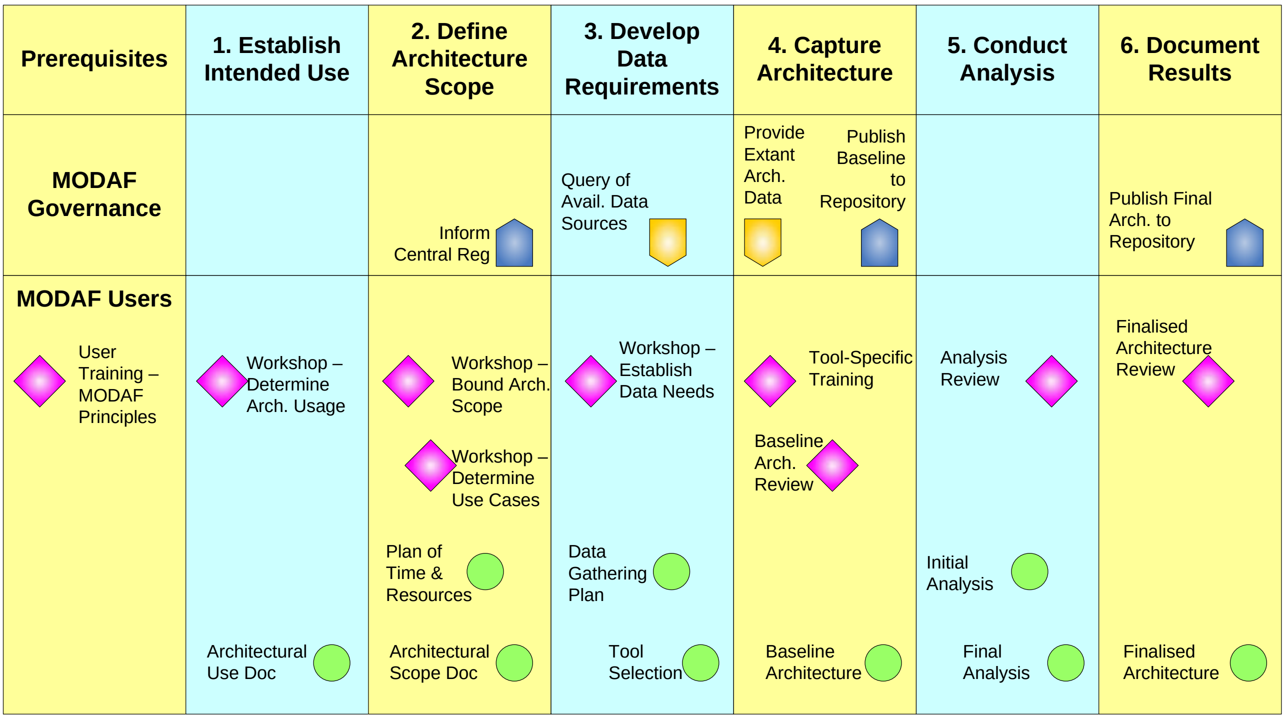
THE MODAF ARCHITECTING PROCESS

What is the MODAF Architecting Process?

The overall approach to developing a MODAF compliant architecture is broadly the same regardless of which MOD community is doing the work, or the MODAF views that are being generated. However, the MOD does not prescribe a “MODAF Method” for architecting and creating MODAF views. What this document presents is an example of one approach to take; there are many different ways to approach the architecting process.

In reality, few, if any, teams within the MOD will simply follow the general six-step process outlined above from start to finish once only and then not utilise the architectures again. In practice there will be a wide variety of approaches to conducting architectural work that will involve various iterations and variations around this general process.



Prerequisites	1. Establish Intended Use	2. Define Architecture Scope	3. Develop Data Requirements	4. Capture Architecture	5. Conduct Analysis	6. Document Results
MODAF Governance		Inform Central Reg ⬟	Query of Avail. Data Sources ▽	Provide Extant Arch. Data ▽ Publish Baseline to Repository ⬟		Publish Final Arch. to Repository ⬟
MODAF Users ◆ User Training – MODAF Principles	◆ Workshop – Determine Arch. Usage Architectural Use Doc ●	◆ Workshop – Bound Arch. Scope ◆ Workshop – Determine Use Cases Plan of Time & Resources ● Architectural Scope Doc ●	◆ Workshop – Establish Data Needs Data Gathering Plan ● Tool Selection ●	◆ Tool-Specific Training Baseline Arch. Review ◆ Baseline Architecture ●	Analysis Review ◆ Initial Analysis ● Final Analysis ●	Finalised Architecture Review ◆ Finalised Architecture ●

