

Step 11: Define Detailed Solution Architecture

Version 1.0, February 2005

1 Step Description/Objectives:

Upon acceptance of the business case (OMB Exhibit 300) or acceptance and execution of the recommended O&M funding strategy, the solution architecture may be defined and implemented. The Step consists of the formation of an integrated product team, the definition of a detailed solution architecture, the validation of the defined architecture against enterprise architecture principles, best practices, and conceptual architectures, concurrence on the architecture by the core team, and finally acceptance of the proposed architecture by the IBAT/ARB prior to implementation. This Step description document establishes the standard set of work products approved for creation on all new application and reengineering projects.

While this Step sets forth an approach and associated work products for the creation of a solution architecture, it's equally applicable to the specification, implementation, and execution of service components within the context of a service oriented architecture (SOA).

Successful service-oriented, component-based architectures will ensure that information technology (IT) capabilities are managed in a structured enterprise-wide technology and business process lifecycle. The DOI's component-based architecture will focus reuse efforts on large business level components in a collaborative environment that includes system owners, capital planners and enterprise architecture.

In support of SOA, the DOI has adopted the FEA Service Component Reference Model (SRM) for use in discovering DOI-wide business and application service components in IT investments and assets. The SRM, as implemented by the DOI, is a service component-based framework that provides a "leverage-able" foundation to support the reuse of business services, service components, and federated business systems - independent of business function and technology.

Fundamentally, the role of enterprise architecture in detailed system design is that of oversight and support. The solution architecture team will facilitate the transfer of BPR models, functional requirements, and enterprise architecture guidance to the development team. The solution architecture team will also provide recommendations and comments on the final design prior to construction and execution. The following table¹ expands on the notion of component granularity to include ROI.

Level of Granularity	Definition	ROI
Federated Business Component	A set of cooperating system-level components federated to resolve the business need of multiple end users often belonging to different organizations. Can be expressed as an IT 300 exhibit or a federation of IT 300 exhibits.	High
Business Component System	A set of cooperating business components assembled together to deliver a solution to a business problem. Can be expressed as an IT 300 exhibit.	High
Business Component	Represents the implementation of an autonomous business concept, business service, or business process. It consists of all the technology elements (i.e., software, hardware, data) necessary to express, implement, and deploy a given business concept as an autonomous, reusable element of a large information system. It is a unifying concept across the development lifecycle and the distribution tiers. Normally not expressed as an IT 300 exhibit, but as a sub-component of a larger business component system.	Medium
Distributed Component	The lowest level of component granularity. It is a software element that can be called at run-time with a clear interface and a clear separation between interface and implementation. It is autonomously deployable. Normally not expressed as an	Low

¹ Taken from Service Component-Based Architectures Version 2.0, FEAPMO, June 2004.

Step 11 Guidance: Define Detailed Solution Architecture

	IT 300 exhibit. A distributed component provides low ROI for capital planning purposes.	
Language Class	A class in an object-oriented programming language to build distributed components. This is NOT considered an SRM component. Normally not expressed as an IT 300 exhibit. A language class provides very low ROI for capital planning purposes.	Very Low

To be effective, enterprise architecture should focus at a higher level of service component granularity. Proper system design documentation and notation will facilitate effective identification, assembly, and usage of service components across the enterprise. Service component aggregation will also facilitate rapid building and implementation of system components in support of a given initiative or line of business. Currently the DOI EA Repository (DEAR) is configured to model and support component granularity from federated business components down to the level of distributed components. The DEAR tool itself, Popkin Systems Architect (SA), is capable of support component granularity down to object-oriented language classes.

2 Step Purpose:

Step 11 is performed for one or more of the following purposes:

- To define the detailed solution architecture
- To ensure that the solution architecture is in compliance with DOI enterprise architecture principles, best practices, and conceptual target application architectures

Impact of Not Performing this Step

If Step 11 is not performed:

- There is increased implementation risk due to ill-defined requirements that may result in the implementation of a system that does not conform to set enterprise architecture standards
 - There is a lessened likelihood of cost savings due to reusable components, shared infrastructure, and leveraged assets
-

3 Tasks to Perform Step:

At a minimum, Step 11 starts with the following inputs:

Inputs:

