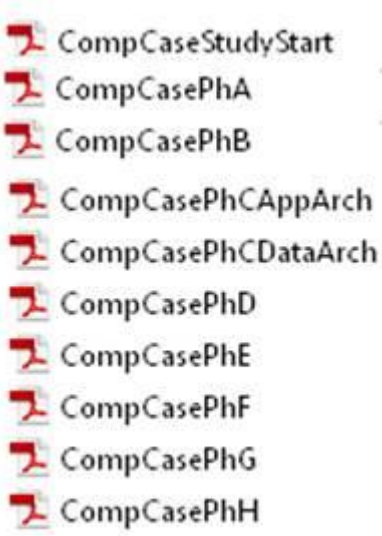
Comprehensive Case Study : Phase H

Change Management

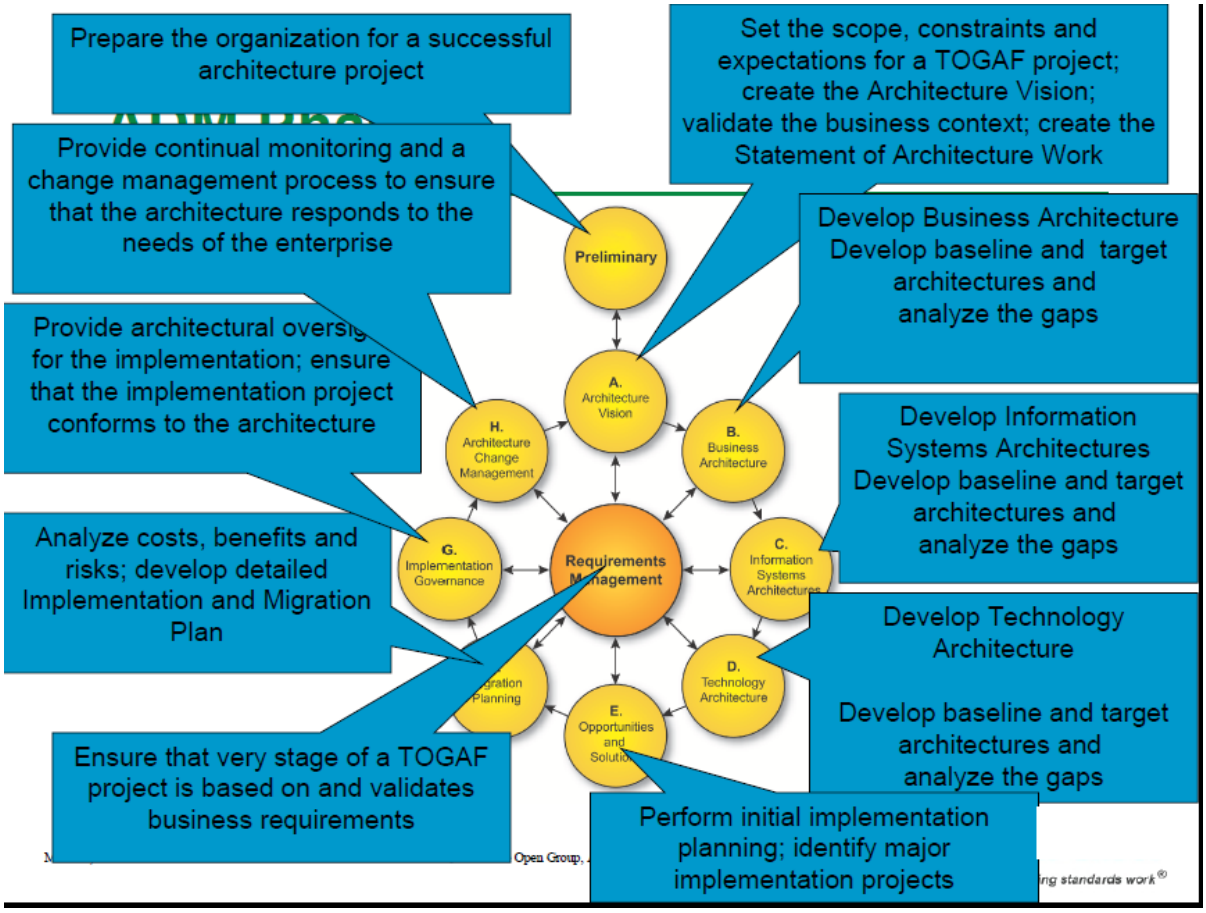
Files of this Case Study are to be referred and continued only in this sequence order, please :



Though Requirement Management Phase is aligned to every Phase from A to H, it is covered towards end of this document

What stage to look at Change ?

During any one of Phase B, C, D, Or even E or F ?
From a Phase onwards, like C onwards through D ,,,
Only at the end of Phase G, as a trigger for fresh ADM from Phase A ?



Step : Establishing Value Realization Process

Influenced business projects to exploit the Enterprise Architecture for value realization (outcomes)

EAs take action to continually exploit value realization as outcomes

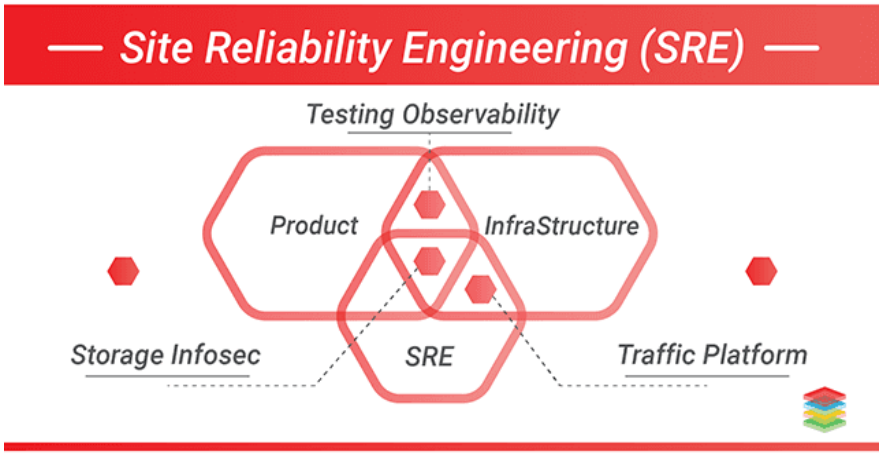
Step : Deploy Monitoring Tools

With CD / CI, Cloud Native and other enabling techniques being in use for the e-Commerce project, the following are necessary :

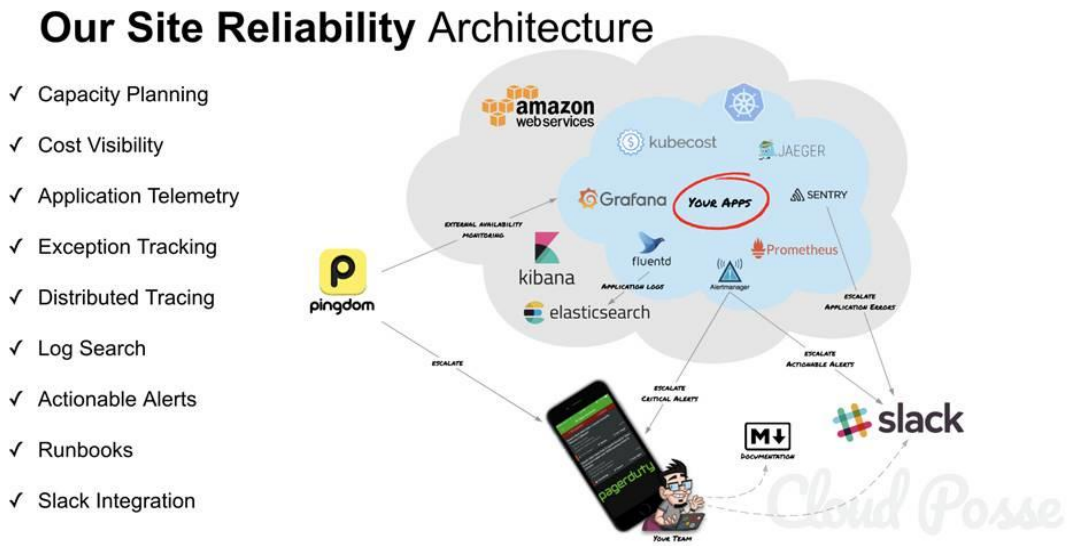
Site Reliability Engineering (SRE) bridges the gap between development and operations, combining software and systems engineering to build large-scale and highly protected systems. It uses automation and orchestration capabilities to scale security and performance, ensuring sites are reliable and efficient.