In addition to showing the steps that a MODAF user should follow in this example method, the diagram also highlights the key interactions that are required with the MODAF governance processes. Amongst the MODAF governance mechanisms is the Architectural Repository that is run by the System Engineering Integration Group (SEIG)[1]. This can be used to run queries and extract existing architectural data; such as information on the systems that a new capability has to interface with. It is also important that all new architectures are recorded with the appropriate repository to inform others and are available for re-use by other architectures. Furthermore, for the acquisition community the SEIG also provides additional integration services that assist in modelling end-to-end performance and interoperability assurance.

Taking each of the columns from the Diagram above:

Prerequisites

Before commencing a MODAF architecture it is important that the team concerned agree on an approach to creating the MODAF architecture, are familiar with the available views and the

[1] The SEIG were previously known as the Integration Authority (IA)

expected nature of architectural activities associated with their Community Of Interest (COI). Although all of this information is available through this online repository of MODAF guidance and support information, it may be appropriate for the affected team to undertake an introductory course regarding the use of MODAF within their COI. At this point it is probably not appropriate to undertake training regarding the use of any particular MODAF architecture tools as subsequent architectural scoping work may influence the team's final tool selection.

Step 1 – Establish Intended Use

It is essential that any architectural activities are conducted with a clear purpose in mind; the production of a suitable abstraction of complex real world situations that are amenable to detailed analysis. Therefore, step 1 of the architecture development process is aimed at determining and documenting the intended usage of the architecture which can subsequently be used to test whether the developed architecture is fit for purpose. It is often useful to elicit statements of intended use for the architecture through a workshop that includes all of the potential stakeholders who are expected to provide data to and / or utilise the resulting architecture.

Some examples of the "exam questions" that MODAF architectures might address for different COIs include:

- Identification of capability gaps and overlaps – Sponsor[2].
- Develop and trade-off capability options in order to optimise the overall Equipment Programme – Sponsor.
- Develop a clear understanding of the operational context and use cases in support of URD production – Sponsor, Acquisition Integrated Project Team (IPT), Core User[3].
- Establish system boundaries and interfaces, including interoperability analysis – Acquisition IPT.
- Documentation of applied concepts (CONUSE, CONEMP, CONOP) – Concepts and Doctrine organisations.

Step 2 – Define Architecture Scope

The key outcome of this stage is a clear definition of the content and boundaries of the architecture that is to be developed. This will include a definition of the architectural scope in relation to many dimensions, examples of which may include:

- Process scope.
- Organisational scope.
- Systems / platforms scope – including those that have to be interfaced with.
- Geographic scope.
- Coverage of the Defence Lines of Development.
- Timescales that are to be considered (eg 'as-is', 'to-be', 'circa 2015').
- Degree of granularity that is to be modelled (eg system, subsystem or component).

[2] Previously known as "Customer One".

[3] Previously known as "Customer Two".

During this stage the team should also start to consider how the architectural information is likely to be presented so as to address the “exam questions” developed during Step 1. This would normally include a list of the key MODAF views that are expected to be produced.

In some cases modified MODAF views may be desirable in order to enhance the required analysis or presentation of results. For example, modified MODAF views may include the addition of overlays to enhance understanding. However, there is a risk that modified views my not be compatible with other tools / repositories. Therefore, advice should be sought through the SEIG to ensure maximum compatibility.

At this stage it is also important to inform the MODAF governance processes of the intended architectural activities. This will help ensure that architecture developers can be made aware of all extant architectural data sources before they commence work and can also be put in touch with other teams that may be developing architectures with similar or overlapping scopes. As repositories become more densely populated this will considerably ease the burden of developing architectures – whole elements could be cut-and-pasted from extant models.

Step 3 – Develop Data Requirements

Before commencing data gathering in order to populate the architecture, it is good practice to establish a data gathering plan. This should include the definition of what data is required, the level of granularity of data that is required, identification of multiple / redundant data sources to provide data validation and / or back-up sources. The data gathering plan should also consider data formats, pre-processing and data migration issues.

Over time, architectural repositories should become a valuable source of existing architectural data which could be re-utilised with little, if any, translation effort required. This is why it is important to inform the MODAF governance processes of the architecture’s intended scope; to enable a central register of all the MOD’s architectural activities to be built. Based upon this scope information, the repository team(s) can provide a summary of the available architectural data that may be of value to the new architecture.

