IBM Solution Design Method

Introduction

IBM Solution Design Method Introduction:

- **Objectives**
- **Introduction**
- **Unified Method Architecture**

Goals and Objectives

Goals:

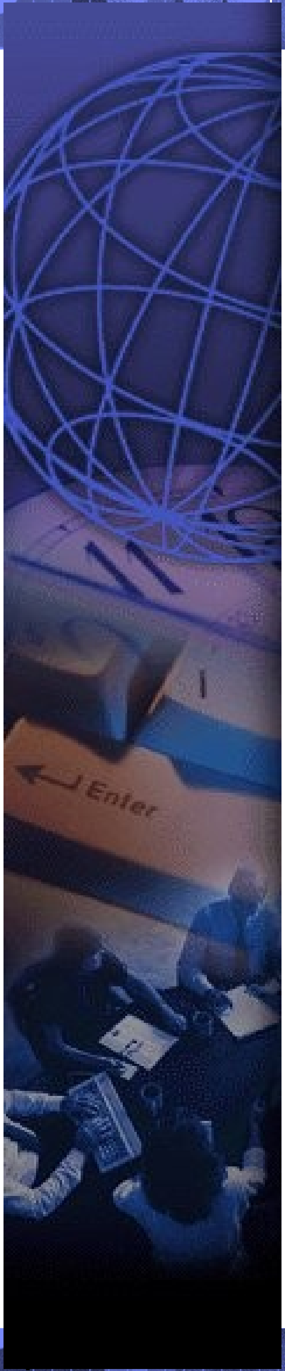
- Disciplined approach to solution design process:
  - Helps you break a large project into manageable 'chunks'
  - Better technical collaboration between IBM and Business Partners
  - Helps you remember where you left off with a customer

Objectives:

- Introduction of a “common technical language”
  - When IBM Architect support in a project engagement
  - When IBM Architect and SMEs are reviewing architecture

- Limited to solution design part of the project

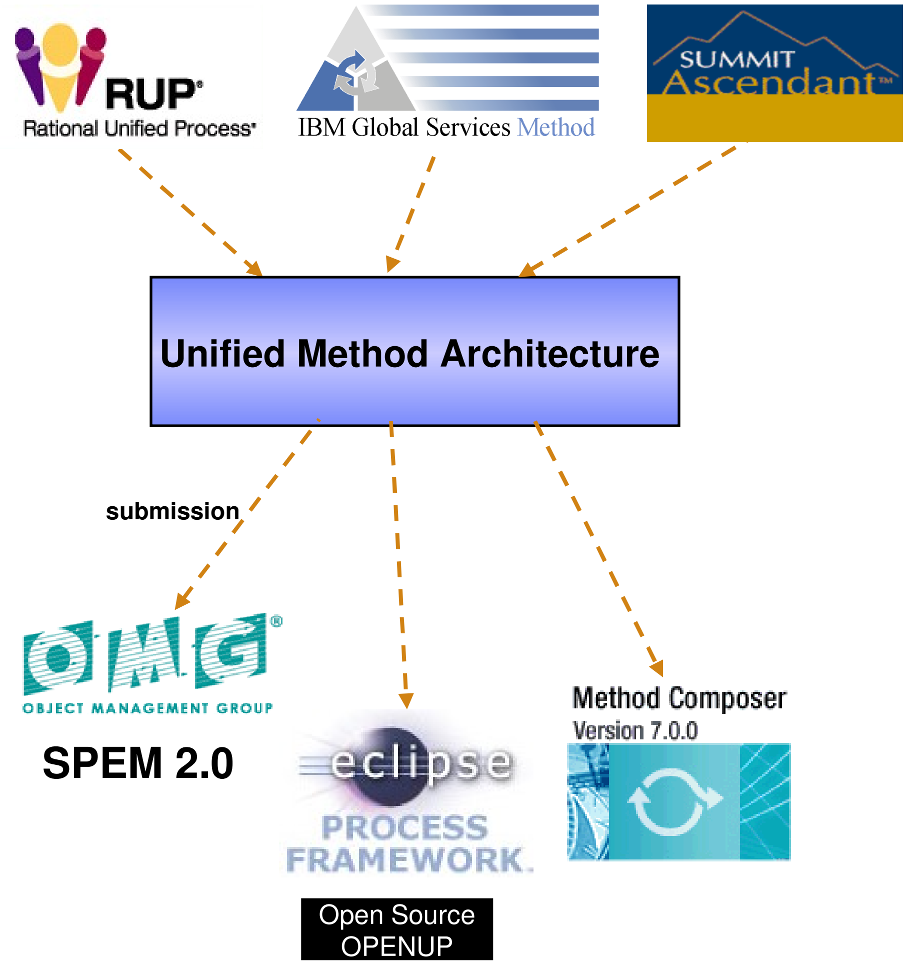
- Using Industry standards (UML, ADS)



Design Principles

1. IBM Solution Design Method Activities drive design process. Work products should not drive process.
2. Work product input should be captured when you get it -- you rarely control timing.
3. Each work product has many inputs and is input to many others. Only primary inputs shown in teamroom.
4. Each work product goes through multiple elaborations -- by you or others before and after the solution is "sold".
5. Some work products have multiple views (e.g., AOD), some are for a specific audience (e.g., Component Model for developer).
6. IBM Solution Design Method allows trace-ability of each decision back to requirements.
7. IBM Solution Design Method encourages the reuse of assets where possible.
8. IBM Solution Design Method adds value/framework for partial or brief activities (or part of the solution for specialists).
9. IBM Solution Design Method will continue to evolve to meet our needs.

IBM Unified Method Architecture (UMA)



Unified Method Architecture

submission

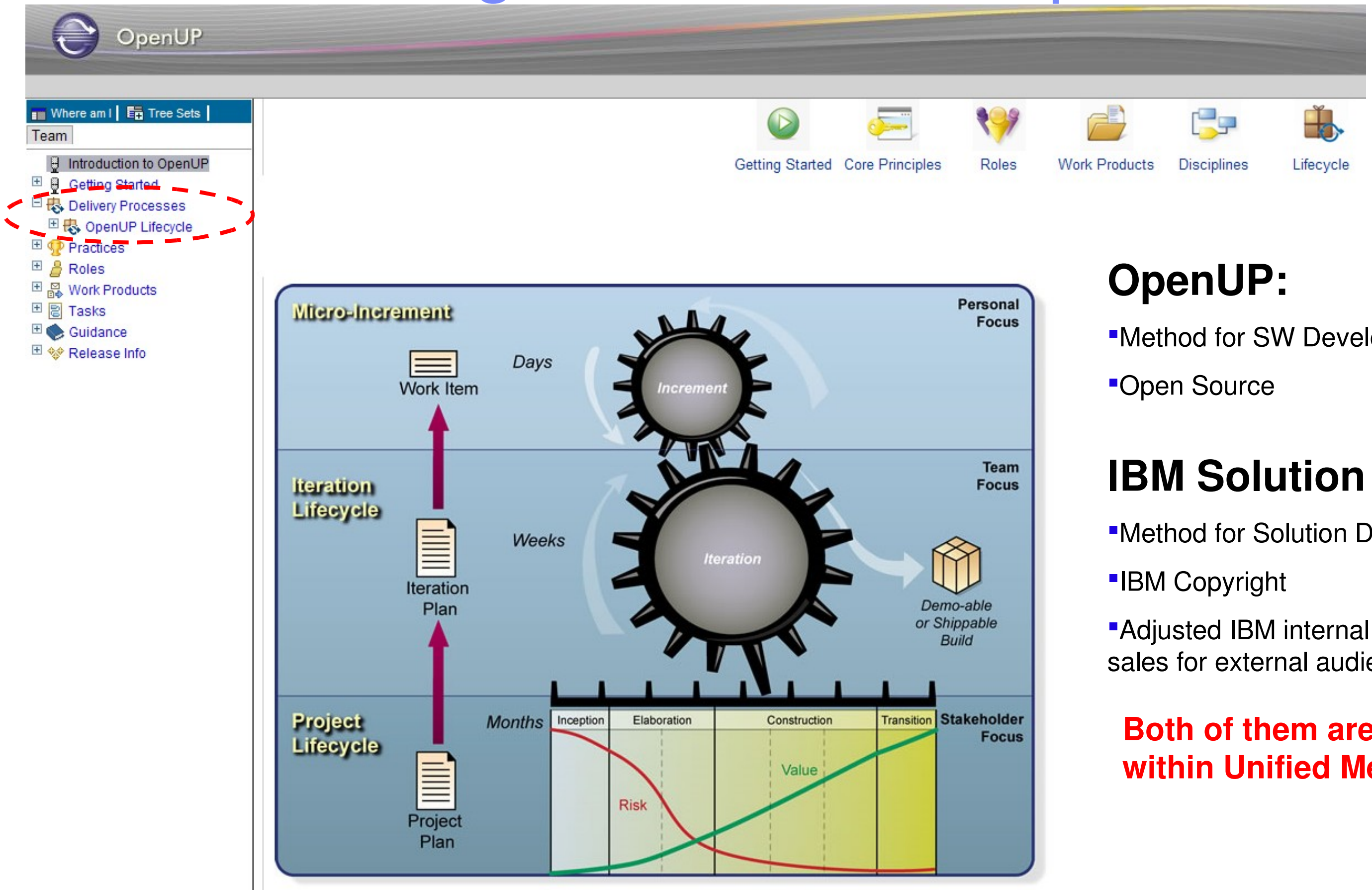
SPEM 2.0

Open Source
OPENUP

Method Composer
Version 7.0.0

- IBM UMA: Unified Method Architecture
  - UML Meta-model Specification (provides one IBM-wide method structure and terminology)
- Developed by interdisciplinary team with members from all three Methods
- Provides one integrated Method Engineering Solution: Prepares for common management and structural integration of all of IBM's method offerings
- Submitted to OMG to become Software Process Engineering Meta model (SPEM) 2.0 standard – Accomplished 2007