Site reliability engineering is critical to many Cloud, DevOps and automation initiatives, and includes tactics like:

- Test coverage
- Load balance testing
- CI/CD best practices
- Modernizing legacy systems
- Executing integrations
- Platform configuration management



SRE is fundamentally doing work that has historically been done by an operations team, but using engineers with software expertise and banking on the fact that these engineers are inherently both predisposed to, and have the ability to, substitute automation for human labour. In general, an SRE team is responsible for availability, latency, performance, efficiency, change management, monitoring, emergency response, and capacity planning."



The behavior and state of a microservice should be observable : at any time, we should be able to determine whether the service is healthy and whether it's processing its workload in the way we expect. If something affects a key metric —this should send an actionable alert to the engineering team

For example, for a critical service, one might aim for 95% of requests to return in under 100ms with 99.99% uptime. Failing to meet these thresholds should result in alerts being sent to the service owners.

A service timeout may be due to several underlying reasons: network issues, problems with service-internal dependencies —such as databases —or unhealthy behavior from other services.

To avoid such problems in the future, they had to add

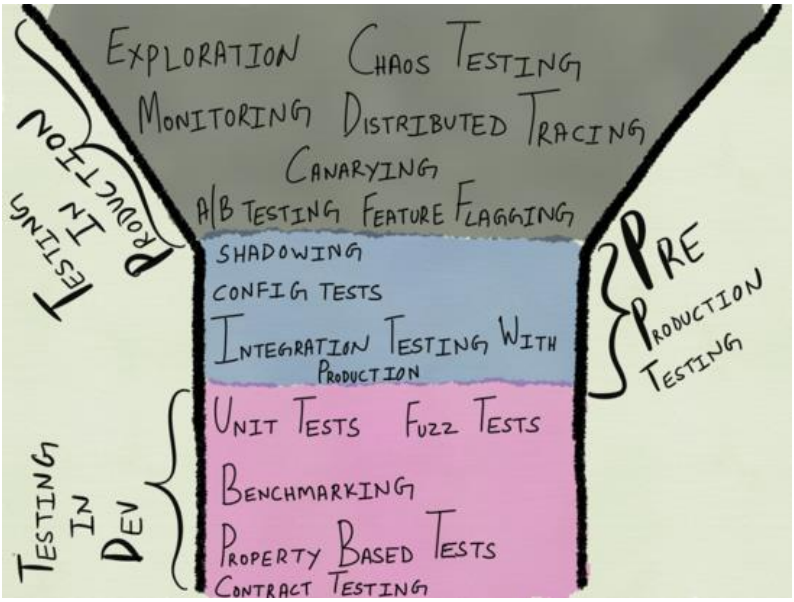
thorough instrumentation to the microservices. Collecting data about application activity —at all layers —is vital to understanding the present and past operational behavior of a microservice application.

They installed a logging collection agent on each instance. This ships application log data to a central repository where one can index, search, and analyze it further.

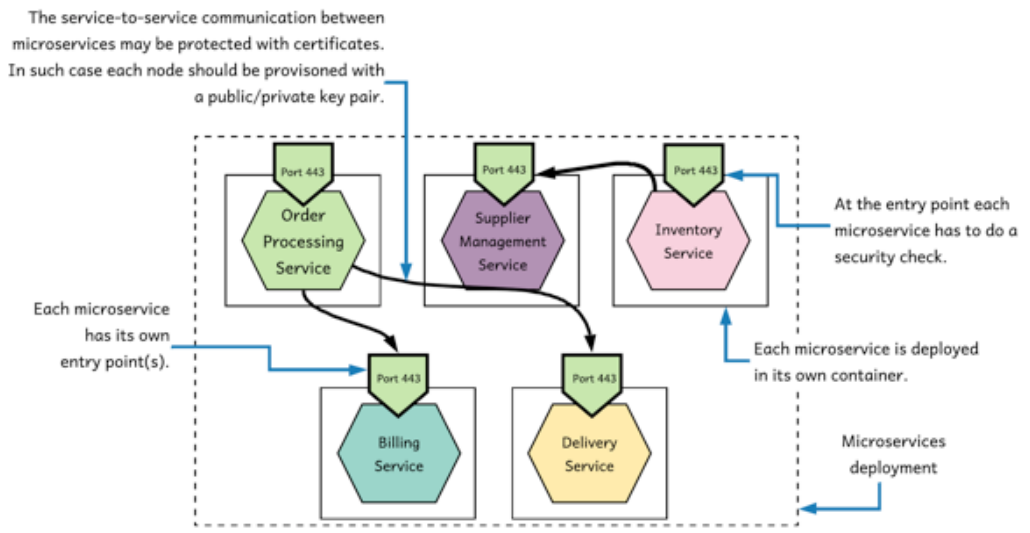
Step : Manage Risks

Manage Enterprise Architecture risks and provide recommendations for IT strategy.

As the number of microservices increases, managing them gets more challenging. It is important that the management aspects are planned before or while they are being built. While the modularity helps, things can very quickly get out of hand if not managed well.



For example : Security related issues and Risks :


The service-to-service communication between microservices may be protected with certificates. In such case each node should be provisoned with a public/private key pair.
Port 443
Order Processing Service
Port 443
Supplier Management Service
Port 443
Inventory Service
At the entry point each microservice has to do a security check.
Each microservice has its own entry point(s).
Port 443
Billing Service
Port 443
Delivery Service
Each microservice is deployed in its own container.
Microservices deployment

Step : Provide Analysis for Architecture Change Management

The EA Team, in coordination with the Operations Team, and on a continues basis :

- Analyses performance
- Conducts Enterprise Architecture performance reviews with service management
- Assesses Change Requests and reporting to ensure that the expected value realization and Service-Level Agreement (SLA) expectations of the customers are met
- Undertakes a gap analysis of the performance of the Enterprise Architecture
- Ensures that change management requests adhere to the Enterprise Architecture Governance and framework

Step : Develop Change Requirements to Meet Performance Targets

The EA Team, from time to time, makes recommendations on change requirements to meet performance targets and development of position to act.

Step : Manage Governance Process

We do have a Governance Board. How come the EA Team is asked to do Governance ?

This step really means playing a role as part of the Change Management, which is a Governance function.