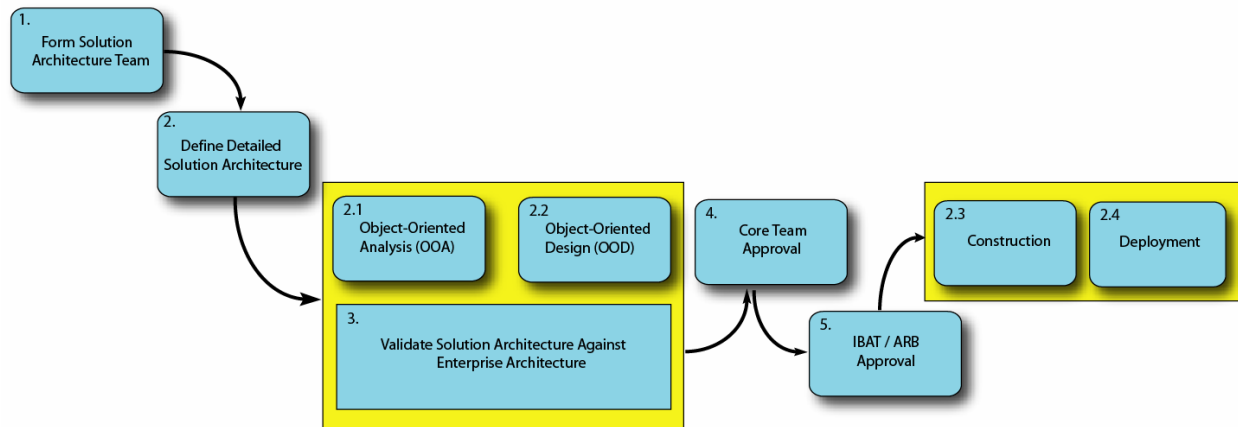
- Business Focus Area (BFA) Conceptual Data Model
- BFA Integrated Services and Systems Model
- BFA Data Standards Document
- BFA Recommendations and Findings for Process Improvement Document from Step 8 – Conduct Business Process Reengineering (BPR)
- BFA Exhibit 300 (if applicable)
- BFA-related documents such as requirements and normalized processes from Step 8 – Conduct Business Process Reengineering (BPR)

Detail:

Within this step, there are five major tasks, each with associated sub-activities:

Note: Task 3, “Validate solution architecture against the enterprise architecture”, should be viewed as an oversight and coordination task between the Integrated Project Team (IPT) and the Core Team across the tasks that make up the definition of the detailed solution architecture.

Step 11 Guidance: Define Detailed Solution Architecture



Task 1: Form Solution Architecture Team.

Activities:

1. Form Solution Architecture Team(s) with representatives from sponsor business areas.

Task Outputs:

- Solution Architecture Team Roster Document

Task 2: Define Detailed Solution Architecture. This task and associated work products are applicable to new business applications and existing applications developed in object-oriented languages. To establish the context for the proposed standardized work products, the following is a brief summary of major design phases within an object-oriented methodology.

Activities:

1. Conduct Object-Oriented Analysis (OOA).

Object-oriented analysis (OOA) is concerned with developing software engineering requirements and specifications that are expressed in a system's object model (which is composed of a population of interacting objects), as opposed to traditional data or functional views of systems. OOA can yield the following benefits: maintainability through simplified mapping to the real world, which provides for less analysis effort, less complexity in system design, and easier verification by the user; reusability of the analysis artifacts which saves time and cost; and depending on the analysis method and programming language, productivity gains through direct mapping to features of Object-Oriented Programming Languages. Some aspects of the OOA may have been developed as part of the BPR Task. The foundations of OOA, in any event, will have been laid by the BPR Task and the BPR Task outcomes will form the foundation of OOA use cases and functional requirements. Most standard OOA work products are supported by the DEAR modeling tool, Popkin System Architect. In cases where the party responsible for a given WP may not use Popkin SA, third-party conversion tools such as Meta Integration Bridge have been shown to allow the sharing model artifacts between Popkin SA and other vendor tools such as Rational XDE.