IBM Solution Design Method and OpenUP



OpenUP:

- Method for SW Development
- Open Source

IBM Solution Design Method:

- Method for Solution Design
- IBM Copyright
- Adjusted IBM internal method for technical pre-sales for external audience

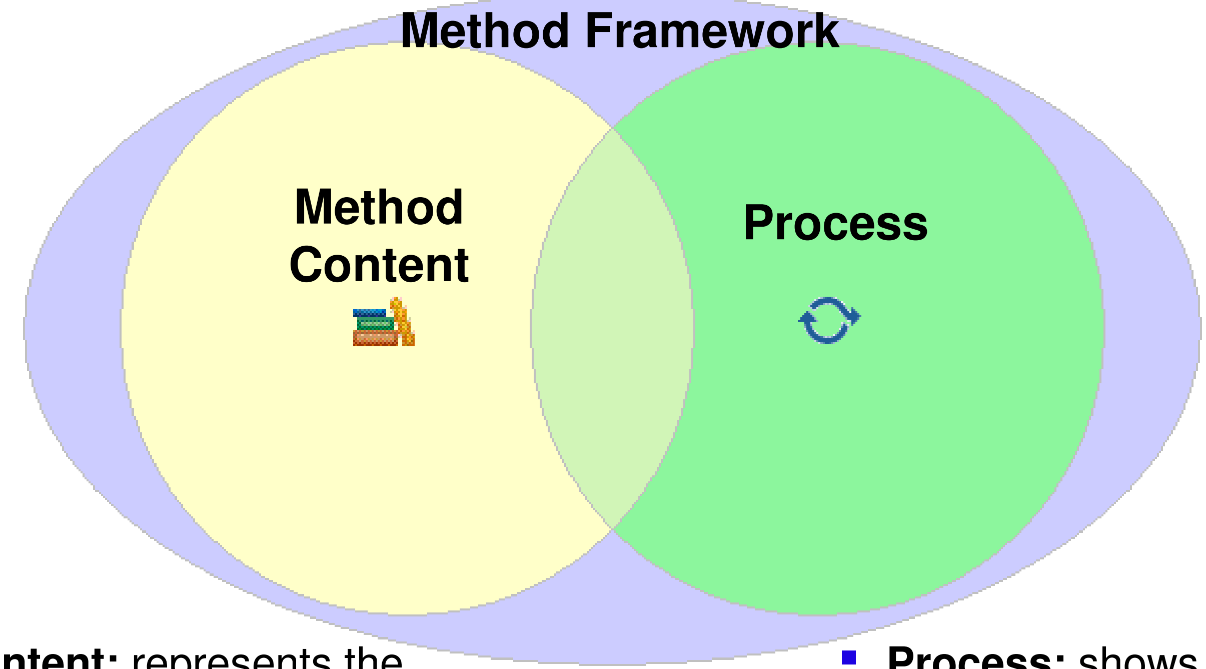
Both of them are Delivery Processes within Unified Method Architecture

What is OpenUP?

OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle. OpenUP embraces a pragmatic, agile philosophy that focuses on the collaborative nature of software development. It is a tools-agnostic, low-ceremony process that can be extended to address a broad variety of project types

Method content elements separated from process elements

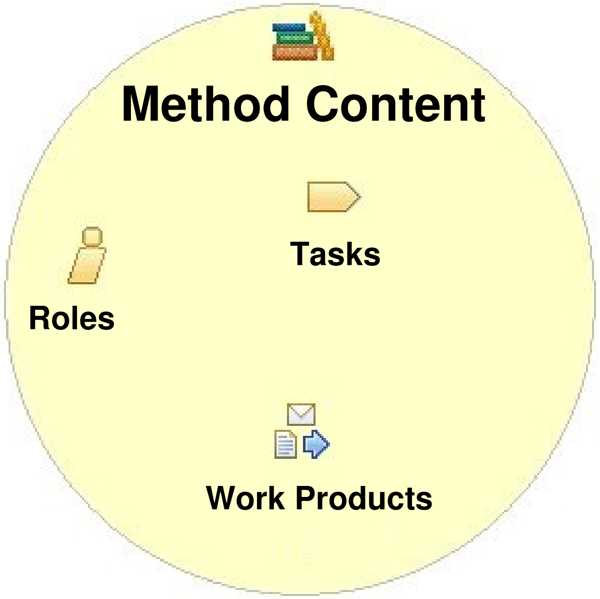
- **Method:** represents a consistent and repeatable approach to accomplishing a set of objectives based on a collection of well-defined techniques and best practices



- **Method Content:** represents the primary reusable building blocks of the method that exist outside of any predefined lifecycle

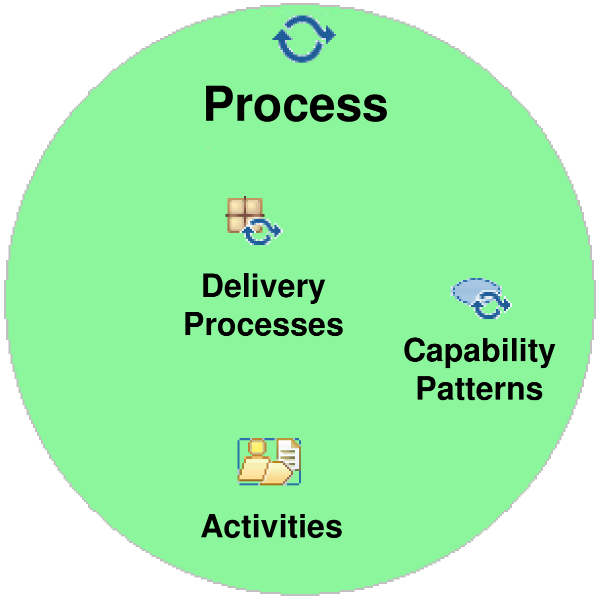
- **Process:** shows the assembly method content into a sequence or workflow (represented by a work breakdown structure) used to organize the project and develop a solution

Method content is made up of: roles, work products and tasks



- ***Roles:*** describe the skills necessary to perform a task or create a work product
- ***Tasks:*** provides guidance on the work that needs to be done to transform inputs into outputs through a series of steps performed by one or more roles (independent of a WBS)
- ***Work Products:*** define the items needed as input or created as output of one or more tasks that are typically the responsibility of a single role
  - **Artifacts**: provide **guidance** on work products that represent tangible items that may have examples or a predefined template and serve as basis for reuse
  - **Outcomes:** provide **guidance** on work products that are intangible items used to show the completion of some set of activities or a result that does not represent a harvestable asset (i.e. trained students, configured system, installed software, etc.)
  - **Deliverables:** describe the primary **outputs that represent value** to the client, customer or other stakeholders and are typically the resulting of packaging other work products for sign-off and delivery

Process is made up of delivery processes, capability patterns and activities



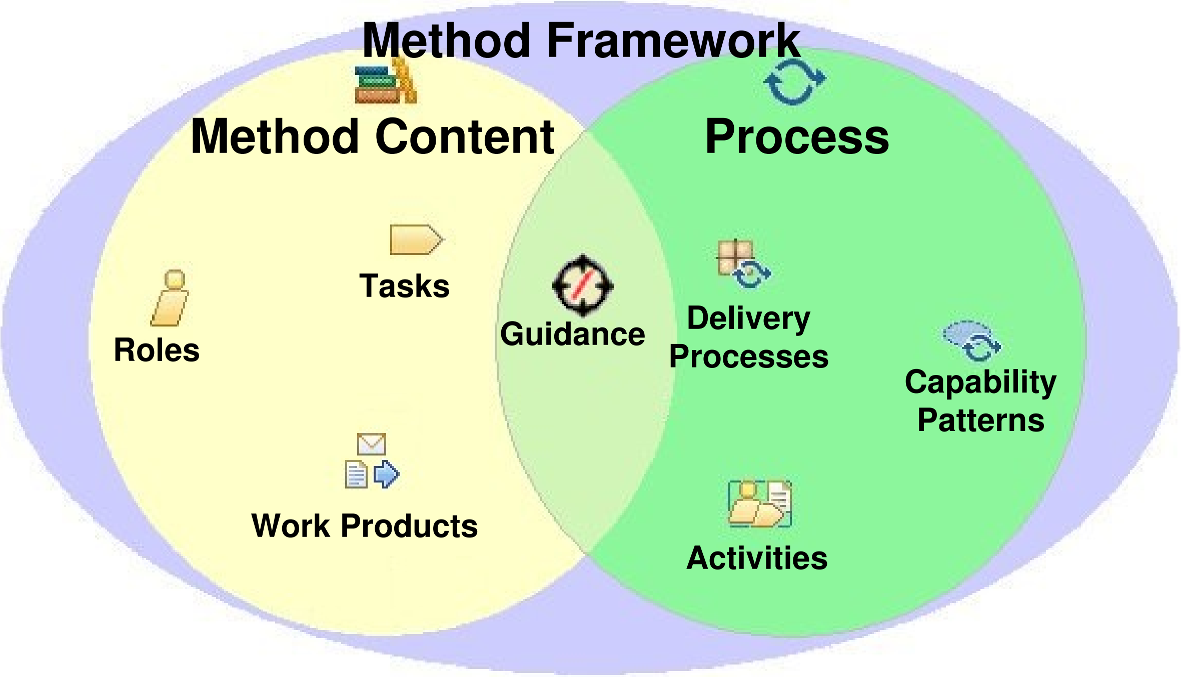
- **Delivery Processes:** are used to define a complete integrated approach to specific type of project
- **Capability Patterns:** are a special type of process used to define a stereotypical way of performing work related to a particular subject which may be used as a courser grained building block for assembling delivery processes
- **Activities:** are used to support the nesting and logical grouping of related breakdown elements