Arranging meeting of Architecture Board for considering major changes

Hold periodical meetings with the Architecture Board with the aim to decide on handling changes (technology and business and dispensations)

Step : Activate the Architecture process to implement change

EA Team produces a new Request for Architecture Work and request for investment before every fresh Phase A. It is planned for Year 2 and Year 3

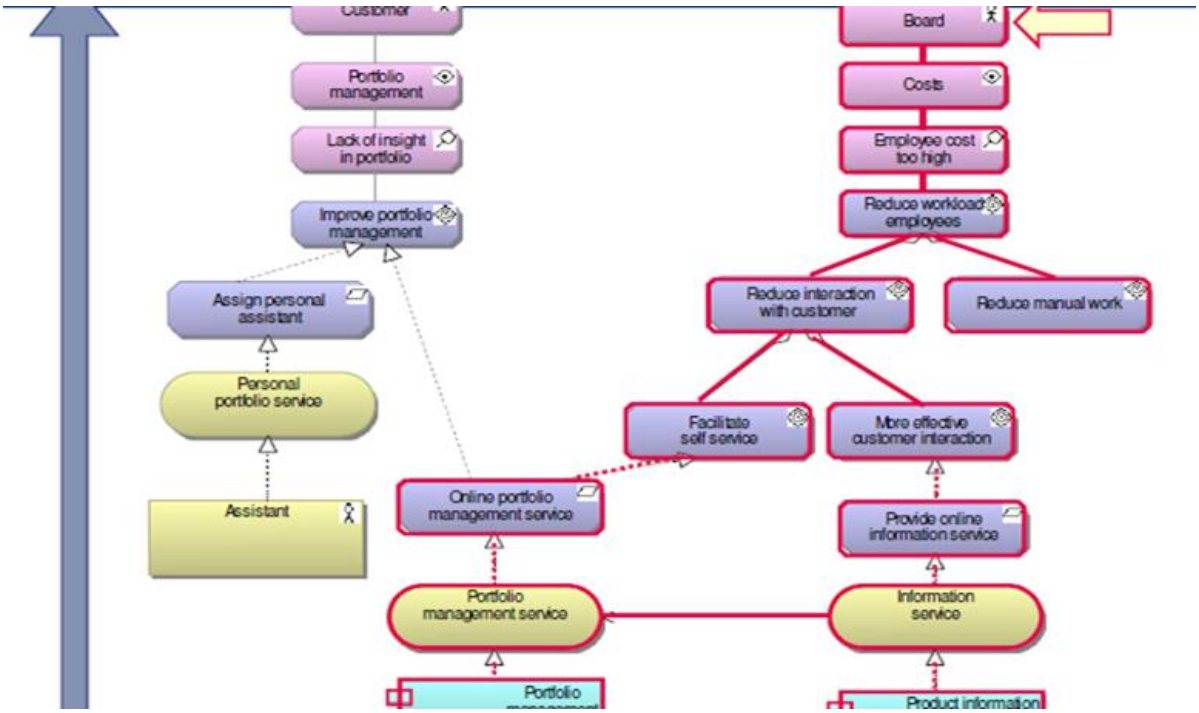
EA Team further ensures that any changes implemented in this Change Management Phase at any time are captured and documented in the Architecture Repository

Recollect : What stage to look at Change ?

During any one of Phase B, C, D, Or even E or F ?

From a Phase onwards, like C onwards through D ,,,

Only at the end of Phase G, as a trigger for fresh ADM from Phase A ?



Architecture Change Management & Requirements Management : ArchiMate Sample

Documents Updated in this Phase :

Requirement Impact Statement

Architecture Change Request

Change Request Register							
#	Change Request	Requested By	Requested Date	Request Type	Request Priority	Est. Comp Date	Status
1	Request 1	Name	Date	Scope Change	High	Date	On Target
2							
3							
4							

Change Request Register *maintains all Architectural change requests*

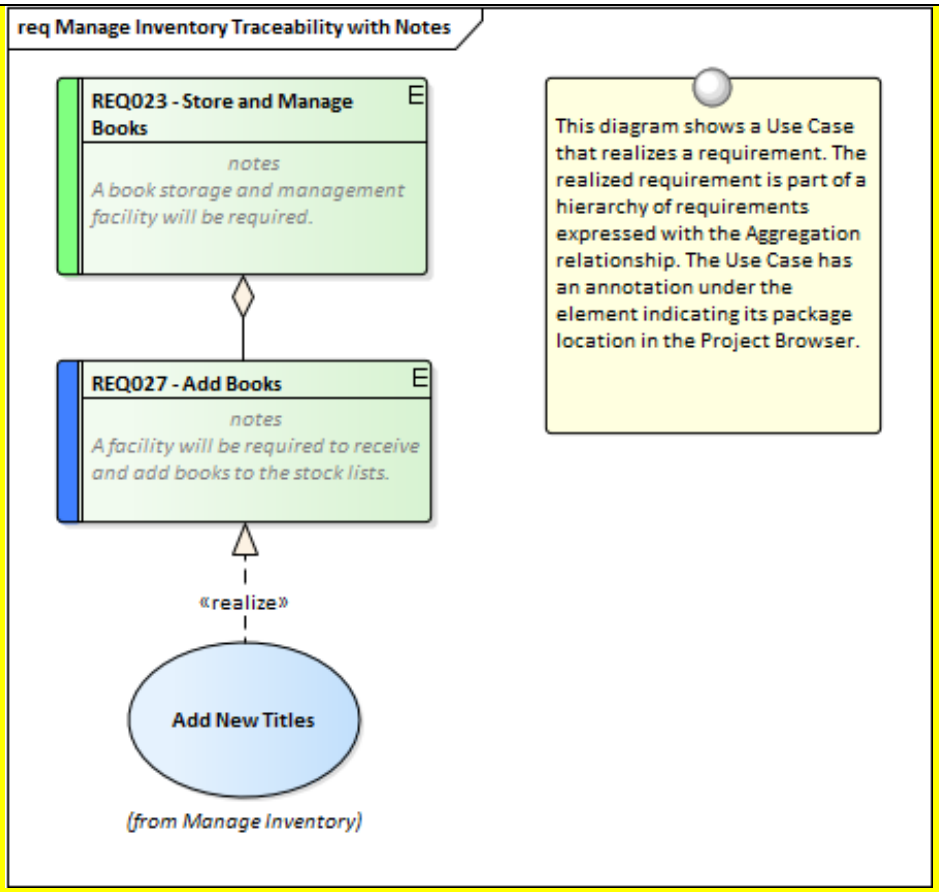
Documents updated in this Phase are ones that are more related to Requirement Management – the central Phase

Case Study – Requirement Management Phase :

Typical contents of a **Requirements Impact Assessment** are :

- Reference to specific requirements
- Stakeholder priority of the requirements to date
- Phases to be revisited
- Phase to lead on requirements prioritization
- Results of phase investigations and revised priorities
- Recommendations on management of requirements
- Repository reference number

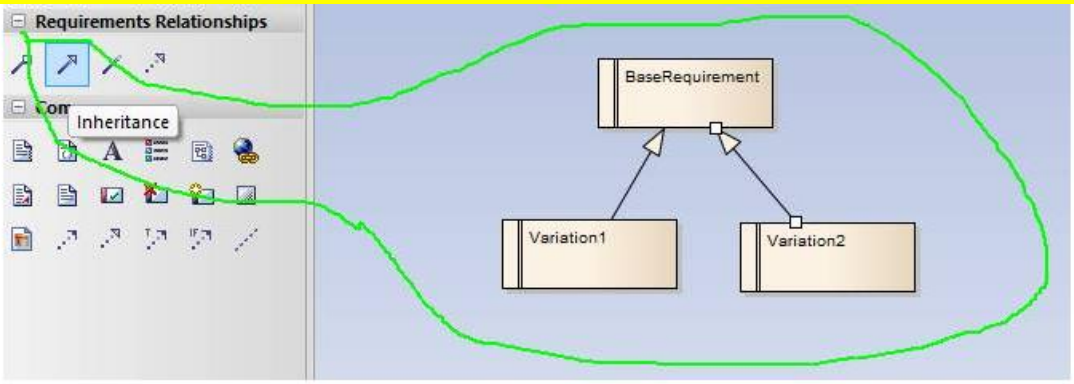
The Requirements Diagram provides a visual representation of how Requirements are related to each other and to other elements in the model, including Business Drivers, Constraints, Business Rules, Use Cases, User Stories, design Components and more.

