Either high level (breadth) or focused detail (depth) but unlikely to be both	Single purpose	Reasonable effort needed to develop, maintain and communicate
Directive	Strong governance - aligned with established systems engineering practices	Similar to life of programme or project	Level of detail similar breadth or depth of programme or project	Limited purpose	Considerable effort to develop and maintain
Coordinative	Selective governance - focus on validity of source information	Valid for duration of source information	Aggregated information from projects	Multiple purposes	Reasonable effort to support decision makers
Supportive	Review to ensure validity of recommendations	Focused on specific intervention points, with limited reuse post this	Limited detail focused on single decision	Limited or single purposes	Focused effort to support a defined decision point

Figure 3 - The characteristics of the four architecture styles

Authoritative

Provides direction/policy to one or more domains to drive coherency, consistency, reuse and alignment with corporate objectives. The resultant architecture is an enduring reference source for other architecture activities and supports all phases of the enterprise life cycle. It requires constant management and strong governance to ensure it remains relevant and valid. A specialist team commonly enables this style with a wide remit for coherence or interoperability across the enterprise.

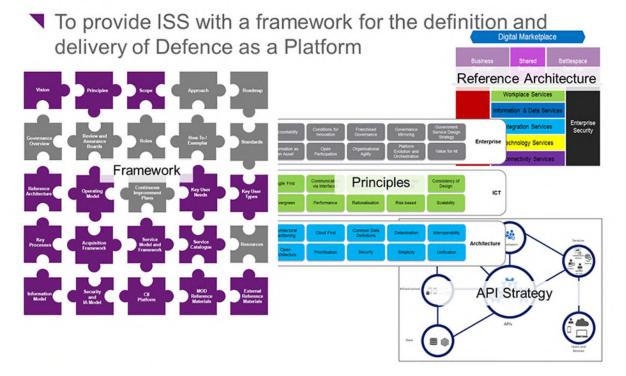| Context		Approach
Purpose	**VALUE**	**Outputs & Outcomes**
• Defines or support standards or policies for single or multiple domains to ensure coherency. • An enduring need that supports both change and business as usual.		• Standards or polices (or contribution to) and Reference Architectures that are reusable. • Normally published on corporate intranet.
Area of Concern	**SCOPE**	**Reference Model**
• Applicable to an established domain to reflect best/common practice. • Can cover strategic, business, systems, information and technology perspectives; but breadth is priority over depth.		• Will have connections to wider best practice/sources outside of the enterprise. • Initial development draws on existing artefacts to reflect best practice.
Level of Change	**CONTROL**	**Governance**
• Typically at the enterprise level to drive a coherent Systems of Systems approach. • Reflects established business practices rather than acting as a starting point for change. • Aligns, or in turn drives, corporate governance.		• Requires strong governance to ensure it remains relevant and valid to the enterprise. • Aligns to corporate structures. • Ongoing assessment of applicability and value.
Development Methods	**DELIVERY**	**Enablers**
• Continuous activity to refine and maintain artefacts and published on a regular basis, eg www or company intranet. • Draws on patterns and taxonomies and underpinned by corporate principles. • Ensures stakeholder are bought in by strong communications.		• Supported by a team with a wide remit for coherence or interoperability across the enterprise. • Varied tools can be used depending on scope and hosting environment. • Commonly use 'community of practice' to support development.

Example: Authoritative

A key foundation to realising the vision of information capabilities that serve as a "force multipler" is the establishment of Defence as a Platform (DaaP). DaaP is the means by which the delivery of information capabilities moves towards a shared-platform model in order to provide a more cost-efficient and unified service.

Niteworks were tasked with developing the strategic reference architecture for DaaP, drawing on pan-industry input to define a mature view of the capabilities required to realise the vision. The Niteworks partnership between MOD and over 160 industry organisations was the ideal vehicle to deliver this project.

Niteworks developed the strategic reference architecture for DaaP and documented the results in the 'DaaP jigsaw' which provides a framework for the guiding principles, reference models, processes required for the continued iterative development, delivery and maintenance of DaaP.