The following work products are associated with the OOA step of Task 2:

OOA WP Name	WP Description	Tool	Responsible	Usage

Step 11 Guidance: Define Detailed Solution Architecture

OOA WP Name	WP Description	Tool	Responsible	Usage
User Requirements	A description of the business requirements for an application, including the capabilities and features it must support. These requirements must be from a business perspective.	MS-Word	Solution Architecture Team	Reqd
System Context Diagram	A graphical depiction, supported by text, that shows the desired system as a single component or process and identifies the interfaces between the system and external entities.	Popkin SA (Project)	Solution Architecture Team	Reqd
Use Case Scenarios	A text description of the most prevalent course of events that a user performs in his or her work. It gives the facts of a real situation a user has to deal with and gives details about how a particular Step is conducted.	MS-Word	Developer	Reqd
Use Case Diagrams	Graphical depiction of the functional requirements of the system. These are developed by the developer from the use case text scenarios / narratives and reviewed by the solution architecture IPT.	Popkin SA (Project)	Developer	Reqd
Use Case Narratives	Textual descriptions of how the system must behave from a user's point of view; they do not describe how the system works internally or its internal structure or mechanisms; they do not describe any aspects of the user interface implementation itself.	MS-Word	Developer	Reqd
Nonfunctional Requirements	A specification of the qualitative and other nonfunctional requirements or conditions that the system must satisfy.	MS-Word or MS-Excel	Solution Architecture Team	Reqd
Application Security Group Structure Requirements	A description of the required security structure and data regarding users, groups, rights, etc.	MS-Word	Developer	Reqd
Analysis Object Model	A Class Diagram focused on defining the problem domain by understanding <u>what</u> aspects of a business model are to be included in the system, not <u>how</u> .	Popkin SA (Project)	Developer	Reqd
Class Descriptions	A brief textual definition for each core business model Class.	MS-Word or MS-Excel	Developer	Reqd
User Interface Mockups	A graphical depiction of the user interface screens, reflecting the data presented, style, and navigational options.	MS-Word	Developer	Reqd
Data Elements	A depiction of data element definitions, data types and sizes, domains, standard DOI business names, and other metadata about the data elements and entities.	DEAR	Developer	Reqd
Architectural Decisions	A record of important decisions about any aspect of the architecture including the structure of the system, the provision and allocation of function, and adherence to standards. Although this work product is initially created during Analysis and maintained during Design, all Architectural Decisions must be resolved by the end of Design.	MS-Word or MS-Excel	Developer	Reqd
Requirements Matrix	A tracking matrix which details all the requirements of the system, including their type, priority, status, etc.	MS-Word or MS-Excel	Developer	Reqd
Test Strategy	A high-level description of major testing Tasks and approach to be used to ensure that the critical attributes of the system are tested adequately.	MS-Word	Developer	Reqd
Glossary	A definition of acronyms and terms encountered during the project.	MS-Word or MS-Excel	Developer	Optional

2. Develop Object-Oriented Design (OOD).

Step 11 Guidance: Define Detailed Solution Architecture

Object-oriented design (OOD) is concerned with developing an object-oriented model of a software system to implement the identified requirements from the Object-Oriented Analysis (OOA). OOD can yield the following benefits: maintainability through simplified mapping to the problem domain, which provides for less analysis effort, less complexity in system design, and easier verification by the user; reusability of the design artifacts, which saves time and costs; and productivity gains through direct mapping to features of Object-Oriented Programming Languages. The vast majority of these work products will be maintained in a project-level encyclopedia (Popkin SA) separate from the enterprise DOI repository (DEAR).

The following work products are associated with the OOD step of Task 2:

