Process Components Case Study

Expand All Sections Collapse All Sections

Main Description

Case Study: Process Components in Eclipse Process Framework Composer

Overview

This published site provides an outlook to how Process Components could be realized in Eclipse Process Composer. It currently presents one Process Component called Define Scope.

To learn more about component, please review first the following documents:

- Concept: SPEM 2.0 Process Component explaining how process components are defined in SPEM 2.0
- Concept: Convansys Process Component explaining how Convasys defined and used components in the past

Case Study Scope

This case study current realizes one component using RUP content that has been donated by Convasys. I re-modeled the original Convasys component using the SPEM 2.0 concepts and created this mock-up site.

Here as summary of the key changes I did when modeling the Define Scope component based on an example from Convasys:

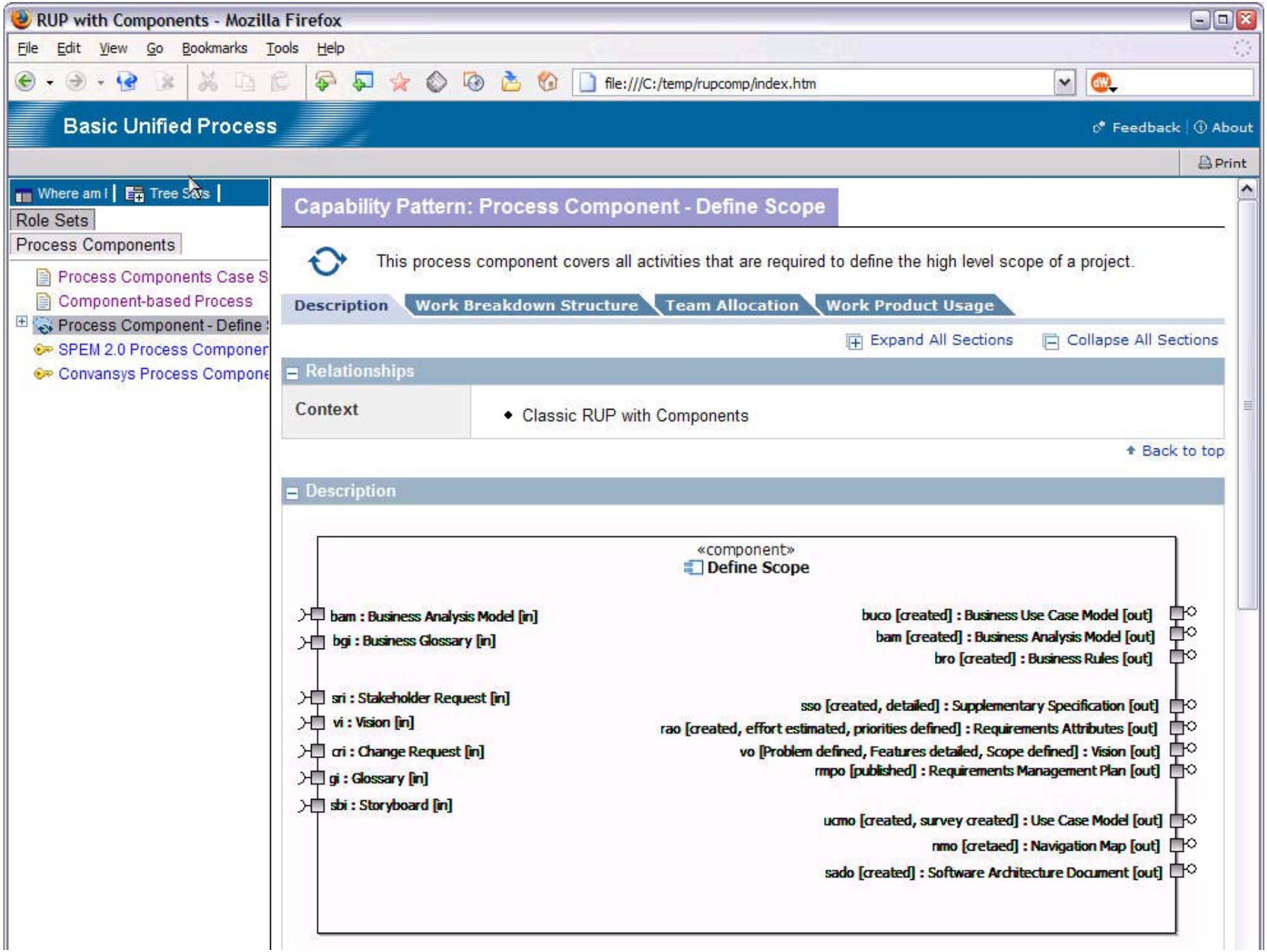
- I did not use the discipline grouping as some activities contained work from other disciplines (e.g. prototype UI). Although the results of that work supports the quality of the requirements discipline it is not requirements work. I also believe that breakdown structures should focus on a breakdown of work and not introduce artificial summary tasks/activity for grouping especially if we are going to create cross-discipline activities in OpenUP.
- I used task from RUP701 and therefore they might look a little bit different than Kirti's original.
- Kirti did not provide me with a black box component diagram for Manage Scope, so I had to do some guesswork and derived what is input and what is output myself by looking at the RUP tasks involved and what makes sense.
- I updated all task descriptors to only cover the work to be covered by the component, i.e. I changed the input and outputs and selected relevant steps only.
- I changed the exist states a bit from Kirtis, because I thought there was room for improvement. For example "Updated" is not really a state and very ambiguous. Whenever you do a change on a ork product it is updated, but exit states should rather express explicitly what as changed and/or what new state is available on the work product. I therefore chose state names such as "features defined", "use case prioritized" instead of "updated".