Authoritative Example - Defence as a Platform (DaaP)

◥ To provide ISS with a framework for the definition and delivery of Defence as a Platform

Coordinative	
Supports the co-ordination of change activities within a single domain or sub-domains by aggregation across lower levels; it is normally used at the Programme or Enterprise level. Governance is needed to ensure that the supplied Management Information (MI) is of suitable quality to support required decisions. The approach draws heavily on broader P3M practices and may be part of a Programme Support Office (PSO). Tooling is specialised with aggregated data being presented through dashboards or composite graphics to meet different decision makers' needs.	

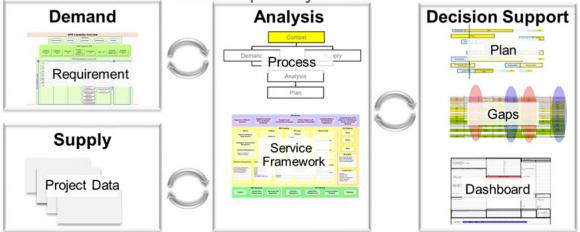| Context		Approach
Purpose	**VALUE**	**Outputs & Outcomes**
• Support co-ordination of change activities by understanding dependencies of: milestone, risks, capability, process, interfaces. • Typically an enduring requirement that requires regular reporting. • Architecture provides more robustness/efficiencies to existing processes/practices.		• Example outputs include roadmaps, dashboards, heat maps to support. • Programme board MI. Capability maturity assessment. Technology/product roadmaps. • Outcome is regular reporting to managers on state of project, programme, capability. • Is used to make decisions on or report on, the direction of the programme or Project or capability.
Area of Concern	**SCOPE**	**Reference Model**
• Typically monitors business as usual and business change to drive decisions for change. • Typically about the aggregation of concepts/issues of a given stakeholder/role eg senior responsible owner or design authority.		• No specific reference models but may need to comply with enterprise reporting requirements or information models. • Where applicable, 'common language' drawn from reference model is used to ensure consistent reporting.
Change	**CONTROL**	**Governance**
• Enterprise, programme, project. • Used as enduring resource to report on change and make decisions on change.		• Assurance of MI must be of suitable quality to support decisions. • Needs to be of value to the stakeholder and in turn is supported by the stakeholder to drive information quality.
Development Methods	**DELIVERY**	**Enablers**
• Strong information management that draws on broader P3M and PSO practices. • Periodic maintenance and update by a specialist team. • Quality of source data key to success. • Identifying the required KPI of the business. • A strategy for the integration of corporate data.		• Initially developed by a small development team. Team skills should include information/data management, visualisation and dashboard development. • Direct access to corporate data. • Used as part of business as usual so needs to be integrated into corporate infrastructure. • Uses specialist MI tools that interface with other databases.

Example: Coordinative

The ISTAR & IO Enterprise has historically provided specific, bespoke, domain and platform centric solutions to deliver Military Capability. The Direct, Process & Disseminate Capability Investigation (DPD CI) sought to identify gaps, overlaps, shortfalls, redundancy and project inter-dependencies in the DPD portfolio in order to support the development of a coherent, affordable and viable DPD Capability Management Plan.

The project developed a Capability model and Service Taxonomy based upon the capability requirements specified in Doctrine. The projects within the DPD portfolio were profiled against the Service Taxonomy, which enabled the identification of functional overlaps and interdependencies in the planned provision of the capability. The Enterprise Model provided the ability to clearly articulate the Capability Requirements to the Delivery Organisation and expressed the contribution of projects in terms of Capability. It gave a Common and improved understanding of the enterprise, meaning that it improved the ability to make decisions holistically across the enterprise. Capability Gaps and overlaps were easily identified and the impact articulated, additionally the use of differentiators to characterise projects allowed the prioritisation of further intervention. This architectural approach resulted in the identification of significant savings in the delivery of ISTAR & IO capability.

Coordinative Example - ISTAR Delivery Team

◥ On-going support to identify gaps, overlaps, shortfalls, redundancy and project inter-dependencies in the Direct, Process & Disseminate (DPD) portfolio to drive a coherent, affordable and viable capability.