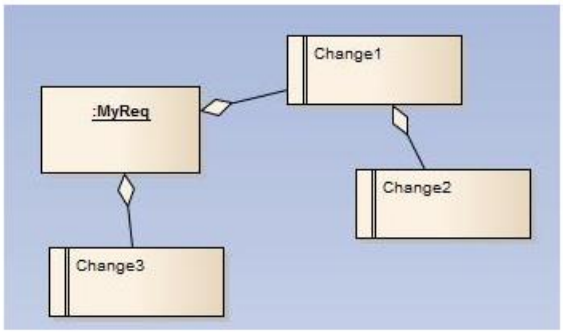


One usage is to show how Requirements are connected together in a hierarchy but a more compelling usage is to show how requirements are connected to other elements.

The experienced modeler will define and manage the requirements in the Specification Manager and then use the requirements diagram to show how one or more requirements are related to up-process elements such as Business Drivers and down-process elements such as Use Cases, User Stories, User Experience designs and solution Components.

Requirement may undergo change :





A Functional requirement : "The client must be able to order a product online."

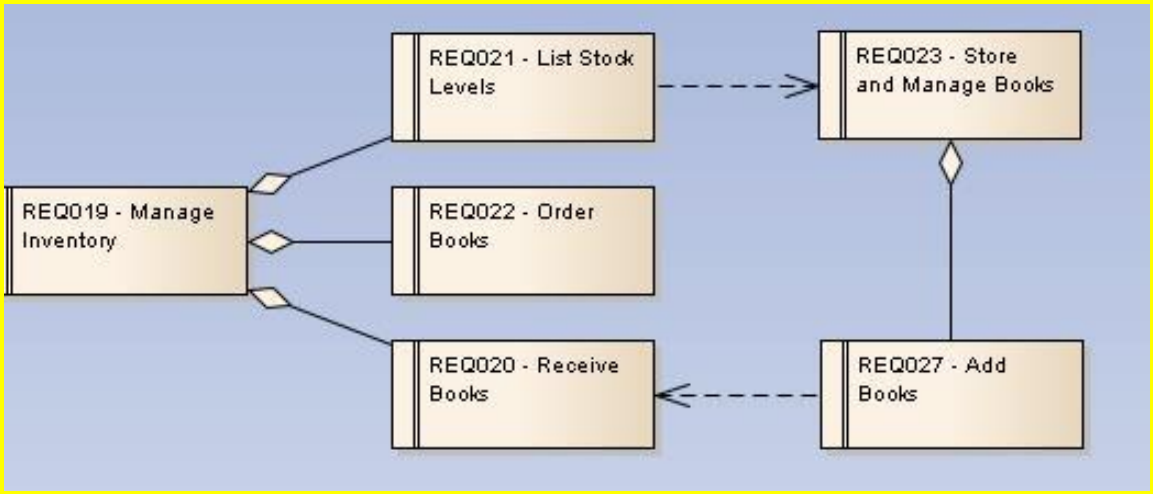
A Non Functional requirement : "It must be possible to place an order at all times."

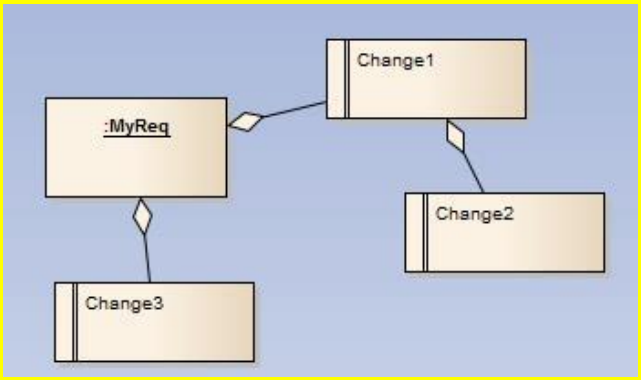
Tracing tools can be used to track what was changed in a model, who changed it and when. While a baseline can be used to show the difference between a model and a snapshot at a point in time, the tool records each individual change; it may not, however, be used to revert to a previous state.

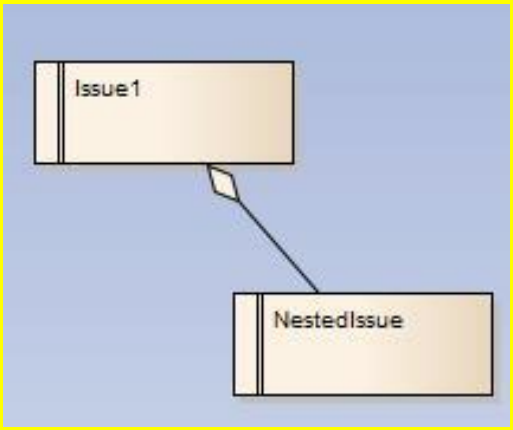
Requirements negotiation

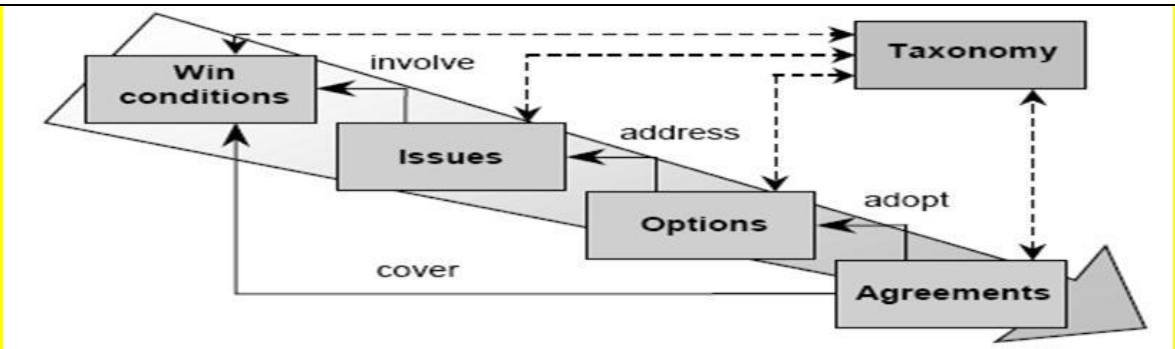
Architect has to negotiate the requirement as it evolves

Capture the Requirement and **Manage changes** to it as the project evolves.