Define Scope process component walkthrough

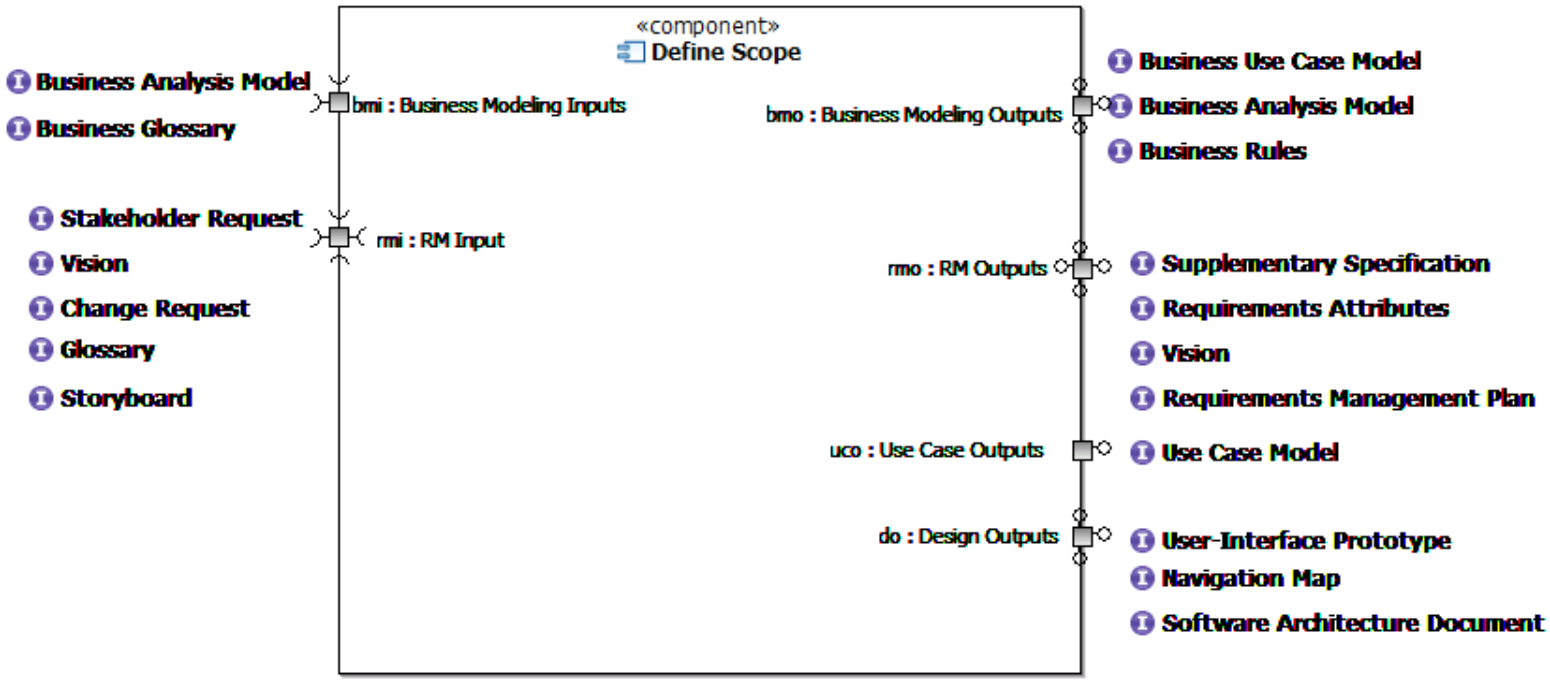
External Process Component view

When loading the published case study site and selecting the process component Define Scope in the tree-browser and then reviewing the component's description tab you will see and overview of the component. The diagram in the Description view shows the external component view listing input and output

ports using the UML 2.0 presentation for components. The output ports specify in their names the state of the work products that get send through that port. Similar states can be define for the inputs, but has not been done in this case study as the Convasys source for this particular component only specified output states.


RUP with Components - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/temp/rupcomp/index.htm
Basic Unified Process
Feedback | About
Print
Where am I | Tree Sets
Role Sets
Process Components
Process Components Case S
Component-based Process
Process Component - Define
SPEM 2.0 Process Componer
Convansys Process Compone
Capability Pattern: Process Component - Define Scope
This process component covers all activities that are required to define the high level scope of a project.
Description
Work Breakdown Structure
Team Allocation
Work Product Usage
Expand All Sections
Collapse All Sections
Relationships
Context
Classic RUP with Components
Back to top
Description
«component»
Define Scope
bam : Business Analysis Model [in]
bgi : Business Glossary [in]
sri : Stakeholder Request [in]
vi : Vision [in]
cri : Change Request [in]
gi : Glossary [in]
sbi : Storyboard [in]
buco [created] : Business Use Case Model [out]
bam [created] : Business Analysis Model [out]
bro [created] : Business Rules [out]
sso [created, detailed] : Supplementary Specification [out]
rao [created, effort estimated, priorities defined] : Requirements Attributes [out]
vo [Problem defined, Features detailed, Scope defined] : Vision [out]
rmpo [published] : Requirements Management Plan [out]
ucmo [created, survey created] : Use Case Model [out]
nmo [cretaed] : Navigation Map [out]
sado [created] : Software Architecture Document [out]

The type of the ports corresponds one to one to work products, which leads to individual ports for each an every work product required by the component as well as produced by the component. An alternative way of using ports could be the following:



This representation makes use of the fact that in UML 2.0 component ports have a type, which can be related to interfaces with a usage dependency or realization relationship. Usage dependencies indicate required interfaces that could be interpreted as inputs and realizations indicate provided interfaces that can be interpreted as outputs. Modeling process component in this way would have the advantage that much less ports need to be specified, but the disadvantage would be that we would need to create interfaces for all work products defined in method content.

Internal Process Component view

When selecting the Work Breakdown Structure in the same page the internal view of the process component is presented, which comprises of the activity diagram and the breakdown structure.