An important consideration associated with the data gathering plan is conducting an assessment of the security aspects of the populated architecture. This needs to consider not only the classification of the individual data sources, but also the potential for a higher classification if certain combinations / aggregations of lower classification data are presented through the architecture. Consideration should also be made of the security implications for accessing the published architectural data and conducting the required analyses.

Tool Selection

This is probably also the most appropriate stage of the overall process in which to consider tool selection. MODAF does not require a particular tool / suite of tools to be implemented; definitive guidance as to tool availability and fit with different COIs is not available.

Architecting teams should, however, consider the following when selecting a tool / suite of tools:

- Does the tool enable modelling of the architecture at the right level (eg is it modelling at the business level or the technical level? Can it provide the right level of detail?)
- Does / can the tool support the MODAF Meta Model (M3)?
- Can architectural models created in the tool be easily shared with other tools or with the SEIG repository?
- Can the tool exploit existing architectural models?

Note: Although it is intended to set a model interchange standard between a (to be defined) set of tools, there will often be an advantage to edit models within the native format that they were developed in – which maintains the intended graphical layout and potentially additional architectural data that goes above and beyond the MODAF specification.

Having made the tool selection it may be necessary to provide tool-specific training to those who are going to be deeply involved in capturing and editing the architectural models. It is expected that there will be a variety of tool-specific MODAF course available through tool vendors and their intermediaries.

Step 4 – Capture Architecture

It is during this stage of the process that the bulk of the architecture development actually takes place: importing and editing extant architectural models, capturing additional data and entering it into the architecture. This is likely to include extracting data from existing architectures via the SEIG or other repositories.

When building the architecture it is important that it is only constructed in accordance with the MODAF Meta Model and MODAF Taxonomy[4]. These constraints underpin the MODAF tool interoperability mechanisms and compliance with them ensures that the architecture will be compatible with the SEIG and other repositories and that others will be able to re-use the content in the future. Help on how to achieve this will be available through the CIO MODAF team, The Information Coherence Authority for Defence or the SEIG.

It is important that before the resulting architecture is baselined for publication and analysis its accuracy and validity is confirmed. This should include a review of the entire architecture by the subject matter experts who have provided key inputs. It may also be advisable to consult the MODAF governance processes / SEIG during the review process to ensure that any dependent architectures (eg with details of interfacing processes or systems) have not changed or are not in the process of changing.

At this point in the architecture development process the baseline (ie pre-analysis) architecture should be published to an appropriate repository in order to provide visibility to others across the MOD.

In order to facilitate the searching and query of architectures it is essential that the All Views (AV-1 with meta data regarding the architecture and AV-2 with the architecture's object dictionary) are completed thoroughly for all architectures before they are published. It may even be appropriate to start the documentation of the AVs during an earlier stage and to refine them as the scope of the architecture evolves.

Step 5 – Conduct Analyses

Given the validated baseline architecture delivered through step 4 of the process, all of the required data should now be available to conduct the analyses that were identified during step 1. These analyses are likely to be COI-specific, and may include a variety of analytical techniques, including but not limited to:

- Static analysis, such as a gap / overlap analysis against the Strategic Views in order to identify capability issues.
- Dynamic analysis such as network traffic / bandwidth analysis based upon network configurations from SV-1 and traffic data from OV-2 / OV-3.
- Experimentation. Using information developed from the architectural analysis to establish the use cases / context for experimentation campaigns such as those run through NITEworks.
- Trials. Using architectures to provide use case / context information for exercises and trials at a variety of scales from battlelabs to full brigade or division level exercises.

[4] See also the document, "*20090203-Ontologies and their Use in MODAF-U*" elsewhere on this MODAF guidance website.

As with the review of the baseline architecture, it would be good practice to conduct a review on the initial analyses and if necessary to revise the analyses before issuing the final product(s).