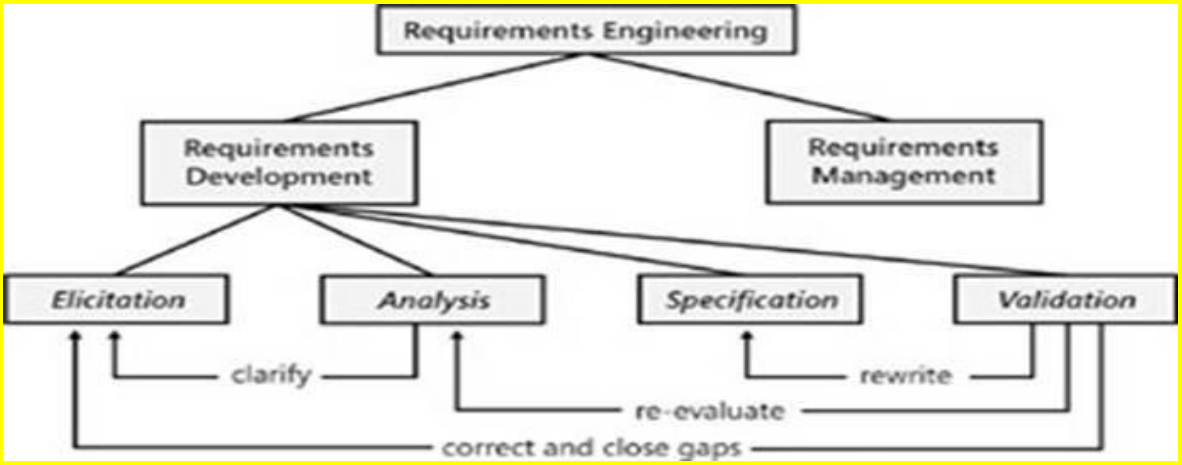




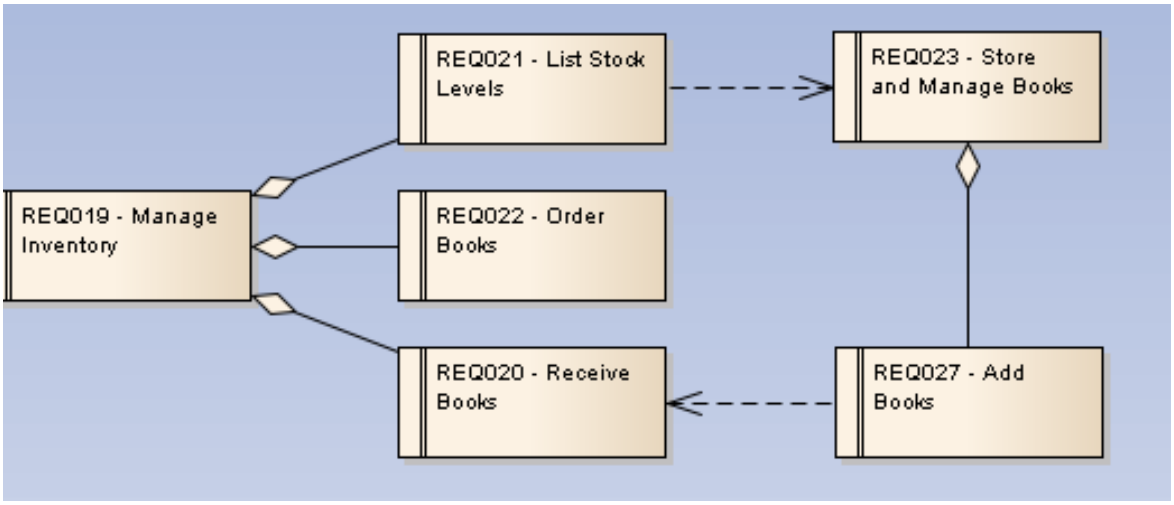


Negotiation activity may vary by stakeholder role.

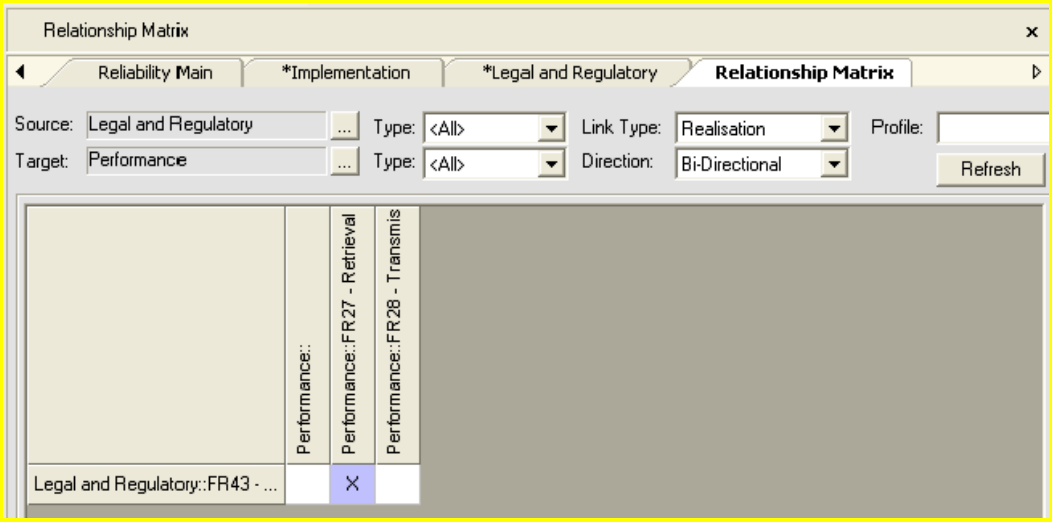
Users and customers would be more active in the early stages; Developers and customers in the late stages



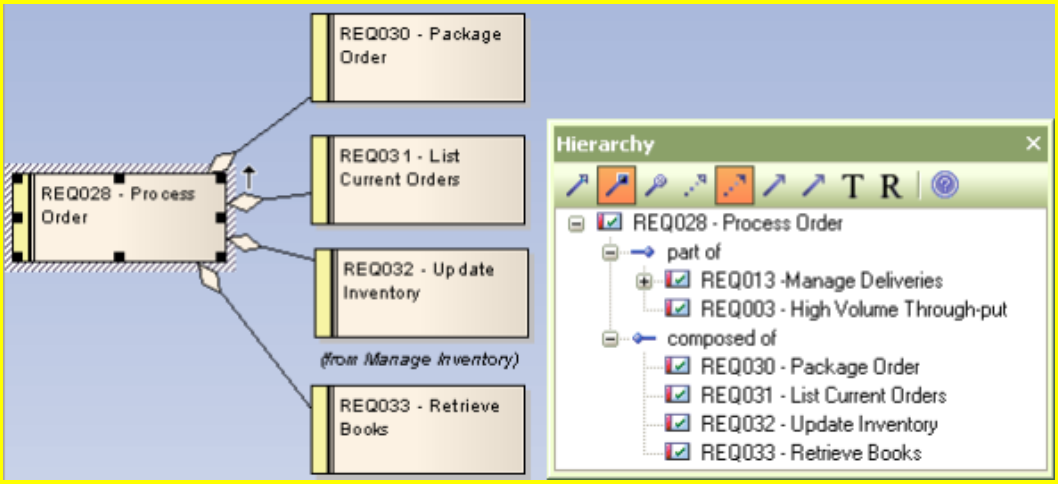
Requirements Management



The relationship matrix :