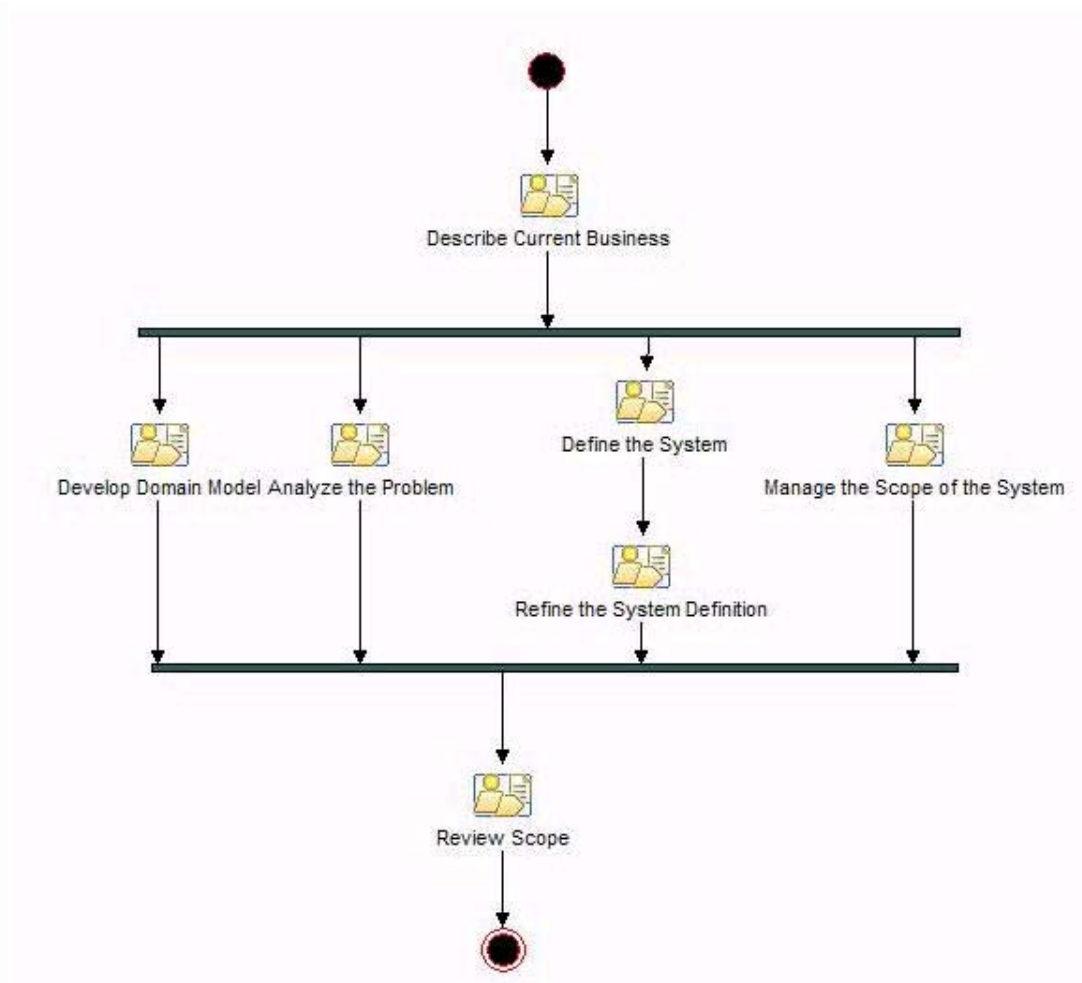
Capability Pattern: Process Component - Define Scope

This process component covers all activities that are required to define the high level scope of a project.

Description | Work Breakdown Structure | Team Allocation | Work Product Usage

Expand All Sections Collaps

Workflow



EPF Composer specific capabilities have been used to model this breakdown in the exact same way as the Convasys source. The different process views such as Team Allocation and Work Product Usage view can now be used to examine the detail of the component. For example the work product states that were visible in the port names for outputs have been specified in the Work Product Usage view as show below. This information can be reviewed in the EPF Composer tool as well as the published site.

Define Scope

Presentation Name	Model Info	Entry State	Exit State	Type	Deliv
Process Component - Define Scope				Capability Pa...	
Describe Current Business				Activity	
Develop Domain Model				Activity	
Analyze the Problem				Activity	
Define the System				Activity	
Stakeholder Requests	Mandatory Input,Optional Input			Artifact Desc...	
Business Glossary	Optional Input			Artifact Desc...	
Business Use Case Model	Optional Input			Artifact Desc...	
Business Analysis Model	Optional Input			Artifact Desc...	
Business Rule	Optional Input			Artifact Desc...	
Storyboard	Optional Input			Artifact Desc...	
Requirements Management Plan	Optional Input			Artifact Desc...	
Vision	Optional Input,Output		Features detailed	Artifact Desc...	
Use-Case Model	Optional Input,Output		Survey created	Artifact Desc...	
Glossary	Optional Input,Output		Contains solution terms	Artifact Desc...	
Supplementary Specifications	Optional Input,Output		Created	Artifact Desc...	
Requirements Attributes	Output		Effort estimated	Artifact Desc...	
Manage the Scope of the System				Activity	
Use-Case Model	Mandatory Input			Artifact Desc...	
Stakeholder Requests	Optional Input			Artifact Desc...	
Change Request	Optional Input			Artifact Desc...	
Risk List	Mandatory Input			Artifact Desc...	
Software Architecture Document	Mandatory Input,Output		Created	Artifact Desc...	
Requirements Management Plan	Mandatory Input,Output		Updated	Artifact Desc...	
Vision	Optional Input,Output		Scope defined	Artifact Desc...	
Requirements Attributes	Optional Input,Output		Priorities defined	Artifact Desc...	
Software Requirement	Optional Input,Output		Prioritized	Artifact Desc...	
Refine the System Definition				Activity	
Review Scope				Activity	