Step 6 – Document Results

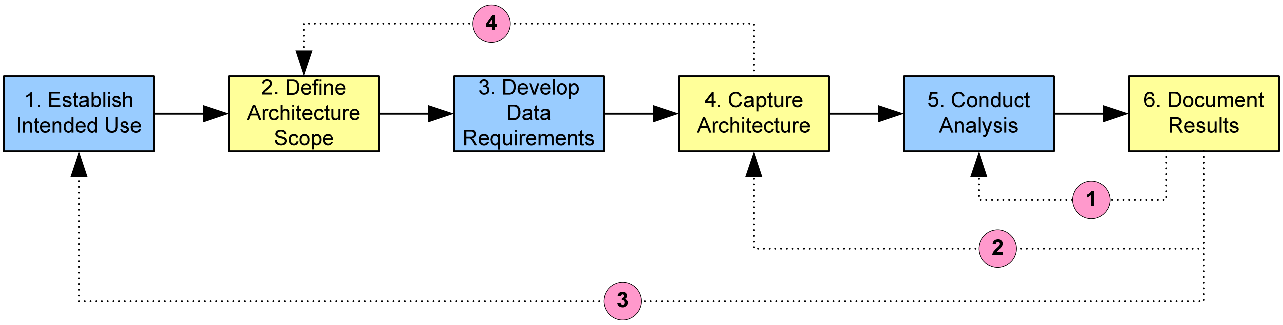
Having conducted the required analyses, changes to the baseline architecture will often be identified. Examples might include:

- Capability analysis may have highlighted a serious capability gap which has been developed into an EP option. The capability, timing and other details of which should then be entered into the finalised architecture
- System interoperability analyses may identify interface problems that have to be rectified by means of changes to the applicable standards or introduction of a gateway equipment, which need to be included in the finalised architecture

When the architecture has been updated with the relevant changes it should again be subjected to a further review and the resulting finalised architecture published to the appropriate repository.

Approaches to Iterative Development

There is no right way of conducting iterations around this general architecting process, but some practical examples are highlighted in this diagram.


1. Establish Intended Use
2. Define Architecture Scope
3. Develop Data Requirements
4. Capture Architecture
5. Conduct Analysis
6. Document Results
1
2
3
4

The first common type of iteration (1) is where having generated the architecture there are periodic analysis / update cycles without any major refresh of the architecture itself. This approach may apply for example to the development and detailing of a number of capability options within the Sponsor’s processes of finalising the Equipment Programme.

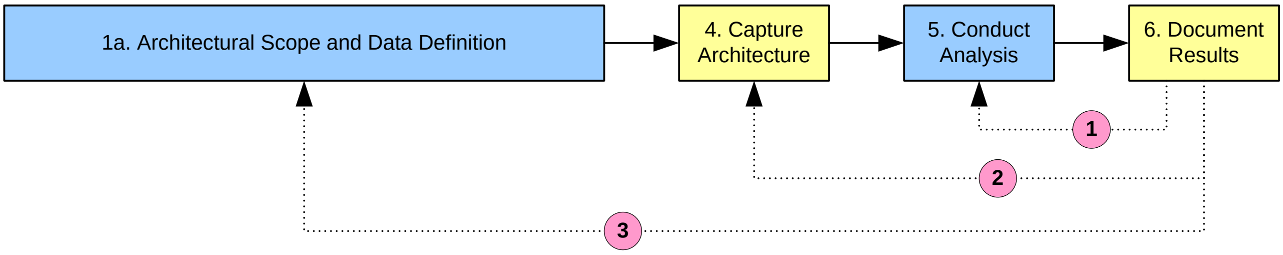
Another type of iteration (2) would be where the architecture is refreshed with more up to date data before the analysis is repeated. This approach may apply for example to the update of the Strategic Views each time the capability audit is conducted within the Sponsor’s processes.

In some cases (3) it may be appropriate to periodically return right back to the start of the architecture processes to review the purpose, scope and data sources. A good example of where this may apply is within an acquisition IPT as it moves between stages in the CADMID / CADMIT cycle; where there are different stage objectives, the solution boundaries may have changed and new data sources may be available. These review activities of the early architectural activities can usually be conducted quite rapidly, possibly covering the review of steps 1 to 3 in a single workshop.

Sometimes, as the data is being gathered and entered into the architecture it may become apparent that it is not going to be possible to achieve the desired results using the elements being considered. In this case (4) it may be necessary to re-visit the architecture scope and / or data gathering plan in order to develop an architecture that will satisfy the original objectives.

Approaches to Rapid Architectural Update

In some cases the team will be working with an architecture that is largely pre-existing (eg from elsewhere within the IA repository) and against a well defined task and scope definition. In these cases it may be possible to abbreviate the process and conduct steps 1 to 3 in a single quick pass through the definition of desired outcomes, architectural scope and data sources as shown here.