Phases: define specialized activities usually based on contractual milestones, major deliverables or decision checkpoints

Iterations: are a specialized type of activity used to describe repeating items within a WBS

Milestones: describe breakdown elements used to represent a significant event in a project

Guidance applies to both method content and process



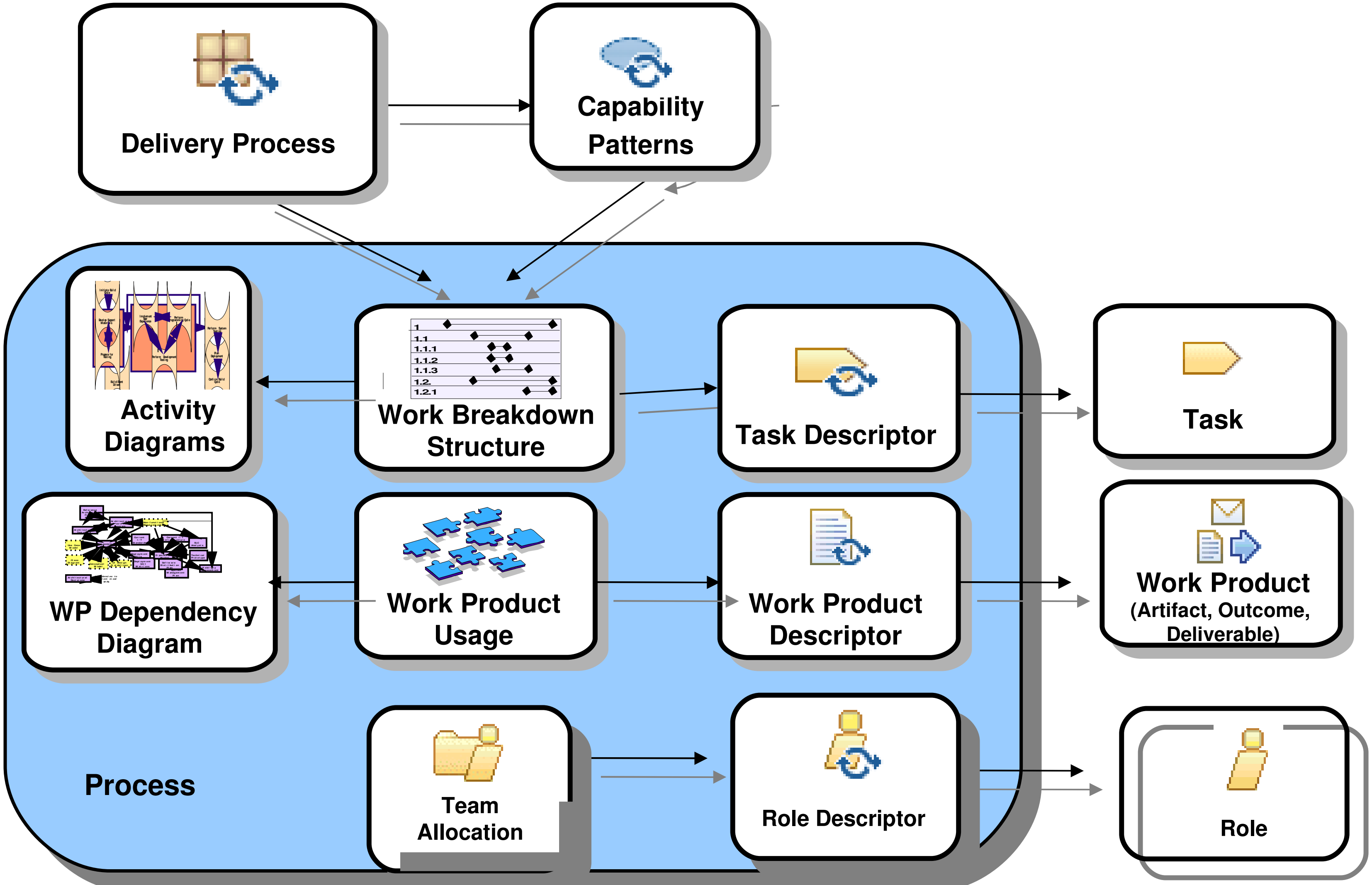
Packaging constructs facilitate the authoring and management of content

- **Library:** overall workspace/repository for developing methods consisting of plug-ins and configurations
- **Configuration:** selection of packages and plug-ins with associated views used for publishing methods
- **Plug-in:** represents a physical container for packages, primary means of relating dependent content
- ***Method Packages:*** allow for the grouping of elements to facilitate configuration management, governance, distribution and tailoring. Packages can be nested as well as be dependent on other packages.

  Content Package (method content)

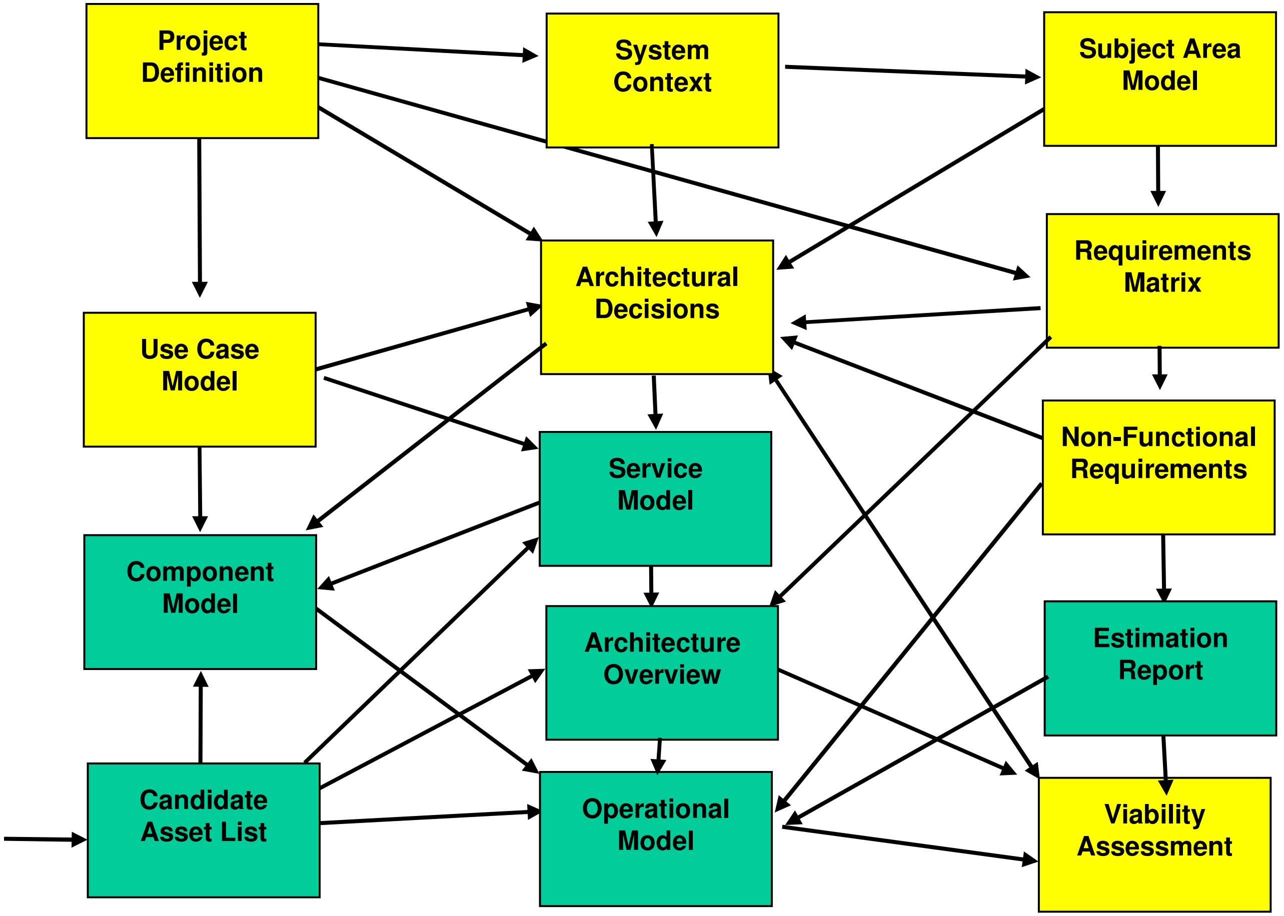
  Process Package (process)

