

LNCs 9786

Osvaldo Gervasi · Beniamino Murgante
Sanjay Misra · Ana Maria A.C. Rocha
Carmelo M. Torre · David Taniar
Bernady O. Apduhan · Elena Stankova
Shangguang Wang (Eds.)

Computational Science and Its Applications – ICCSA 2016

16th International Conference
Beijing, China, July 4–7, 2016
Proceedings, Part I

1
Part I



 Springer

Business Modeling and Requirements in RUP: A Dependency Analysis of Activities, Tasks and Work Products

Carina Campos

Dept. de Sistemas de Informação
Universidade do Minho
Guimarães, Portugal
carina.campos13@gmail.com

José Eduardo Fernandes

Polytechnic Institute of Bragança
Bragança, Portugal
jef@ipb.pt

Ricardo J. Machado

Centro ALGORITMI,
Escola de Engenharia
Universidade do Minho
Guimarães, Portugal
rmac@dsi.uminho.pt

Abstract. Most artifacts developed during the requirements engineering process relate themselves in different ways. In order to understand in detail how they affect each other during the software development process, it is relevant to identify their interdependencies. This paper presents a systematization of the existing interdependencies between the different elements of the Rational Unified Process (RUP) in the Business Modeling and Requirements disciplines. This work, which highlights knowledge about the different interdependencies and traceability of RUP elements, is useful to avoid unconscious decisions during software the development process and also, to detect potential problems due to the violation of the existing interdependencies.

Keywords: *requirements engineering, interdependencies, traceability, artifacts, RUP*

I. INTRODUCTION

Most individual requirements developed during the requirements engineering process relate to and affect each other in different ways and thus cannot be treated in isolation [1, 2]. The fact that the requirements relate to and affect each other makes it necessary to identify and manage the requirements interdependencies in order to avoid potentially costly mistakes during the system development.

Requirements interdependencies are not a problem by themselves, but they influence the number of development activities and decisions made during the software engineering process [3]. Traceability is the basis for studying the requirements interdependencies during the development process [4] since it allows identifying and justifying the artifacts that implement the requirements initially formalized.

Software development produces various kinds of artifacts. The artifacts, such as requirements, do not exist in isolation; instead they relate to and affect each other [5]. During the development of solutions and also during the exploration phase for maintenance issues, frequently arises the need to introduce several changes to the project decisions previously established. These changes should be clearly identified to ensure the complete identification of the artifacts involved in the changes.

To this end, it is necessary to have knowledge about how the different artifacts relate among them since it facilitates the identification of the artifacts affected.

RUP is a process that provides the best practices and guidelines for successful software development [6]. This work, in the context of Business Modeling and Requirements disciplines of RUP, analyzes and systematizes the traceability and the interdependencies that may occur between the various elements during software development projects.

This paper has the following structure: section 2 presents the importance of dealing with the interdependencies and the traceability during the software development; section 3 describes the interdependencies and the traceability between the different elements of the RUP; section 4 presents the conclusions.

II. INTERDEPENDENCIES AND TRACEABILITY

Dahlstedt and Persson [4] refer to traceability as "a basis for addressing the requirements interdependencies". According to Genvigir [7] and Zou, Settimi and Cleland-Huang [8], traceability is intimately associated to the software production process, specifically to the requirements and to the ability to establish links between these requirements and other artifacts that satisfy them.

Sánchez, Alonso, Rosique, Álvarez and Pastor [9] mention that the requirements traceability aims to help determine the impact of changes in the conception phase of software, to support their integration, preserve the knowledge and assure the quality and correction of the global system.

Requirements traceability is as a quality factor [3, 10, 11, 12]. Actively supporting traceability in a software development project can help ensuring other qualities of software, such as adequacy and understandability [11]. On the other hand, neglecting the traceability can lead to less maintainable software and to failures due to inconsistencies and omissions [11]. Dörmges and Pohl [13] refer to neglecting the traceability or capture insufficient and/or unstructured traces leads to a

decrease in system quality, causes revisions, and hence, increases project costs and time.