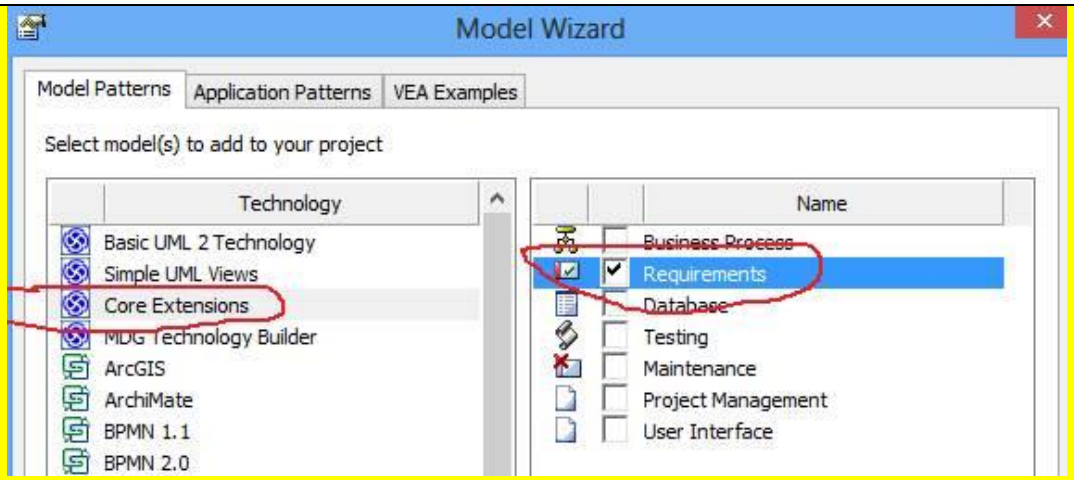


Traceability Using the Hierarchy Window :

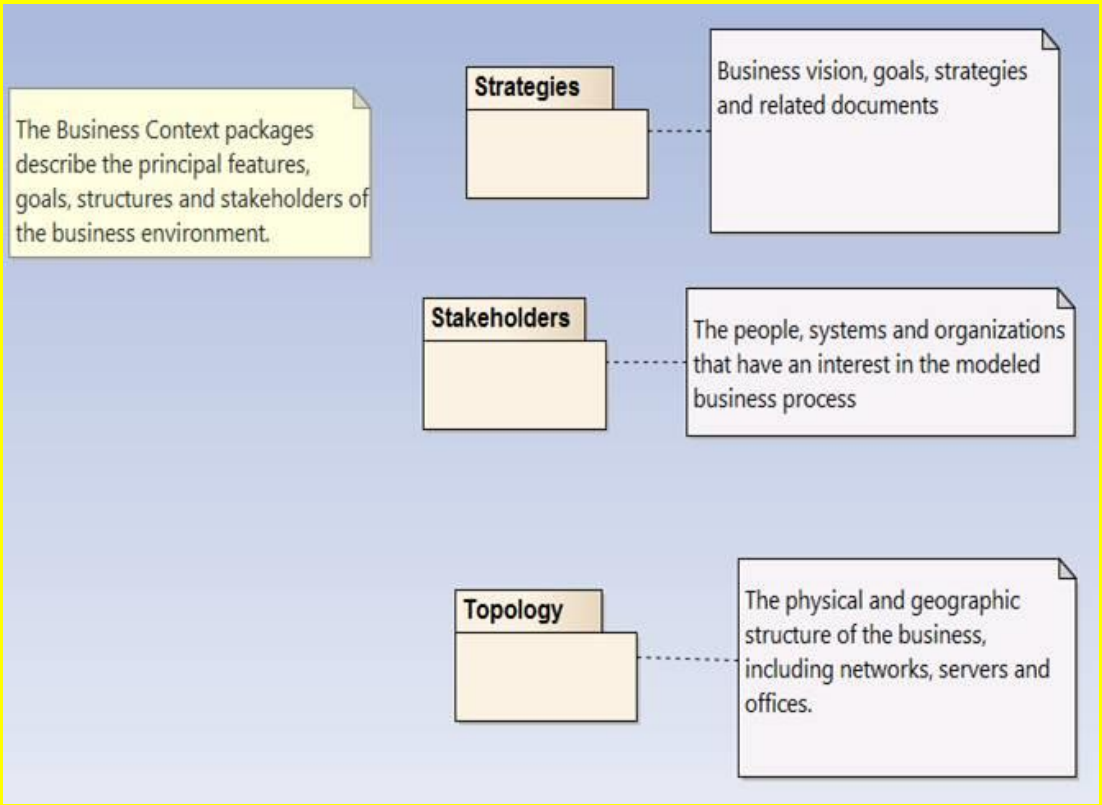


Tool : Enterprise Architect

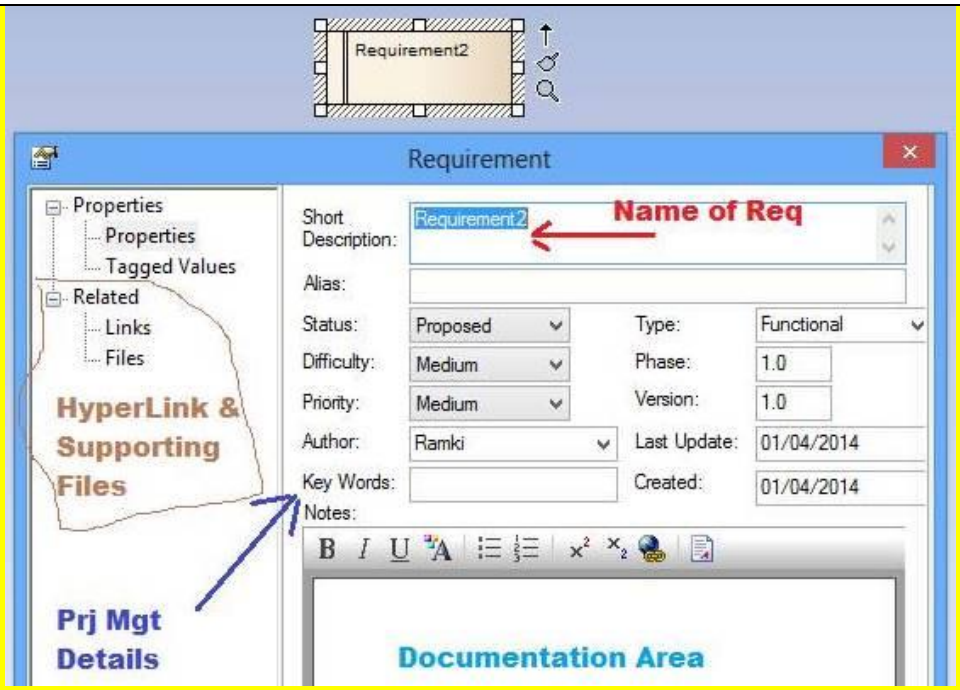
In this Tool, Requirements management is built into the core product, solving many of the issues of traceability, interdisciplinary team divisions, integration with change and configuration management systems.