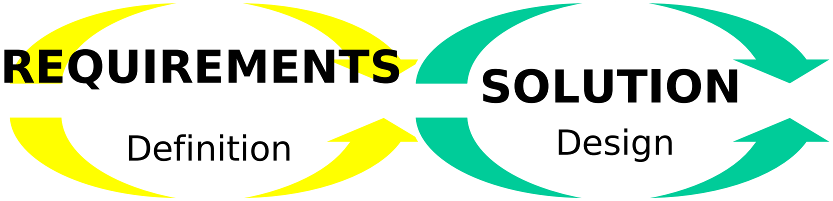
Mapping process to UMA content



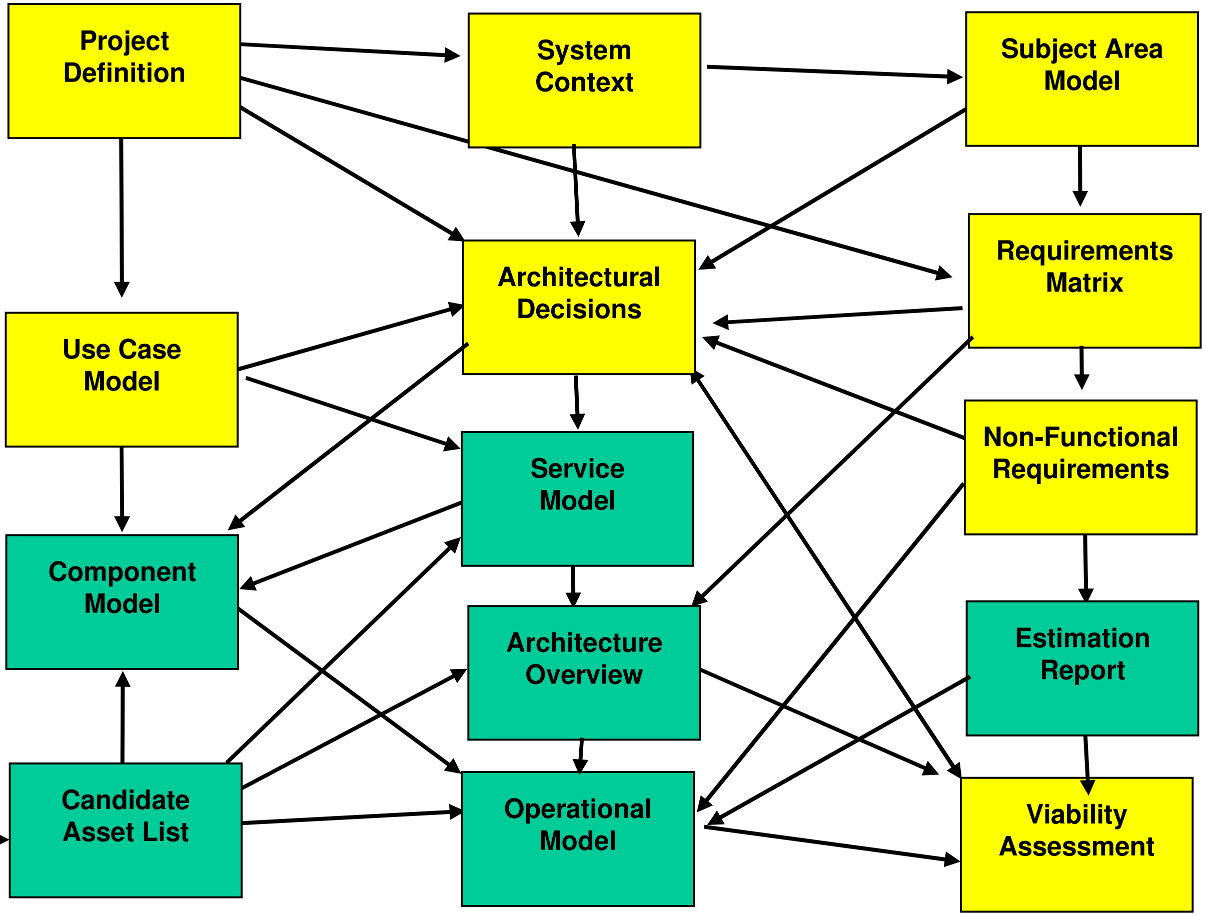
Work Product Dependency Diagram

Assets
Reference Architectures
ITT Solution Brief, Patterns
Redbooks, etc

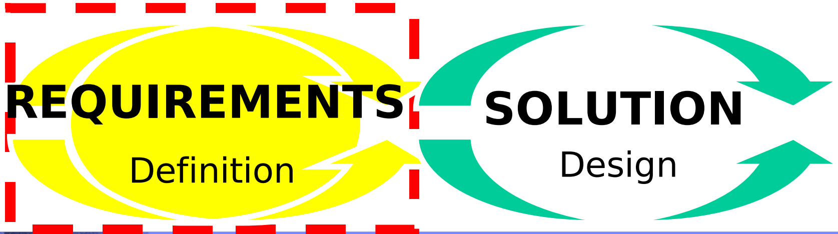




Work Product Dependency Diagram



Assets
Reference Architectures
ITT Solution Brief, Patterns
Redbooks, etc



Phase: REQUIREMENTS Definition

Task : Define Project

- Purpose : to turn an identified and validated opportunity into a defined project.
- Description :
  - This includes: "What are we doing on this project and why?" and getting a project sponsor "signed up".
  - **The Project Definition is updated through the life of the project.**

Work Product Started : Project Definition

- Purpose : to communicate and gain agreement to the project goals and status. This work product is defined by "Team Solution Design" and is required for every project.
- Description :
  - Answers to the questions: What, why, when, where, how and who?
  - Provides a concrete starting point for solution design.
  - Critical work product since most others used in "Team Solution Design" are dependent on it.
  - Together with Architectural Decisions, it ties the other work products together.
  - Project Definition and Viability Assessment are the primary mandatory work products for reviews.
- Note: if functional requirements are complex or in large numbers, they should be moved to Requirements Matrix.

Work Product Input/Updated : Plan Phase: Business Directions, Current Org, Current IT Environment, Standards

Phase: REQUIREMENTS Definition

Task : Identify and Outline Requirements

- Purpose :
  - Define a set of basic use cases that depict how the user will use the proposed system.
  - Understand and document additional functional requirements
- Description :
  - Identify additional functional requirements when required for more complex solutions
  - Develop initial use cases at a conceptual level.

Work Product Started : Use Case Model

- Purpose :
  - Establish a small number of important scenarios that depict how the user will use the proposed system
  - Provide a basis for planning a proof of concept and high level architectural walkthroughs
- Description
  - A set of use cases which illustrate primary usage scenarios and relationships of actors and use cases

Work Product Started : Requirements Matrix

- Purpose :
  - Document functional requirements that more elaborate than those described in the Project Definition
- Description
  - Documents detailed requirements of the proposed solution, especially the functional or software aspects

Work Product Input/Updated : Use Case Model, Project Definition, Component Model, Requirements Matrix

Phase: REQUIREMENTS Definition

Work Product : Requirements Matrix

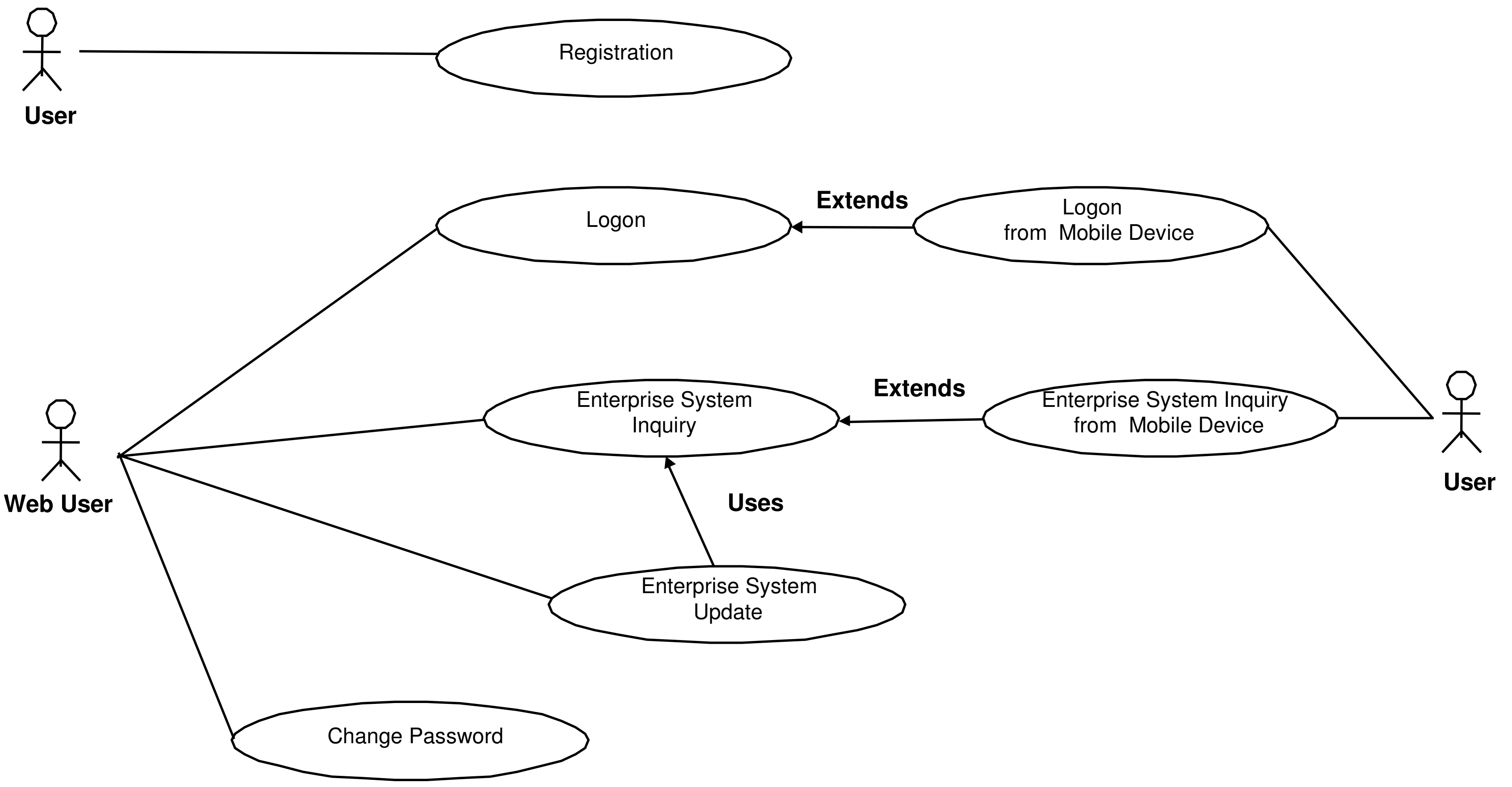
- The Requirements Matrix is a matrix that is used to capture client requirements for software selection and to evaluate the initial functional "fit" of a vendor's software solution to the business needs of the client. It allows the vendor to highlight their ability to satisfy the client's needs and provides an initial means to compare multiple vendors against each other. In addition, vendor response results can be used as elimination criteria in determining the vendor short list.