OOD WP Name	WP Description	Tool	Responsible	Usage
Design Object Model	A class Diagram focused on how a system will be constructed and implemented.	Popkin SA	Developer	Reqd
Service Descriptions	A collection of information that defines the purpose of a service, along with detail on how to interact with it.	MS-Word	Developer	Reqd
Conceptual Data Model	A conceptual model of the data, business definitions and relationships for all internal and external system data components. This will include sharing of common data resources and common enterprise data subject areas ensure data consistency between DOI applications.	DEAR	Developer	Reqd
Sequence Diagrams (Business, User Interface [UI] and Persistence versions)	Graphical representations of the internal behavior of the system; they show how objects collaborate by sending messages and returning responses to each other.	Popkin SA	Developer	Reqd
State Transition Diagrams	Graphical depictions of the behavior or lifecycle of a single class; they describe states that may be attained and how receipt of external stimuli (events, messages, responses) cause a change in state. Typically only created for classes with significant dynamic behavior.	Popkin SA	Developer	Reqd
View Navigation Flow Diagram	A graphical representation of the flow of control among the Application Interface components of the system.	Popkin SA	Developer	Reqd
UI Specifications & Navigational Detail	A description of all Application Interface components (this is applicable to all types of Application Clients [including, but not limited to, those created with JSPs, HTML, Java Swing, etc.] and Application Agents [servlets]) including a depiction of screen appearance, input fields, session data usage, exception handling, and navigational destinations.	MS-Word	Developer	Reqd
Enterprise Information and Data Stores Definition	A graphical and text representation of the major data sources used by the system. For each source, it includes a description plus detail on the characteristics of the source, such as transactional requirements, performance/availability, sharing/locking, security, etc.	MS-Word	Developer	Reqd
Physical Data Model	A representation of the physical organization of data in a graphic format. For the relational database, the physical data model defines primary and foreign keys and constraints.	Popkin SA	Developer	Reqd
Persistence Design Specifications	A description of how all of the persistence services will be fulfilled for the system. For each service, it contains detail on the Persistence Contract (input), associated Persistence Object, return/result (Persistence Object Façade), data source specifics such as table/columns or file/field locations, and custom query info.	MS-Word	Developer	Reqd
Operational Model	A representation of a network of computer systems, their associated peripherals and the	Popkin SA	Developer	Reqd

Step 11 Guidance: Define Detailed Solution Architecture

OOD WP Name	WP Description	Tool	Responsible	Usage
	systems software, middleware, and application software that they run; it is the physical view of the system.			
Component Model	A high-level representation of the entire software hierarchy that describes the responsibilities, interfaces, and collaboration of components.	Popkin SA	Developer	Reqd
Application Resource Descriptions	A description of resources (property files, etc.) used by an application, including their purpose, contents, and physical (file) names.	MS-Word	Developer	Reqd
Detailed Test Plan	A description created for each level of testing identified as necessary in the Test Strategy. Each plan identifies the scope of testing for that level, functions/features to be tested, the testing Steps to be performed, and the personnel responsible for each Step.	MS-Word & MS-Project	Developer	Reqd
Acceptance Test Plan	A description of the steps the customer will use to verify that the constructed system satisfies the stated requirements. The plan describes clearly and fully how to set up the test and the acceptable system behaviors in response to the test.	MS-Word & MS-Project	Developer	Reqd
Deployment Plan	A description of the organization, strategy, resources, and methods used to deploy the new application at customer sites. It provides a detailed schedule of events, expected project duration, persons responsible, and event dependencies required to ensure successful cut-over to the new system.	MS-Word & MS-Project	Developer	Reqd

3. Begin Construction.

Construction is focused on implementing the system design via source code and other types of components. Object-oriented programming languages (OOPLs) are the natural choice for source code implementation of an Object-Oriented Design because they directly support the object notions of classes, inheritance, information hiding, and dynamic binding. Testing of individual components is performed to ensure that the components operate as expected. Many of the tools associated with the construction step will be project specific. At the time this document was created, no standard application development and deployment environment existed across the DOI. Construction work products are included in this method paper for completeness, but actual system construction would not take place until **after** Step 11.

The following work products are associated with the OOC step of Task 2:

Construction WP	WP Description	Tool	Responsible	Usage
Build Procedures	A description of how to generate an executable copy of the system. This may include such items as which files are required, which links must be in place, which libraries are accessed, and the sequence of steps required to generate the system, etc.	MS-Word	Developer	Reqd
Source Code	The actual implementation of the system design in the chosen programming language(s).	Project-Dependent	Developer	Reqd
Deployable Components	The primary software deliverables for a system	Project-Dependent	Developer	Reqd
Test Specification	A set of documents that define and describe the actual test architecture	MS-Word	Developer	Reqd
Operations Guide	A depiction of the processes required in order to support the system in the production environment. Includes details regarding normal operation	MS-Word	Developer	Reqd
Help Manual	A description of the system's functions	MS-Word	Developer	Reqd
Backup and Recovery	A description for backing up the application data	MS-Word	Developer	Reqd

Step 11 Guidance: Define Detailed Solution Architecture

Construction WP	WP Description	Tool	Responsible	Usage
Plan				

4. Deploy.

Deployment is focused on validating that the system is functionally correct and operationally sound, preparing the operational environment, and implementing the system into the production environment. Deployment work products are included in this method paper for completeness, but actual system deployment would not take place until after Step 11.