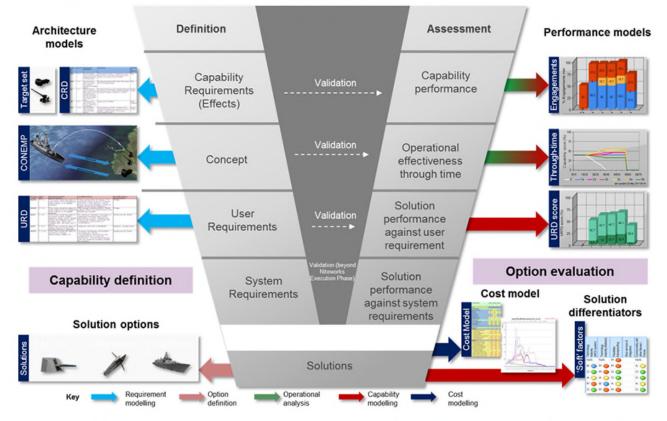| Supportive		
A focused style that supports key interventions or decision points across all levels of change. It can also be used as an initial activity to identify the need for change or explore an area of interest. It can function with limited governance (eg peer review) but must be held to account when supporting key decisions. Normally undertaken by a small team of dedicated architects working as part of a wider team using best available tooling including standard office IT products.		
Context		**Approach**
Purpose	**VALUE**	**Outputs & Outcomes**
• Typical aim is to understanding a complex space or issue. • Emphasis is on bounding the problem rather than solving it. • May be the initial activity for further work using different styles/approaches. • Normally a one off but review may occur of issue but not an enduring ref architecture.		• Architecture views form part of a final report/briefing, a body of evidence or is a coordinating tool. • Visualizations provide insight tailored to the business NOT the architect. • Key output is next steps/recommendation to resolve issues, if found.
Area of Concern	**SCOPE**	**Reference Model**
• Support to key intervention points in the life of the programme or project. • Looks at emerging features of a System of Systems (SoS) or capturing legacy SoS design.		• Will draw on relevant reference models or standards as needed. • Draws on broader concepts that help understand the problem space.
Change	**CONTROL**	**Governance**
• Is in response to a customer need and can be at any level.		• Can function with limited governance (eg peer review) but must be held to account when supporting key decisions. • Focuses on either breadth or depth to support decision.
Development Methods	**DELIVERY**	**Enablers**
• Approach is tailored to support specific purpose. • May reuse existing approaches from previous engagements but needs to be flexible. • In certain circumstances architect may maintain model to support response to future requests/questions. • Typically a short duration task. • Key to success is experience of team rather than a robust architecture.		• Small team or individual working as part of a multiple-discipline team to support business decision or intervention. • Range of dedicated tooling or standard office IT can be used. • Emphasis is on visualisation rather than robustness of model. • Artefacts usually used once and become redundant once task complete.

Example: Supportive

Niteworks provided support to the Future Maritime Fires programme through the generation of Requirements, updating previous Operational Analysis (OA) and the development of Capability and Cost Models. These outputs were exploited to improve and expedite decision making, reduce risk and support the Approvals process within MOD. Niteworks specifically examined the potential Medium Calibre Gun (MCG) solutions for T26 Global Combat Ship (GCS), T45 and/or T23.

The Capability Model integrated the cost and performance models in Microsoft Excel with the capability definition architecture in the architecture modelling tool, MooD, to provide a holistic view of effectiveness, cost and risk for the Capability options. Priorities assigned in the URD were used by the Capability Model to weight the score of the MCG options against Requirements. Whole life costs were taken from the cost model, and operational effectiveness, in terms of the number of engagements satisfied by each weapon system, were drawn from the operational analysis.

The resulting Capability Model supported MOD to make and brief decisions on MCG solutions by capturing and presenting visual summaries of the relevant data in one place.

Directive

Plays a key role in the development, definition, design and implementation of new capabilities, areas of responsibilities, processes or systems. It is applicable to all domains but is normally focused on a single domain or sub-domain that is planning for, or going through change. This style will usually use relevant reference models, policies or standards along with a range of dedicated/specialist tooling.