- The Requirements Matrix also is used to identify initial functional gaps or special software enhancements needed to enable each vendor's software to fulfill the client's desired system capabilities. Project Definition and Viability Assessment are the mandatory work products for quality reviews.

- The Requirements Matrix is a document detailing the functionality required by the client in its software solution. This document is generated from the results of the business process definition task and details the system requirements for the software being evaluated. This work product consists of business, general, and technical software functionality requirements.

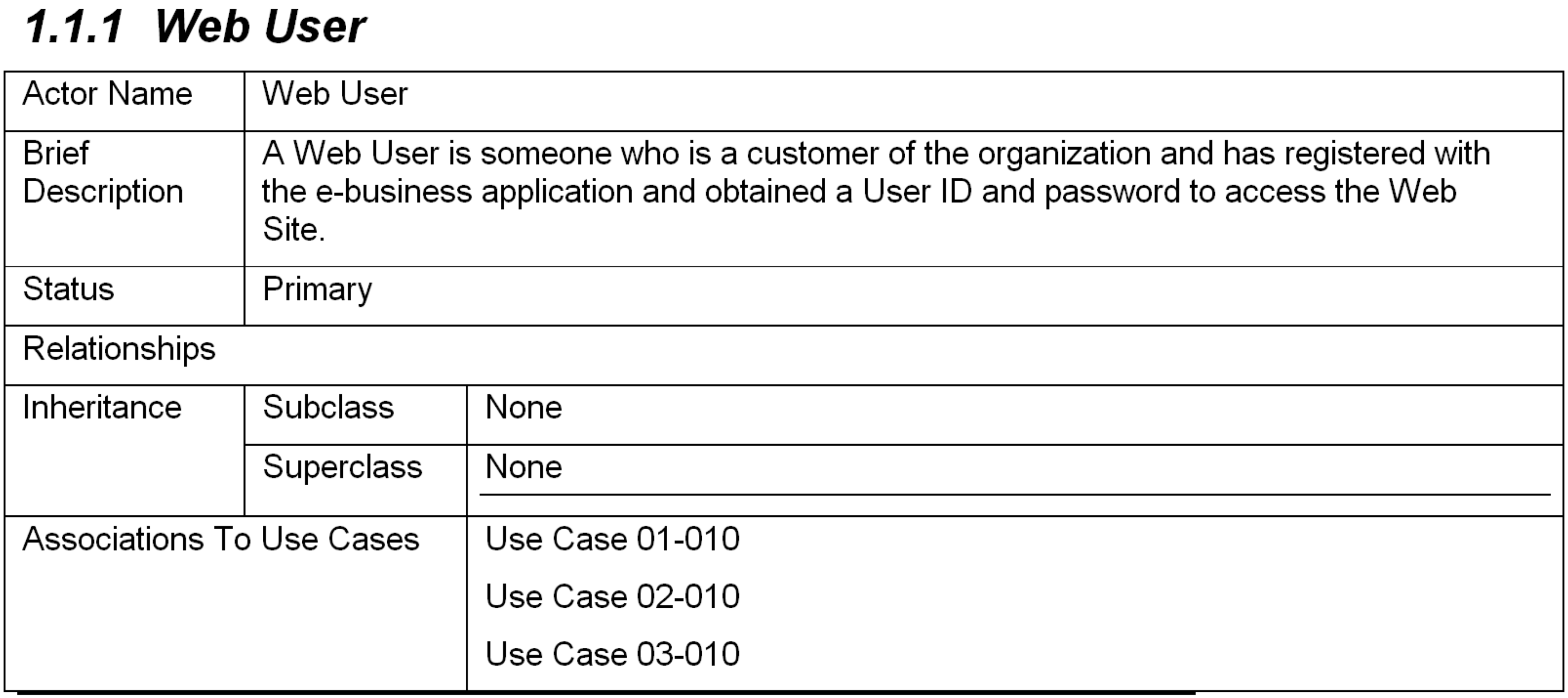
Phase: REQUIREMENTS Definition

Work Product : Use Case Model



Phase: REQUIREMENTS Definition

Work Product : Use Case Model

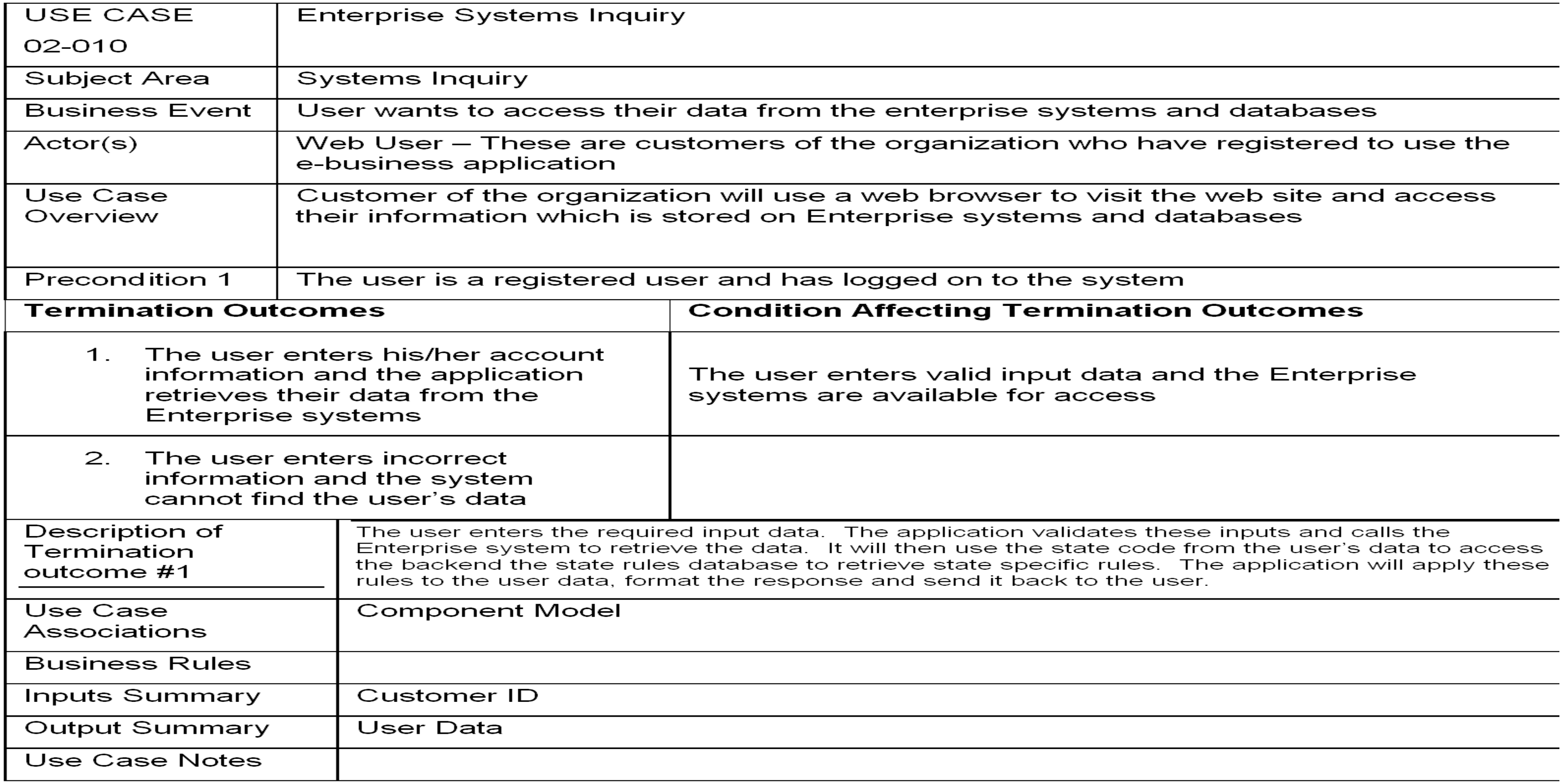


1.1.1 Web User

Actor Name	Web User	
Brief Description	A Web User is someone who is a customer of the organization and has registered with the e-business application and obtained a User ID and password to access the Web Site.	
Status	Primary	
Relationships		
Inheritance	Subclass	None
	Superclass	None
Associations To Use Cases		Use Case 01-010 Use Case 02-010 Use Case 03-010