Aizenbud-Reshef, Nolan, Rubin and Shaham-Gafni [14] refer that, from the perspective of requirements management, traceability facilitates the interconnection of requirements to their origins and reasons. Additionally, it allows capturing the information needed for understanding the evolution of requirements and for verification of requirements fulfillment. Complete traceability allows calculate more accurately the costs, as well as to determine lists of changes, without depending on the programmer knowledge of all the areas that these changes affect [14]. All these reasons make crucial to implement traceability practices throughout the software development.

It is essential to identify and manage the interdependencies that occur throughout the system development in order to, if needed, in any context, to properly consider related artifacts and as such, to avoid potentially costly mistakes by neglecting either those relations or eventually relevant artifacts. As mentioned earlier, through traceability, it is possible to manage these interdependencies; hence, traceability is fundamental to the development process.

The purpose of dealing, systematically, with requirements interdependencies improves the decisions made during software development as well as to detect the potential problems that may arise because of the requirements interdependencies [3]. Managing requirements interdependencies consists in identify, store and maintain information about how the requirements relate to and affect each other [3].

Maintain traceability of the requirements interdependencies is essential in order to support various situations and activities in the system development process [4]. Traceability should be included and treated along the development projects, thus representing, an asset to their success. Knowing the whole story of the artifacts, as well as their interdependencies, will enable easier identification and management of existing interdependencies from the early stages of development. Therefore, this knowledge minimizes problems that may arise during the software development process.

III. INTERDEPENDENCIES AND TRACEABILITY IN RUP

RUP aims to ensure the production of quality software that meets the needs and expectations of its users in a predictable schedule and cost [6]. RUP guidelines entail several elements such as activities, tasks, roles and work products. Throughout the development process, at several moments, RUP elements become interconnected; by this way, a simple change in an element causes various subsequent adjustments in others. Therefore, the knowledge of existing interdependencies between the various elements is particularly useful since it allows easier identification of elements affected during a change.

As Dahlstedt and Persson [4] refer, is essential to maintain the traceability of interdependencies since it allows to know, in detail, how the elements relate, as well as to support various situations and activities in the software development. Through the traceability of various elements of RUP, it is possible to easier identify and manage the interdependencies that may occur between elements. For those practitioners that adopt RUP

guidelines, it is useful to understand the interdependencies that may exist between the various RUP elements.

A. Dependency Analysis of Activities and Tasks

RUP is organized in various disciplines and phases. However, the study mentioned in this paper focuses in two transitions (see Fig. 1): (1) from the Business Modeling discipline to the Requirements discipline, at Inception phase; (2) from the Inception to the Elaboration phase, within the Requirements discipline.

To facilitate an overview and analysis of all tasks of RUP (for Business Modeling and Requirements disciplines), the conduction of an initial RUP review allowed the construction of information presented in TABLE I and TABLE II.

These tables show the different tasks of the disciplines of Business Modeling and Requirements, the activities associated with these tasks, the phase where they are performed, and the roles responsible for them. TABLE I details the activities, tasks, phases and roles of the Business Modeling discipline considering both processes of Classic RUP Lifecycle and Business Modeling Lifecycle.

The column Activities presents the five activities performed in this discipline. The activities performed in the Classic RUP Lifecycle process are signaled in the table by an α , the activities performed in the Business Modeling Lifecycle process are signaled by an β and the activities performed in both processes are signaled in the table by $\alpha\beta$.

Column Tasks exposes all tasks practiced in the Business Modeling discipline. Only the tasks with a gray background were studied, since the other stand in phases that are outside the scope of our study. The intersection of column Activities with the lines of column Tasks indicates (through an 'x') the tasks included in the activities.

Column Phases presents which phases include the different tasks and activities. The abbreviations B1, B2, B3, B4 and B5 (for the various activities) associate tasks and their activities to the several phases. Column Role main refers which are the roles responsible for performing the different tasks. The activities with blue background (Assess Business Status, Describe Current Business and Develop Domain Model) and the phase (Inception) refer to the activities and the phase studied in Business Modeling discipline.