Context		Approach
Purpose	**VALUE**	**Outputs & Outcomes**
• Focus is on delivering a new or change to an existing capability, process or system. • Ensuring that the system solution is justified by the Business Need. • Identification, development and justification of options for change.		• Configured artefacts that reflect the engineering process/need of the project including handover. • Outputs include 'as is' and 'target 'architecture; transition plan; requirements; options. • Solutions are complaint with the System of Systems architecture.
Area of Concern	**SCOPE**	**Reference Model**
• Focus on a single domain or x-domain. • Typically focused on supporting a phase (or phases) of the programme/solution life cycle. • Aligns with business/commercial boundaries. • Cognisant of legacy solutions and the need to integrate or replace.		• Will use relevant reference models or standards. • Strong reuse of organization's and industry best practice. • Customer needs to avoid risk of preventing innovation by supplier. • Outputs may in turn be feed back into existing reference models.
Change	**CONTROL**	**Governance**
• Predominately at the programme, project or task level. • Needs to part of established programme, project structure with a Single Responsible Owner. • Business change normally driven by relevant operating group.		• Strong governance that aligns with existing engineering practice and design authority(s) to align with enterprise requirements. • Clear ownership and sign off of artefacts required. • Acts as a co-ordination tool between multiple organizations.
Development Methods	**DELIVERY**	**Enablers**
• Follows prescribed process as part of delivery or service provision including change process, eg Programme Management Plan or the System Engineering Management Plan. • Strong collaboration with all stakeholders to ensure validity. • Key to success is applying a pre-considered and robust architecture method from the outset.		• Supported by a recognized architecture or systems engineering tool set; federated tooling is often needed. • Established framework and notation. Architecture library to identify reuse of existing material. • Enabled by a multidisciplinary team responsible for artefacts. • Suitably qualified and experienced architects for given purpose and area of concern.

Example: Directive

Defence Operational Training Capability (Air) (DOTC(A)) will radically change the way that Defence conducts operational training across the Enterprise in the air domain. DOTC(A) will be a complex system of systems that requires careful planning and design so that it can establish the correct people, organisational structures, technology and business processes to succeed.

An Enterprise Architectural approach, underpinned by MOD Architecture Framework (MODAF)-based modelling has been used to support analysis of the user requirement and the subsequent development of the system design. The system design has been captured as a series of Enterprise Architecture views and descriptions from a variety of specific stakeholder perspectives and are collectively referred to as the DOTC(A) Enterprise Architecture Model (EAM).

The DOTC(A) EAM has been specifically configured to support the capture and definition of DCS&S system requirements in an Acquisition Systems Guidance (ASG) compliant format. A methodology has been successfully developed, tested and executed to provide the captured DCS&S system requirements in a suitable export format that can be directly exploited in the generation of a System Requirements Document (SRD). And in the future can be synchronised between the
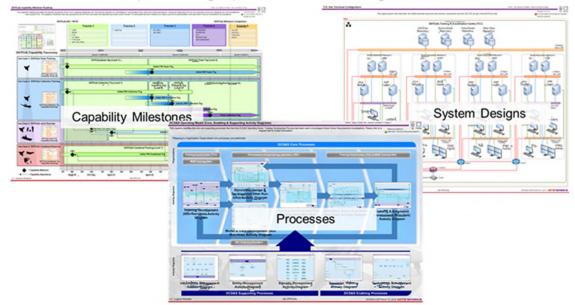
MOD's preferred requirements management tool (IBM Rational DOORS) and the DOTC(A) DEAM development tool, (MooD Business Architect).

The DOTC(A) EAM has also been developed to incorporate other related Niteworks DOTC(A) outputs, including the Integrated Test, Evaluation and Acceptance (ITEA) Framework Guide, Test Case Scenario, Model Based Requirements Navigation and a Mission Analysis Model Directory.

The DOTC(A) Mission Analysis Modelling activity that has been conducted during previous stages of Niteworks DOTC(A) support had the primary purpose of informing and ensuring the DCS&S system requirements, as captured in the DOTC(A) EAM, are robust. DOTC(A) Mission Analysis Models (MAMs) have also been exploited for other purposes, which have provided additional benefits to MOD. By conducting mission analysis for the agreed DOTC(A) Force Elements, all interaction types and entities can be identified. Therefore, representing the 'Training Requirement' that effectively must be delivered by the future DOTC(A) synthetic training capability.

See also the Niteworks White Paper 'Styles of Architecting - A smarter approach to architecting the Defence Enterprise'

Author: John Kendall
John Kendall is the Chief Architect at the Niteworks partnership. John has been involved in Enterprise Architecture for over 10 years and has been Lead Enterprise Architect on a number of large programmes delivering capability or performing business

transformation. John was also a member of the MODAF Partners team who originally developed the MOD Architecture Framework (MODAF).