Phase: REQUIREMENTS Definition

Work Product : Use Case Model



USE CASE 02-010	Enterprise Systems Inquiry
Subject Area	Systems Inquiry
Business Event	User wants to access their data from the enterprise systems and databases
Actor(s)	Web User – These are customers of the organization who have registered to use the e-business application
Use Case Overview	Customer of the organization will use a web browser to visit the web site and access their information which is stored on Enterprise systems and databases
Precondition 1	The user is a registered user and has logged on to the system
Termination Outcomes	**Condition Affecting Termination Outcomes**
1. The user enters his/her account information and the application retrieves their data from the Enterprise systems	The user enters valid input data and the Enterprise systems are available for access
2. The user enters incorrect information and the system cannot find the user's data	
Description of Termination outcome #1	The user enters the required input data. The application validates these inputs and calls the Enterprise system to retrieve the data. It will then use the state code from the user's data to access the backend the state rules database to retrieve state specific rules. The application will apply these rules to the user data, format the response and send it back to the user.
Use Case Associations	Component Model
Business Rules	
Inputs Summary	Customer ID
Output Summary	User Data
Use Case Notes	

Phase: REQUIREMENTS Definition

Task : Describe System Context

- Purpose :
  - Provide a basis for understanding the system to be proposed.
  - Define how the proposed system will interoperate with other existing systems.
  - Establish boundaries on the scope of the proposed system.
- Description :
  - The proposed system is treated as a black box with connections to other systems.
  - Documents all connections between the proposed system and external systems/components.
  - For each connection identify important attributes such as protocol, formats, frequency and volume.

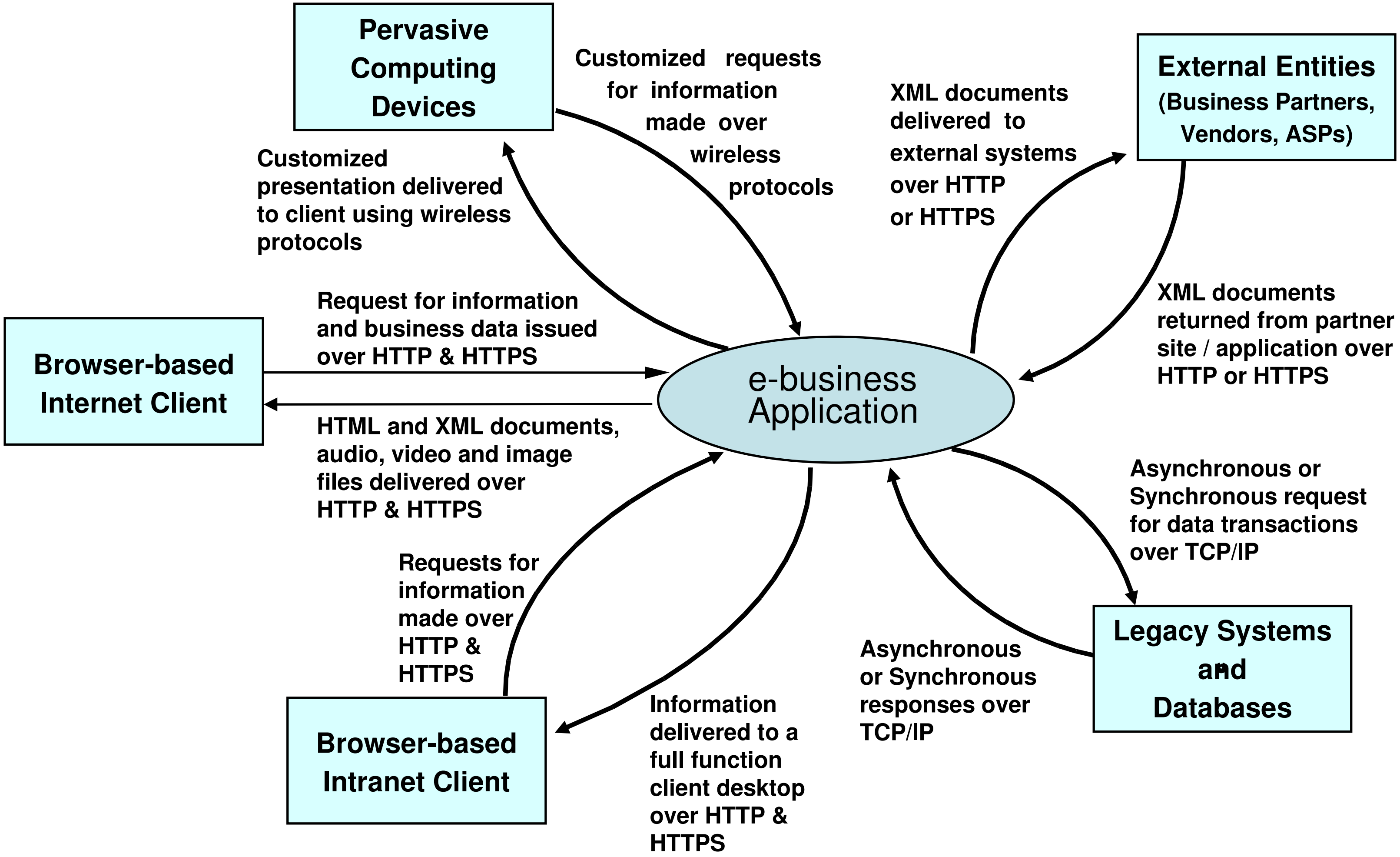
Work Product Started : System Context

- Purpose :
  - To clarify and confirm the environment in which the system has to operate.
  - To provide the details at an adequate level to allow the creation of the relevant technical specification.
- Description : Highlight important characteristics of the system:
  - Users, external systems, batch inputs and outputs, and external devices.
  - External events and data to which the system must respond.
  - Events and data that the system generates that affect external entities.

Work Product Input/Updated : Project Definition, System Context

Phase: REQUIREMENTS Definition

Work Product : System Context


Pervasive Computing Devices
Customized requests for information made over wireless protocols
Customized presentation delivered to client using wireless protocols
External Entities (Business Partners, Vendors, ASPs)
XML documents delivered to external systems over HTTP or HTTPS
XML documents returned from partner site / application over HTTP or HTTPS
Browser-based Internet Client
Request for information and business data issued over HTTP & HTTPS
HTML and XML documents, audio, video and image files delivered over HTTP & HTTPS
e-business Application
Asynchronous or Synchronous request for data transactions over TCP/IP
Legacy Systems and Databases
Asynchronous or Synchronous responses over TCP/IP
Requests for information made over HTTP & HTTPS
Browser-based Intranet Client
Information delivered to a full function client desktop over HTTP & HTTPS

Phase: REQUIREMENTS Definition

Task : Identify Non-Functional Requirements

- Purpose :
  - The purpose of this task is to identify non-functional requirements that will affect the design and resulting performance of the system.

- Description : In this task, various non-functional requirements and constraints are identified.
  - Identify service level requirements such as performance, capacity, volumes, availability, portability, maintainability, systems management and security.
  - Identify system constraints imposed by the client with regard to cost, location, configuration, standards, vendor preferences and technology preferences.

Work Product Started : Non-Functional Requirements

- Purpose :
  - To define requirements and constraints on the IT system.
  - Provide a basis for early system sizing and estimates of cost and viability assessment of the proposed IT system.
- Description :
  - Documents the non-functional aspects of an IT system including examples such as:
  - Performance, scalability, availability, maintainability, manageability, usability, accessibility, and data Integrity

Work Product Input/Updated : Non-Functional Requirements, Project Definition, Use Case Model

Phase: REQUIREMENTS Definition

Work Product : Non-Functional Requirements

- Availability
- Backup & Recovery
- Capacity Estimates and Planning
- Configuration Management
- Disaster Recovery
- Environmental factors
- Extensibility/Flexibility
- Failure Management
- Maintainability
- Performance
- Quality of Service
- Reliability
- Scalability
- Security
- Service Level Agreements
- Standards
- Systems Management

Phase: REQUIREMENTS Definition

Task : Identify High Level Data Sources

- Purpose :
  - Identify and describe at a high level, data sources which are relevant to the proposed solution.
  - Contribute to an understanding of architectural aspects which are necessary to use existing data sources
- Description
  - Document high existing level data sources that will be used as a part of the solution
  - Document additional high level data sources that will be required for the solution
  - Develop a conceptual data model of all relevant high level data sources

Work Product Started : Subject Area Model

- Purpose :
  - Convey, graphically or textually, the scope of an enterprise, a desired capability or application from the point of the view of the data or information required to support the enterprise, application or capability.
- Description
  - Graphical as well as textual document of the major groupings of entities that are needed to support an enterprise, a capability or an application
  - Also referred to as a Conceptual Data Model or Business Information Model

Work Product Input/Updated : Subject Area Model, Project Definition, Component Model, Requirements Matrix

Phase: REQUIREMENTS Definition

Work Product : Subject Area Model

- Purpose:

  Identify and describe at a high level, data sources which are relevant to the proposed solution.

  Contribute to an understanding of architectural aspects which are necessary to use existing data sources.

- Description:

  Document high existing level data sources that will be used as a part of the solution.