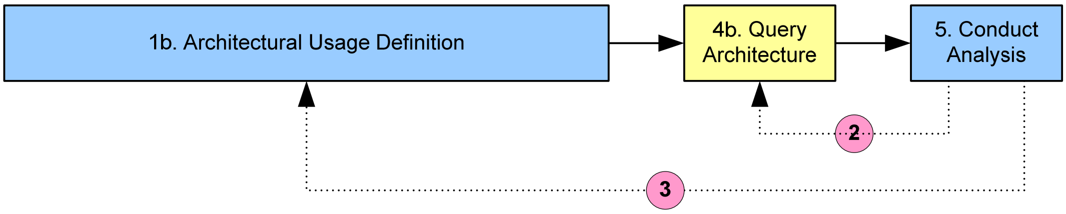


It is still good practice to document the key deliverables of each of these architectural stages even if they are in a single document that has been captured during a single workshop.

It should be noted that similar iterative options could still exist with this rapid update approach.

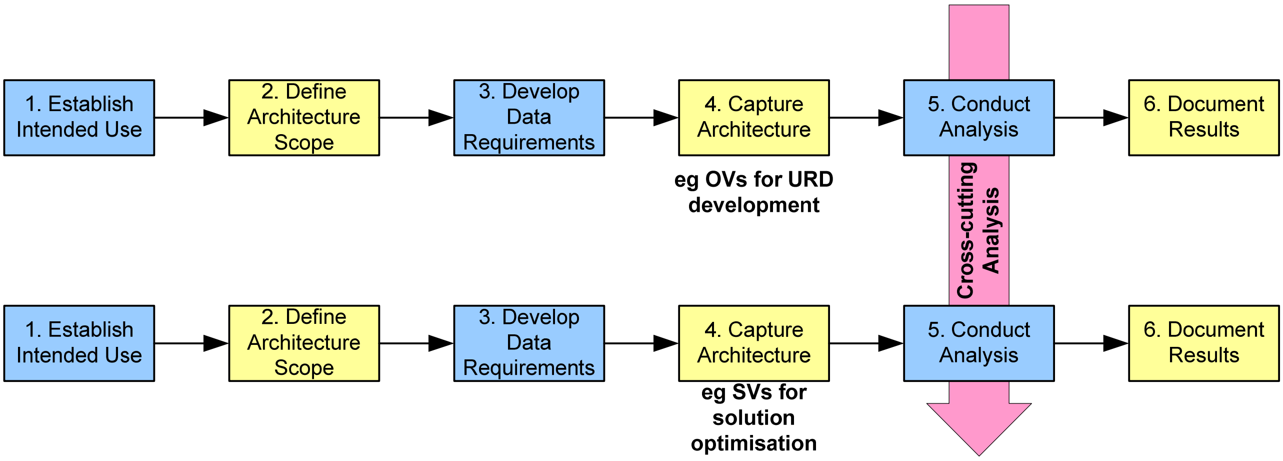
Read-Only Architectural Usage

In some cases particular groups of MOD architecture users will not need to create architectures of their own but will be conducting analysis on the architectures produced by others. For instance, this may apply to the assurance and scrutiny communities who want to examine the adequacy and maturity of architectural activities conducted by IPTs at various stages of the acquisition lifecycle. In this case, a rather abbreviated version of the six-step process may apply; there will be no update or publication of the architecture, as shown in this diagram.



Parallel Architectural Activities

Another common situation in the MOD will be where there are a number of parallel streams of architectural activities being conducted in relation to the same overall project. For example, within the concept stage of the acquisition cycle there will be refinement activities on the User Requirement Document (URD) being conducted largely using the OV suite of MODAF views while simultaneously a high level suite of SVs will be in the process of being developed for the purpose of optimising different system solutions. In some cases these parallel streams of architectural activity may be being conducted by quite separate teams. However, in most cases these various architectural streams will need to converge at certain points in the project when joint / cross-cutting analyses are required (see the diagram below), such as an IPT conducting an overall risk assessment using elements from both the OVs and SVs to assess issues such as the clarity of use cases and the degree of interface definition.


1. Establish Intended Use
2. Define Architecture Scope
3. Develop Data Requirements
4. Capture Architecture
5. Conduct Analysis
6. Document Results
eg OVs for URD development
Cross-cutting Analysis
1. Establish Intended Use
2. Define Architecture Scope
3. Develop Data Requirements
4. Capture Architecture
5. Conduct Analysis
6. Document Results
eg SVs for solution optimisation