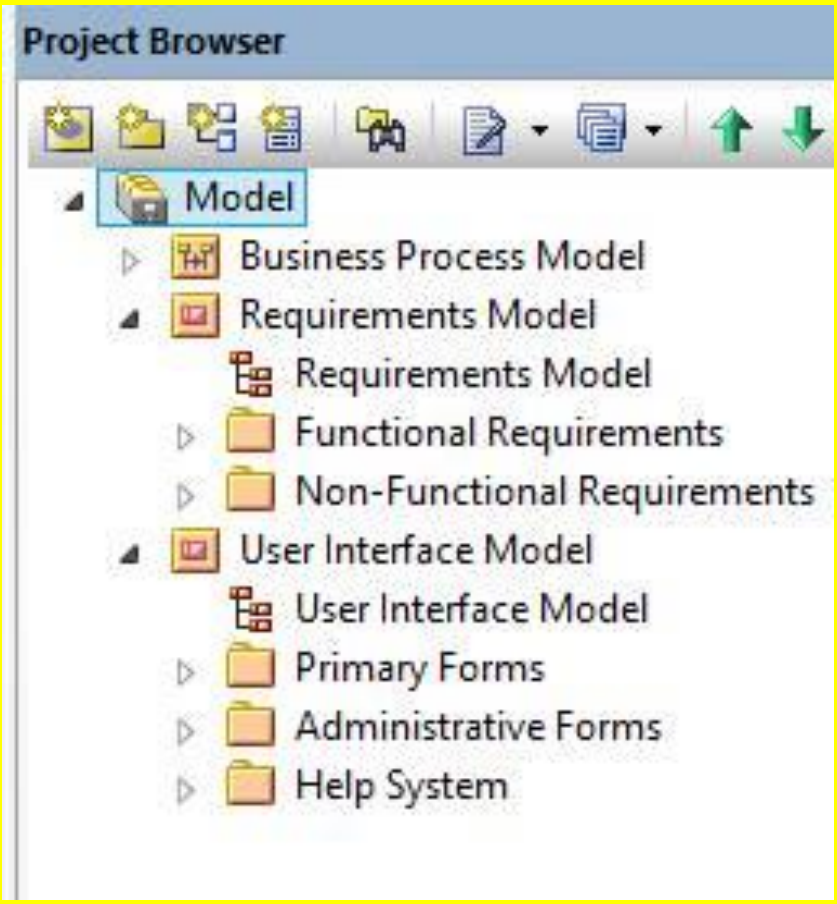
Requirements Modeling, part of EA Modeling



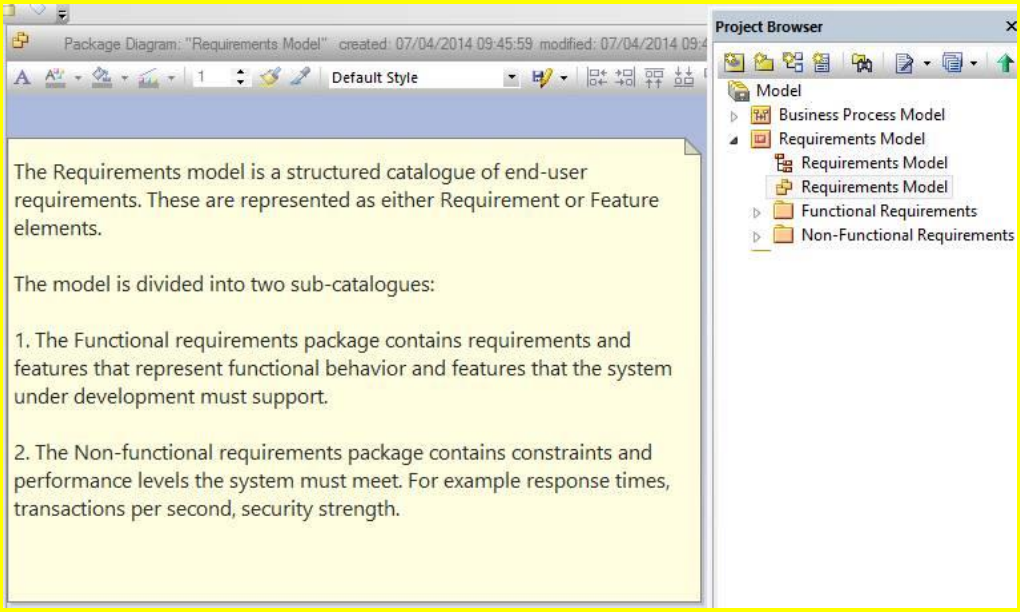
Eliciting from Stakeholders, inferring from others



Requirement as a Modeling Element

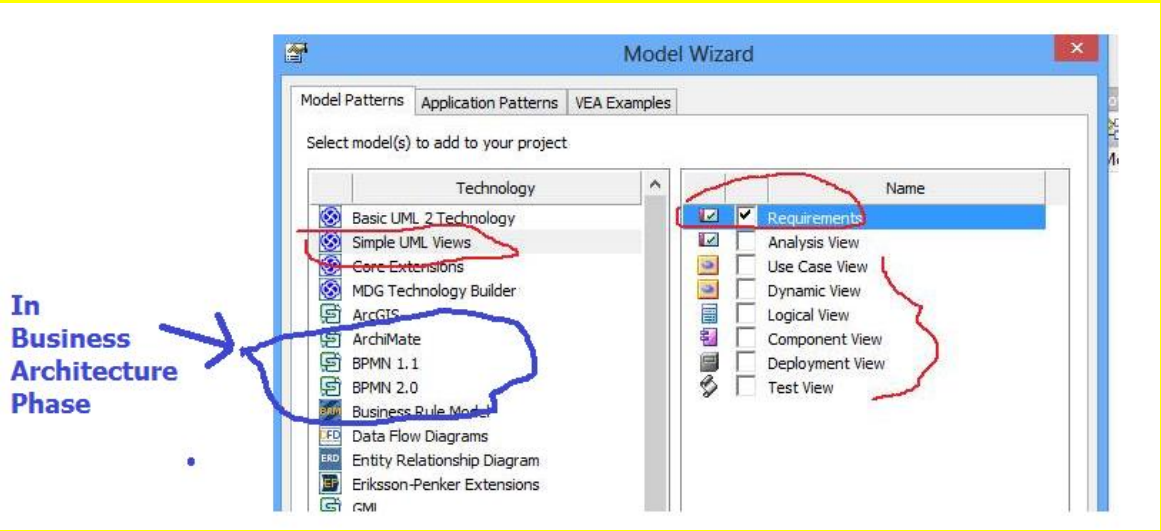


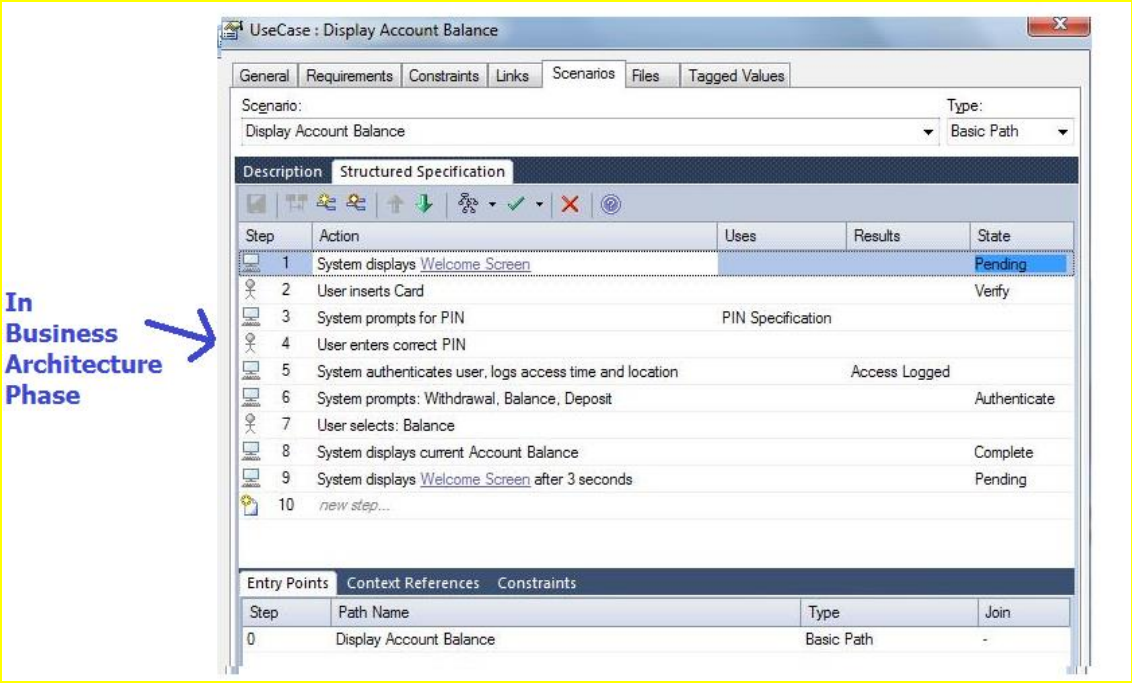
Relating Requirements to Business Processes