  Document additional high level data sources that will be required for the solution.

  Develop a conceptual data model of all relevant high level data sources.

- The primary output work product from this task is: Subject Area Model (APP 408)

- The Subject Area Model work product is a graphical as well as textual document of the *major groupings of entities that are needed to support an enterprise*, a capability or an application. It may alternatively be referred to as a Conceptual Data Model or Business Information Model. A Subject Area Model work product is deliberately high-level in definition as well as use. As such, it is an excellent communication vehicle for an executive or business management audience. This work product is generally developed in parallel to work products that document a high-level vision of a solution.

Phase: REQUIREMENTS Definition

Task : Document Architectural Decisions

- Purpose :
  - Identify and document important architectural decisions where alternatives exist, choices are unclear and impact is likely significant.
- Description :
  - Document architectural decisions regarding principles or policies.
  - Document architectural decisions regarding elements of the architecture.
  - Ensure the issue or problem is clearly stated, evaluate the options that are available, make the decision.

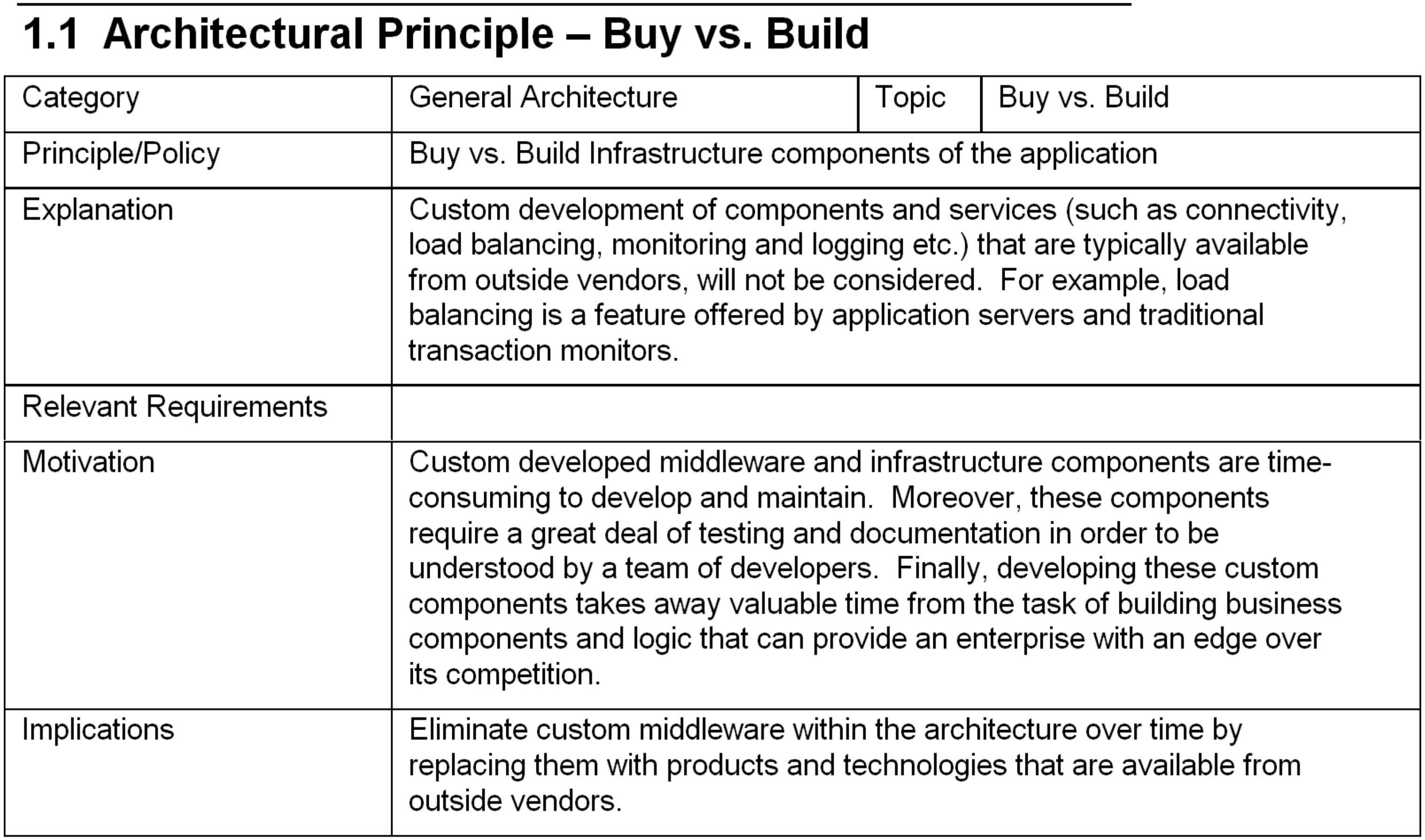
Work Product Started : Architectural Decisions

- Purpose :
  - Provide a single place to find important architectural decisions
  - Make explicit the rationale and justification of Architectural Decisions
  - Avoid unnecessary reconsideration of the same issues
- Description :
  - An Architectural Decisions work product documents important decisions about any aspect of the architecture including the structure of the system, the provision and allocation of function, the contextual fitness of the system and adherence to standards.

Work Product Input/Updated : Architectural Decisions, Project Definition, Non-Functional Requirements

Phase: REQUIREMENTS Definition

Work Product : Architectural Decisions



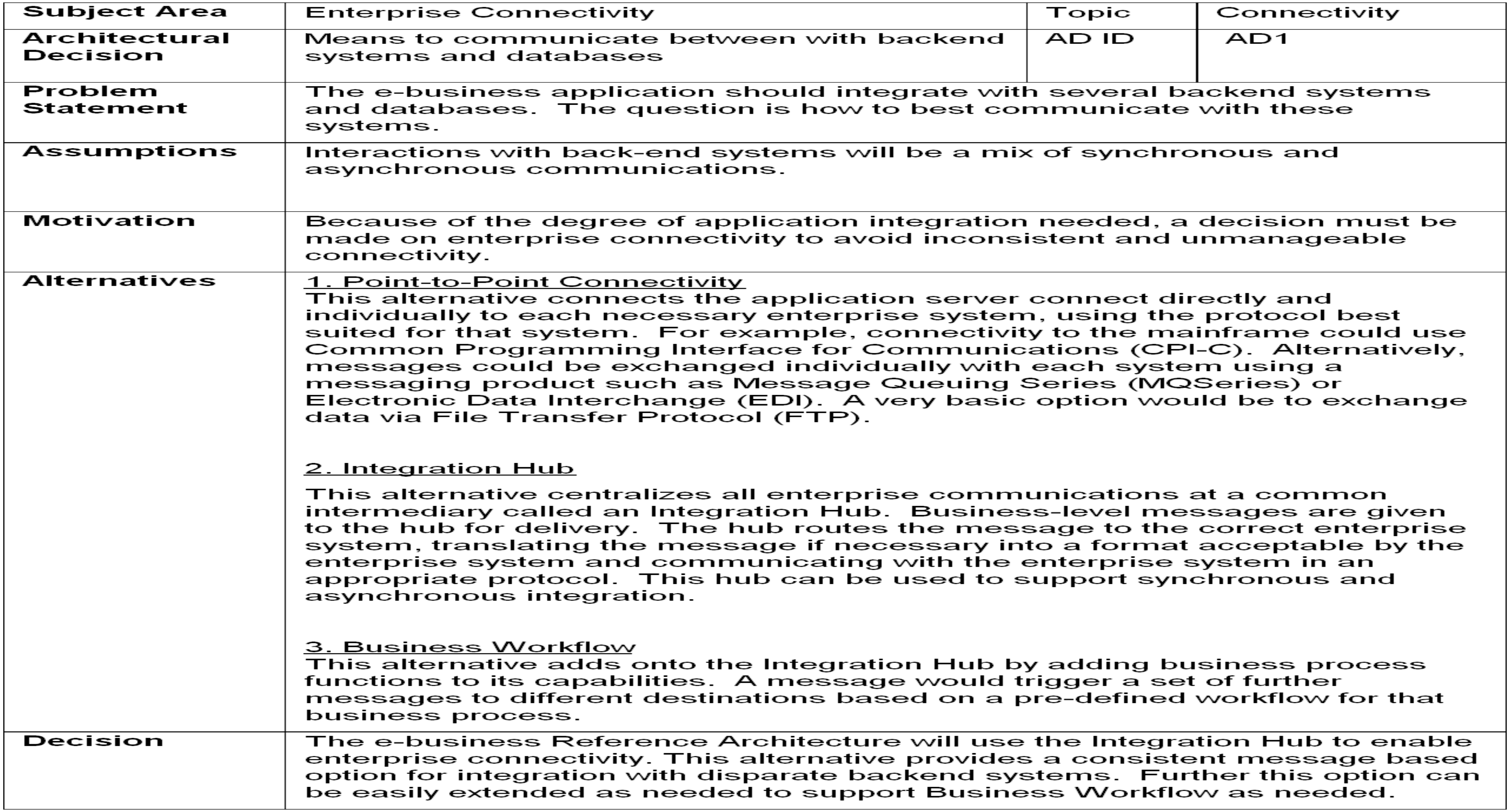
1.1 Architectural Principle – Buy vs. Build

Category	General Architecture	Topic	Buy vs. Build
Principle/Policy	Buy vs. Build Infrastructure components of the application		
Explanation	Custom development of components and services (such as connectivity, load balancing, monitoring and logging etc.) that are typically available from outside vendors, will not be considered. For example, load balancing is a feature offered by application servers and traditional transaction monitors.		
Relevant Requirements			
Motivation	Custom developed middleware and infrastructure components are time-consuming to develop and maintain. Moreover, these components require a great deal of testing and documentation in order to be understood by a team of developers. Finally, developing these custom components takes away valuable time from the task of building business components and logic that can provide an enterprise with an edge over its competition.		
Implications	Eliminate custom middleware within the architecture over time by replacing them with products and technologies that are available from outside vendors.		