In addition to presenting work products and their states activity by activity, EPF Composer can also create so-called roll-ups that combine the work products from lower breakdown levels and summarize them for a higher breakdown level. Such roll-ups can be used as a bill of materials for the component showing all participating work products. For example, the image below shows now all the work products used and produced by this process components, if they are used as input and/or outputs by the underlying activities, as well as a summary of all the states the work product go through inside the component.

Define Scope

Presentation Name	Model Info	Entry State	Exit State	Type
Process Component - Define Scope				Capability Pattern
Business Analysis Model	Optional Input,Output		Created	Artifact Descriptor
Business Glossary	Optional Input			Artifact Descriptor
Business Rule	Output		Created	Artifact Descriptor
Business Use Case Model	Output		Created	Artifact Descriptor
Business Vision	Mandatory Input			Artifact Descriptor
Change Request	Optional Input			Artifact Descriptor
Glossary	Optional Input,Output		Created,Contains solution terms	Artifact Descriptor
Iteration Plan	Mandatory Input			Artifact Descriptor
Navigation Map	Mandatory Input,Output		Created	Artifact Descriptor
Project-Specific Guidelines	Optional Input			Artifact Descriptor
Requirements Attributes	Output		Created,Effort estimated,Priorities defined,Priorities defined	Artifact Descriptor
Requirements Management Plan	Optional Input,Output		Published,Updated	Artifact Descriptor
Review Record	Output		Created	Artifact Descriptor
Risk List	Mandatory Input			Artifact Descriptor
Software Architecture Document	Mandatory Input,Output		Created	Artifact Descriptor
Software Requirement	Optional Input,Output		Prioritized,Detailed	Artifact Descriptor
Software Requirements Specification	Output		Detailed	Artifact Descriptor
Stakeholder Requests	Mandatory Input,Optional Input			Artifact Descriptor
Storyboard	Optional Input			Artifact Descriptor
Supplementary Specifications	Optional Input,Output		Created,Detailed	Artifact Descriptor
Use-Case Model	Optional Input,Output		Created,Survey created	Artifact Descriptor
User-Interface Prototype	Optional Input,Output		Created	Artifact Descriptor
Vision	Optional Input,Output		Problem defined,Features detailed,Scope defined	Artifact Descriptor

For example, the Vision document goes through three different states, because it is manipulated in three different activities in the component. As one can see one image above the "Define the System" activity establishes the state "Features Detailed" and the "Manage the Scope of the System" activity establishes the state "Scope defined". The roll-up displayed above shows the combination of all of these states.

When specifying ports for this component the end-user would need to manually select now which of these states will be the final states of the work products that pass through the ports. This information cannot be automatically be detected as many components such as this one would define parallel work. The EPF Component model either would need to list all states a work product passed through in the components or an end-user needs to make the selection, which the final state should be.

EPF Composer provides other views on the component in the tool or published Web-site such as a role overview presenting a list of all roles involved in the component or a so-called consolidated view that combines all the information modeled for the component.

Capability Pattern: Process Component - Define Scope

This process component covers all activities that are required to define the high level scope of a project.