The following work products are associated with the Deployment step of Task 2:

Construction WP	WP Description	Tool	Responsible	Usage
Test Results	A description of the outcomes of a specific test execution, and comparison of the actual test results to the expected results from the test case. It identifies and documents any variances that indicate further analysis is required..	MS-Word	Developer	Reqd

Task 3: Validate Solution Architecture Against Enterprise Architecture. This task is to ensure oversight over the object-oriented analysis and design steps that are part of the “Define Detailed Solution Architecture” task (Task 2 above). Validation needs to continue after the Core Team and Business Architecture Team (IBAT)/Architecture Review Board (ARB) approvals of the detailed solution architecture on to the construction and deployment steps of Task 2.

Task Outputs:

- Systems Development Compliance Document / Matrix

Task 4: Submit Solution Architecture for Core Team Approval. After the performance of analysis and design under the oversight of the solution IPT, the solution is submitted to the Core Team for review and approval.

Task Outputs:

- none

Task 5: Submit Solution Architecture for IBAT/ARB Review. After Core Team approval, the detailed design is submitted to the Interior Business Architecture Team (IBAT)/Architecture Review Board (ARB) for review and approval. If approval is granted, system construction and deployment occurs.

Task Outputs:

- Recommendations and Findings Presentation

4 Step Participants: (Use Bullets)

- **Core Team Representative:** Person,Org
- **Chief Technology Officer:** Person,Org
- **Interviewer:** Person,Org
- **Note Taker:** Person,Org
- **Stakeholder:** Person,Org
- **Other:** Person,Org

5 Dependencies:

TBD.

6 Step Deliverables:

Solution Architecture Team Roster Document: The Solution Architecture Team Roster document will be created as part of the Define Detailed Solution Architecture Step and updated as needed. It will document core team membership and contact information. It will contain Project Sponsor, Project Proponent, Core Subject Matter Experts, and other Teams or groups the project will need to work with or interview during the project.

- **TEMPLATE :** Hyperlink to Template (**Deliverable Type: Put Type Here**)
- **SAMPLE :** Hyperlink to Sample (**Deliverable Type: Put Type Here**)

User Requirements Document: The User Requirements Document contains a description of the business requirements for an application, including the capabilities and features it must support. These requirements must be from a business perspective.

- **TEMPLATE :** Hyperlink to Template (**Deliverable Type: Put Type Here**)
- **SAMPLE :** Hyperlink to Sample (**Deliverable Type: Put Type Here**)

System Context Diagram: A graphical depiction, supported by text, that shows the desired system as a single component or process and identifies the interfaces between the system and external entities.

- **TEMPLATE :** Hyperlink to Template (**Deliverable Type: Put Type Here**)
- **SAMPLE :** Hyperlink to Sample (**Deliverable Type: Put Type Here**)

Nonfunctional Requirements Document: A specification of the qualitative and other nonfunctional requirements or conditions that the system must satisfy.

- **TEMPLATE :** Hyperlink to Template (**Deliverable Type: Put Type Here**)
- **SAMPLE :** Hyperlink to Sample (**Deliverable Type: Put Type Here**)

Systems Development Compliance Document / Matrix: A list of required and optional systems analysis, design, construction, and deployment documents. The Solution Architecture Team will verify the creation, the format, and location of required documents and ensure source models from DEAR are made available to the Core Team as required.

- **TEMPLATE :** Hyperlink to Template (**Deliverable Type: Put Type Here**)

Step 11 Guidance: Define Detailed Solution Architecture

- **SAMPLE** : Hyperlink to Sample (**Deliverable Type: Put Type Here**)

Recommendations and Findings Presentation: The final system design recommendations and findings will be documented and delivered as a formal presentation to the IBAT/ARB.

- **TEMPLATE** : Hyperlink to Template (**Deliverable Type: Put Type Here**)
- **SAMPLE** : Hyperlink to Sample (**Deliverable Type: Put Type Here**)

7 Communications Considerations:

TBD.

8 References:

1. Mayer, Richard J., Dewitte, Paula S., "Delivering Results: Evolving BPR from art to engineering", 1998
2. Rummler, G.A., Brache, A.P., "Improving Performance: How to Manage the White Space on the Organizational Chart, Wiley, 1995