Phase: Pre-sale Solution Design

Activity: EXPLORE Options and Approach

Work Product : Architectural Decisions



Subject Area	Enterprise Connectivity	Topic	Connectivity
Architectural Decision	Means to communicate between with backend systems and databases	AD ID	AD1
Problem Statement	The e-business application should integrate with several backend systems and databases. The question is how to best communicate with these systems.		
Assumptions	Interactions with back-end systems will be a mix of synchronous and asynchronous communications.		
Motivation	Because of the degree of application integration needed, a decision must be made on enterprise connectivity to avoid inconsistent and unmanageable connectivity.		
Alternatives	<u>1. Point-to-Point Connectivity</u> This alternative connects the application server connect directly and individually to each necessary enterprise system, using the protocol best suited for that system. For example, connectivity to the mainframe could use Common Programming Interface for Communications (CPI-C). Alternatively, messages could be exchanged individually with each system using a messaging product such as Message Queuing Series (MQSeries) or Electronic Data Interchange (EDI). A very basic option would be to exchange data via File Transfer Protocol (FTP). <u>2. Integration Hub</u> This alternative centralizes all enterprise communications at a common intermediary called an Integration Hub. Business-level messages are given to the hub for delivery. The hub routes the message to the correct enterprise system, translating the message if necessary into a format acceptable by the enterprise system and communicating with the enterprise system in an appropriate protocol. This hub can be used to support synchronous and asynchronous integration. <u>3. Business Workflow</u> This alternative adds onto the Integration Hub by adding business process functions to its capabilities. A message would trigger a set of further messages to different destinations based on a pre-defined workflow for that business process.		
Decision	The e-business Reference Architecture will use the Integration Hub to enable enterprise connectivity. This alternative provides a consistent message based option for integration with disparate backend systems. Further this option can be easily extended as needed to support Business Workflow as needed.		

Phase: REQUIREMENTS Definition

Work Product : Architectural Decisions

Justification	Because of the number of different systems that must be accessed by any e-business application, it would seem prudent to have the application server communicate using one method (messages) to one system (the Integration Hub). This also provides the flexibility of relocating data and functions to different enterprise systems with no change to systems that access that data. Furthermore, the Integration Hub will simplify communications among all legacy systems and databases, allowing other projects within the enterprise to leverage this investment. The translation capabilities of the Integration Hub are also a big advantage in the business-to-business world. It is expected that this hub would be accessible to authorized trading partners, each of whom may adopt a different message "standard" for their business-to-business (B2B) communications. The Integration Hub would be able to translate these different message formats into a consistent format for communication with the organization's enterprise system. Another advantage of this option is its ability to support different collaboration models such as blackboard, point-to-point or Publish/Subscribe. Although the business workflow option is intriguing and seems to offer benefits, at this time there does not seem to be a sufficient gain in this process for the cost incurred. However, as the organization's business and technical processes evolve, this option could be added onto the Integration Hub's functionality.
Implications	All enterprise applications will need to be enabled for working with the Integration hub. A set of business messages will need to be defined, as well as translation rules for other systems.
Derived requirements	
Related Decisions	AD2, Enterprise Message Format

Phase: REQUIREMENTS Definition

Task : Conduct Viability Assessment

- Purpose :
  - To qualify the opportunity: assess if the opportunity qualifies for further investment by IBM and the client.
  - To make an initial assessment of the viability of the solution ensuring that it lies within the "art of the possible".
  - To identify unrealistic or challenging requirements as early as possible, and seek to re-negotiate them.
- Description :
  - Pre-Bid Consulting engages the Risk Manager role.
  - Pre-bid Consulting also names the End-to-End Delivery Owner.

Work Product Started : Viability Assessment

- Purpose : to document the status of the project.
  - Together with Project Definition, it is the primary vehicle for communication.
  - Besides project risks, the "Team Solution Design" version documents issues, assumptions and dependencies that might impact the proposal, implementation and delivery.
- Description:
  - Depending on the risk and complexity of the project, several formal peer reviews that might be required.
  - Reviews include: Technical Delivery Assessment (Solution Assurance), Integrated Technical Review, Proposal Baseline Assessment and Project Management Review.

Work Product Input/Updated : Project Definition. Viability Assessment

Phase: REQUIREMENTS Definition

Work Product : Viability Assessment

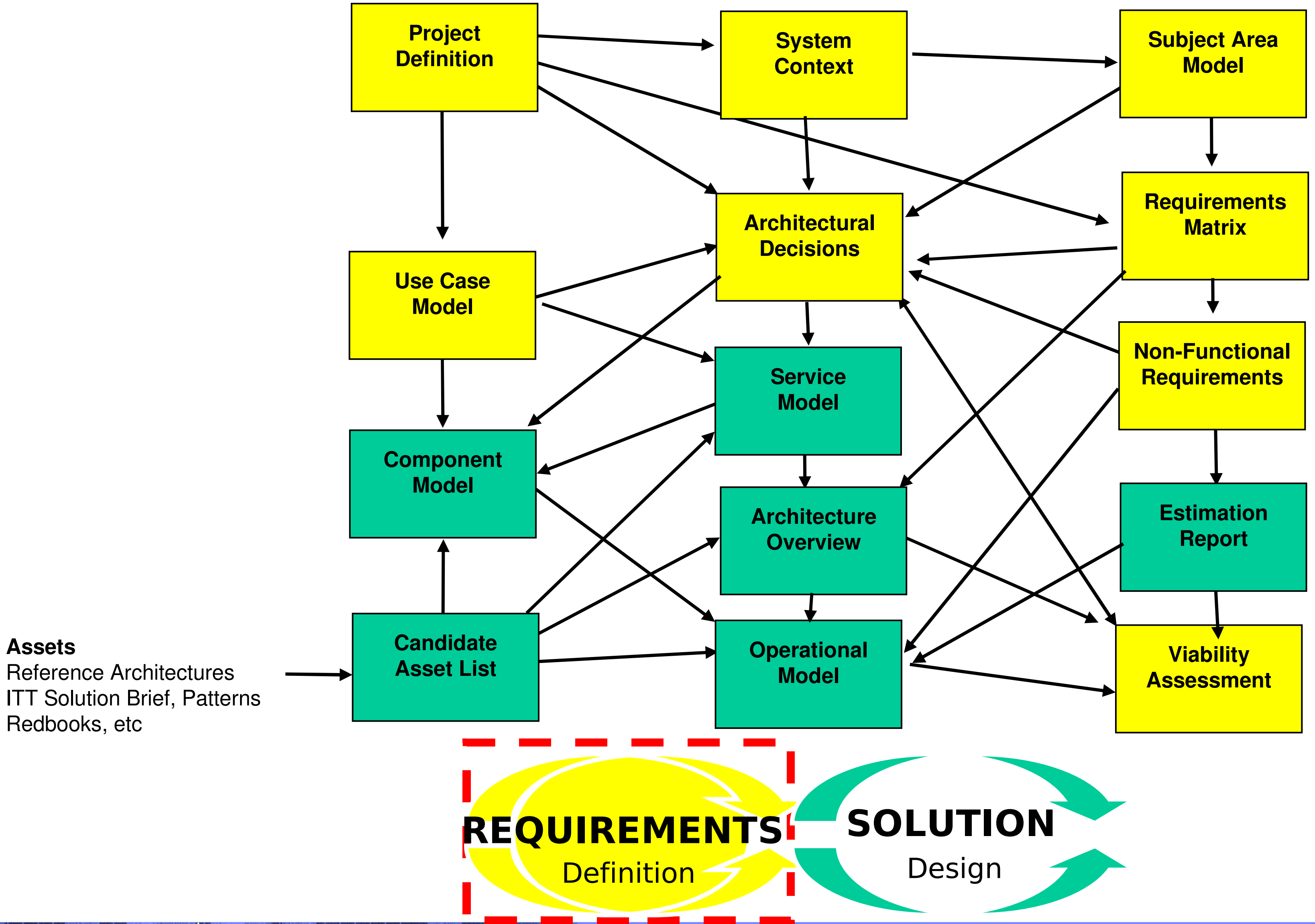
Risk ID	Finding / Risk Description	Probability (H/M/L)	Impact (H/M/L)	Effort / Cost	Contingency / Mitigation Recommendation	Person Responsible	Review Date
R01							
R02							
R03							

Assumption ID	Finding / Assumption Description	Confidence Level (H/M/L)	Impact (H/M/L)	Assumption Identified By	Review Date	Closed Date
A01						
A02						
A03						

Issue ID	Finding / Issue Description	Priority (H/M/L)	Raised By / Date	Issue Responsibility	Review Date	Action: Closed or Risk / Change Reference
I01						
I02						
I03						

Dependency ID	Finding / Dependency Description	Effect on Plan	Required By Date	Owner	Associated Risk ID	Closed Date
D01						
D02						
D03						

Work Product Dependency Diagram



Assets
Reference Architectures
ITT Solution Brief, Patterns
Redbooks, etc