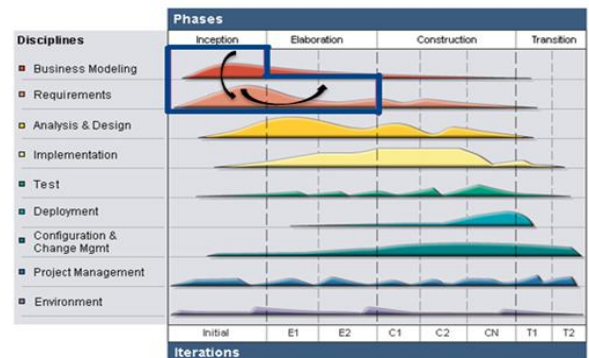


Fig. 1. Positioning of the study in the RUP (Based on [15])

TABLE I. ACTIVITIES, TASKS, PHASES AND ROLES OF THE BUSINESS MODELING DISCIPLINE

	Activities					Tasks	Phases				Role main
	B1 Assess Business Status $\alpha\beta$	B2 Describe Current Business β	B3 Define Business β	B4 Explore Process Automation β	B5 Develop Domain Model α		Inception	Elaboration	Construction	Transition	
Business Modeling	x					Assess Target Organization - ATO	B1				Business-Process Analyst
	x	x	x		x	Business Architectural Analysis - BAA	B1, B2, B5	B2, B3	B3	B3	Business Architect
			x			Business Operation Analysis		B3		B3	Business Designer
			x			Business Operation Design		B3	B3	B3	Business Designer
			x			Business Use-Case Analysis		B3	B3	B3	Business Designer
	x	x	x		x	Capture a Common Business Vocabulary - CCBV	B1, B2, B5	B2, B3	B3		Business-Process Analyst
				x		Construct Business Architectural Proof-of-Concept		B4			Business Architect
				x		Define Automation Requirements		B4	B4	B4	Business Designer
			x			Define Business System Context		B3			Business-Process Analyst
			x		x	Detail a Business Entity - DBE	B5		B3	B3	Business Designer
			x			Detail a Business Use Case			B3		Business Designer
			x			Detail a Business Worker			B3	B3	Business Designer
		x	x			Find Business Actors and Use Cases - FBAUC	B2	B2, B3	B3		Business-Process Analyst
	x	x	x			Identify Business Goals - IBG	B1, B2	B3			Business-Process Analyst
	x	x	x		x	Maintain Business Rules - MBR	B1, B2, B5	B2, B3	B3	B3	Business-Process Analyst
			x			Prioritize Business Use Cases		B3	B3		Business Architect
			x		x	Review the Business Analysis Model - RBAM	B5	B3	B3	B3	Technical Reviewer
			x			Review the Business Use-Case Model		B3	B3		Technical Reviewer
	x	x	x			Set and Adjust Objectives - SAO	B1, B2	B3			Business-Process Analyst
			x			Structure the Business Use-Case Model		B3	B3		Business-Process Analyst
							Activities				
							Classic RUP Lifecycle				α
							Business Modeling Lifecycle				β
							Classic RUP Lifecycle + Business Modeling Lifecycle				$\alpha\beta$

TABLE II. ACTIVITIES, TASKS, PHASES AND ROLES OF THE REQUIREMENTS DISCIPLINE

	Activities						Tasks	Phases				Role main
	R1 Analyze the Problem $\alpha\beta$	R2 Understand Stakeholder Needs $\alpha\beta$	R3 Define the System α	R4 Manage the Scope of the System α	R5 Refine the System Definition α	R6 Manage Changing Requirements α		Inception	Elaboration	Construction	Transition	
Requirements	x	x	x				Capture a Common Vocabulary - CCV	R1 α , R2 α , R3			R1 β , R2 β	System Analyst
					x		Detail a Use Case - DUC		R5			Requirements Specifier
					x		Detail the Software Requirements - DSR		R5			Requirements Specifier
	x						Develop Requirements Management Plan - DRMP	R1 α				System Analyst
		x	x		x		Develop Supplementary Specifications - DSS	R2 α , R3	R5		R2 β	System Analyst
	x	x	x	x			Develop Vision - DV	R1 α , R2 α , R3, R4			R1 β , R2 β	System Analyst
		x					Elicit Stakeholder Requests - ESR	R2 α			R2 β	System Analyst
	x	x	x				Find Actors and Use Cases - FAUC	R1 α , R2 α , R3			R1 β , R2 β	System Analyst
		x	x	x		x	Manage Dependencies - MDep	R2 α , R3, R4	R6	R6	R6	System Analyst
				x			Prioritize Use Cases - PUC	R4				Software Architect
						x	Review Requirements - RReq		R6	R6	R6	Technical Reviewer
						x	Structure the Use-Case Model - SUCM		R6	R6	R6	System Analyst
							Activities/ Phases					
							Classic RUP Lifecycle				α	
							Business Modeling Lifecycle				β	
							Classic RUP Lifecycle + Business Modeling Lifecycle				$\alpha\beta$	