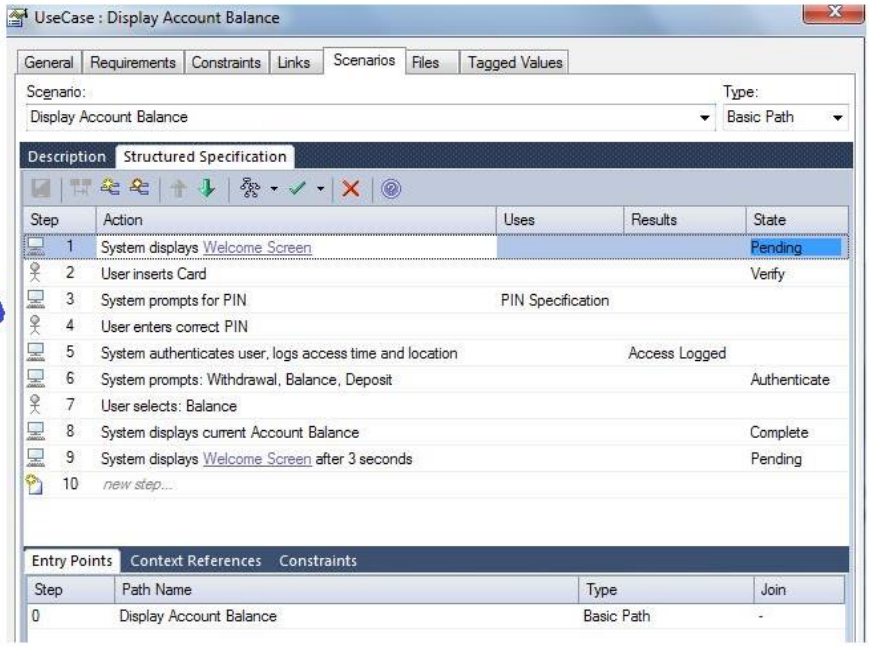
NFRs and Functional Requirements

You can think of Business Process Diagrams and Use Case Diagrams





In Business Architecture Phase



Sample of Requirement that will apply to many projects including ARS

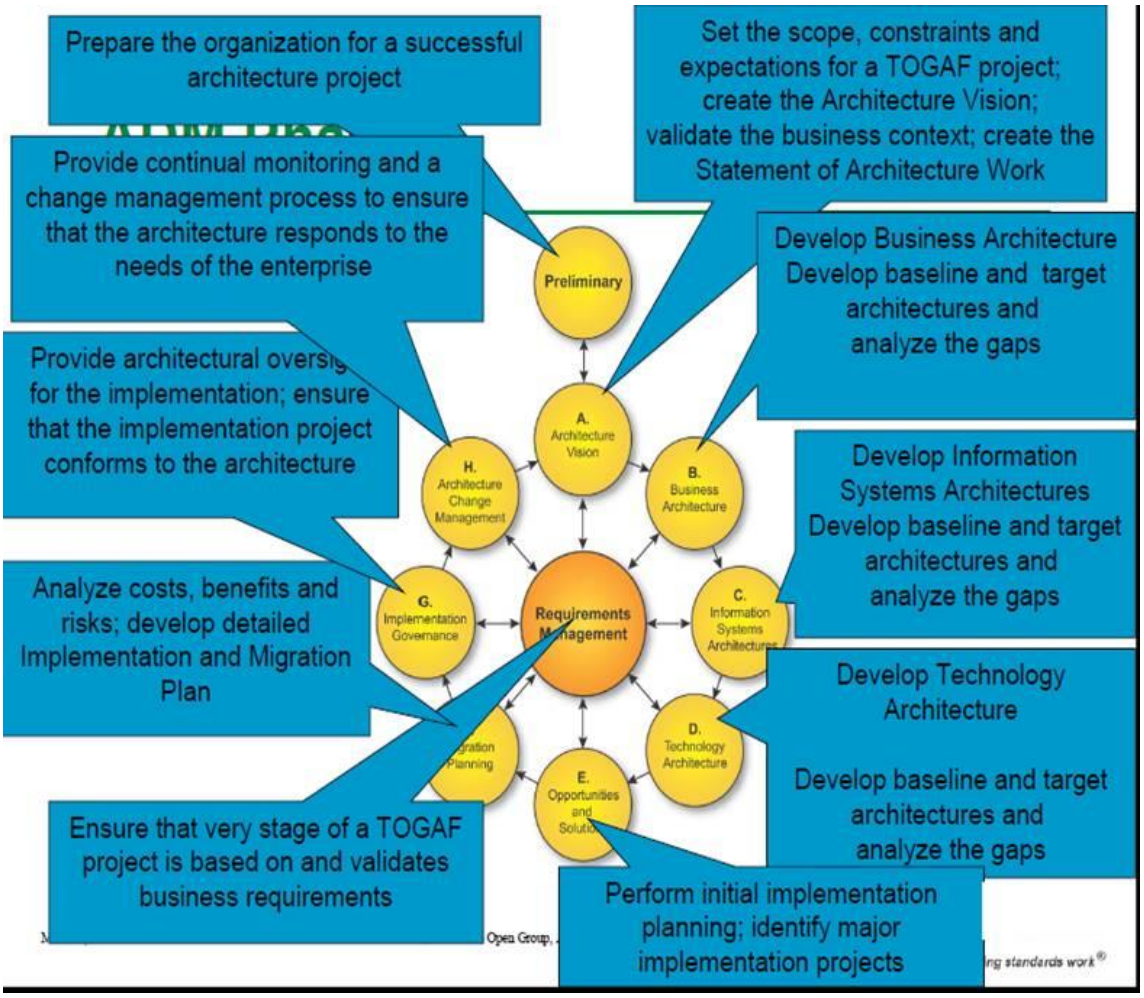
Functional Requirements:

Name	Description	Benefit	Risk	Target Version
Internet access	.....	Critical	Medium	1
Client autonomy	....	High	Low	1
Process automation	....	High	Low	1
Purchase automation	This activity will be automated on the new website by direct connection to the internal payment	Critical	Low	1

	gateway			

Non Functional Requirements

Name	Description	Benefit	Risk	Target Version
Site availability	The site must have an availability rate of 99.4%, that is, less than one hour of unavailability per month	High	Medium	1
Reliability	The IS must operate 24 hours a day, 7 days a week, with a maximum interruption rate of 2/1000; redundancy and hot standby mechanisms must guarantee the continuity of the system	Critical	Low	1



Thus we end this Comprehensive Case Study