Description | Work Breakdown Structure | Team Allocation | Work Product Usage

Expand All Sections Collapse All Sections

Team Breakdown

Prefix	Breakdown Element	Model Info	Type
	Business-Process Analyst		Role Descriptor
	Requirements Specifier		Role Descriptor
	Actor	Responsible For	Work Product Descriptor
	Use Case	Responsible For	Work Product Descriptor
	Develop Supplementary Specifications	Performs Additionally	Task Descriptor
	Software Architect		Role Descriptor
	System Analyst		Role Descriptor
	Requirements Management Plan	Responsible For	Work Product Descriptor
	Storyboard	Responsible For	Work Product Descriptor
	Requirements Attributes	Responsible For	Work Product Descriptor
	Glossary	Responsible For	Work Product Descriptor
	Use-Case Model	Responsible For	Work Product Descriptor
	Stakeholder Requests	Responsible For	Work Product Descriptor
	Vision	Responsible For	Work Product Descriptor
	Requirements Management Plan	Modifies	Work Product Descriptor
	Vision	Modifies	Work Product Descriptor
	Requirements Attributes	Modifies	Work Product Descriptor
	Glossary	Modifies	Work Product Descriptor
	Use-Case Model	Modifies	Work Product Descriptor
	Develop Requirements Management Plan	Performs As Owner	Task Descriptor
	Develop Vision	Performs As Owner	Task Descriptor
	Capture a Common Vocabulary	Performs As Owner	Task Descriptor
	Find Actors and Use Cases	Performs As Owner	Task Descriptor
	Technical Reviewer		Role Descriptor
	User-Interface Designer		Role Descriptor

Back to top

The image above shows the Team Allocation view from the published Web-site that shows all roles participating in the component, plus the work products they are responsible for and the task they will perform. this view allows team member that are assigned to play any of these roles in the component with a quick overview of what they key information relevant to them.

Define Scope

Presentation Name	Index	Model Info	Type	Predecessors	Repea...	Ongoing	Event-...	Optional	Planned
Process Component - Define Scope	0		Capability Pattern		false	false	false	false	true
Describe Current Business	1		Activity		false	false	false	false	true
Find Business Actors and Use Cases	2		Task Descriptor		false	false	false	false	false
Business-Process Analyst		Primary Performer	Role Descriptor					false	true
Business Vision		Mandatory Input	Artifact Descriptor					false	true
Project-Specific Guidelines		Optional Input	Artifact Descriptor					false	true
Business Use Case Model		Output	Artifact Descriptor					false	true
Develop Domain Model	3		Activity	1	false	false	false	false	true
Maintain Business Rules	4		Task Descriptor		false	false	false	false	false
Business-Process Analyst		Primary Performer	Role Descriptor					false	true
Business Glossary		Optional Input	Artifact Descriptor					false	true
Business Analysis Model		Optional Input	Artifact Descriptor					false	true
Business Rule		Output	Artifact Descriptor					false	true
Business Domain Model Analysis	5		Task Descriptor		false	false	false	false	false
Business-Process Analyst		Secondary Performer	Role Descriptor					false	true
Business Analysis Model		Optional Input	Artifact Descriptor					false	true
Business Analysis Model		Output	Artifact Descriptor					false	true
Analyze the Problem	6		Activity	1	false	false	false	false	false
Develop Vision	7		Task Descriptor		false	false	false	false	false
Find Actors and Use Cases	8		Task Descriptor		false	false	false	false	false
Capture a Common Vocabulary	9		Task Descriptor		false	false	false	false	false
Develop Requirements Management Plan	10		Task Descriptor		false	false	false	false	false
Define the System	11		Activity	1	false	false	false	false	true
Find Actors and Use Cases	12		Task Descriptor		false	false	false	false	false
Develop Vision	13		Task Descriptor		false	false	false	false	false
Develop Supplementary Specifications	14		Task Descriptor		false	false	false	false	false
Capture a Common Vocabulary	15		Task Descriptor		false	false	false	false	false
Manage the Scope of the System	16		Activity	1	false	false	false	false	true
Prioritize Use Cases	17		Task Descriptor		false	false	false	false	false
Manage Dependencies	18		Task Descriptor		false	false	false	false	false
Refine the System Definition	19		Activity	11	false	false	false	false	true
Detail a Use Case	20		Task Descriptor		false	false	false	false	false
Detail the Software Requirements	21		Task Descriptor		false	false	false	false	false
Develop Supplementary Specifications	22		Task Descriptor		false	false	false	false	false
Design the User Interface	23		Task Descriptor		false	false	false	false	false
Prototype the User-Interface	24		Task Descriptor		false	false	false	false	false
Review Scope	25		Activity	19,6,3,16	false	false	false	false	true
Review Requirements	26		Task Descriptor		false	false	false	false	false

Finally the image show above presents the Consolidated View showing the complete breakdown of work, roles performing tasks as well as the input/output for the tasks.

Back to top