TABLE II presents the activities, tasks, phases and roles of the Requirements discipline.

Column Activities presents the six activities practiced in this discipline. As before, in the table, α signals activities performed in the Classic RUP Lifecycle process, β signals activities executed in the Business Modeling Lifecycle process, and $\alpha\beta$ signals the activities performed in both processes.

Column Tasks exposes all tasks practiced in the Requirements discipline. In this discipline, all tasks have a gray background since they are in the phases of the scope of this study and as such, covered by this study. An 'x' at the intersection of column Activities with the lines of column Tasks indicates the tasks practiced in the activities.

Column Phases presents tasks and activities performed in the different phases. Abbreviations R1, R2, R3, R4, R5 and R6 (for the various activities) associate the tasks and their activities to phases where they are performed. In the intersections, we use α for Classic RUP Lifecycle, β for Business Modeling Lifecycle and $\alpha\beta$ for both processes. The intersections show the process where the tasks and their activities are performed.

As in the previous table, the column Role main refers to the roles responsible for performing the different tasks.

The activities with blue background (Analyze the Problem, Understand Stakeholder Needs, Define the System, Manage the Scope of the System, Refine the System Definition and Manage Changing Requirements) and the phases (Inception and Elaboration) refer to the activities and the phases studied in Requirements discipline.

The information provided in these tables is useful throughout the software development because it allows to perceive how activities, tasks, and roles relate in a particular discipline and phase.

B. Dependency Analysis of Work Products and Tasks

The elaboration of the previous two tables allowed to perceive the tasks and activities covered in the disciplines and phases considered in this study. TABLE III and TABLE IV have the purpose of clarifying the interconnection of all work products of both disciplines to their respective tasks.

The first column of TABLE III shows all the work products of the Business Modeling discipline and the second column presents all the tasks. Only the tasks with a gray background were analyzed because the other tasks are in phases that are not within the scope our study.

The intersection of these two columns depicts the work products consumed and produced in the various tasks. These intersections use the terms IN, OUT and I/O: the term IN is used to refer work products consumed by the associated task; the term OUT represents the work products produced by the task in question; the term I/O represents that the work products are both consumed and produced by the task in question. Besides these terms, the term IN* refers to work products that are an optional entry of the associated task; these work products are not necessarily consumed in the task. In the tables, the use of colors

facilitate the identification of terms IN, OUT, and I/O: the term IN is represented by the green color, the term OUT by the red color and the term I/O by the yellow color.

The first column, TABLE IV shows all the work products of the Requirements discipline and in the second column presents all the tasks. All the tasks of this discipline were analyzed because all the tasks are in phases that are within the scope our study. The intersection of these two columns depicts which work products are consumed and produced in the various associated tasks. These intersections use the terms IN, OUT and I/O, which were previously defined.

TABLE III and TABLE IV show the work products produced and consumed by the different tasks. The information available in these tables allows the identification of existing interdependencies between tasks and work products that are produced and consumed.

Fig. 2 and Fig. 3 present two graphical representations that were developed to enhance the perception of the information contained in the previous tables; i.e., all the existing interdependencies between the activities and the tasks and work products of a given phase and discipline. This visualization facilitates the analysis of the existing interdependencies along the development process, thus allowing for a better understanding and management.

The representation of the TABLE I and TABLE III. This representation refers to the Business Modeling discipline in the Inception phase. It presents the five activities belonging to this discipline. These activities interconnect to their tasks; two of these activities have no associated tasks because they are outside the scope of this study. Each task has its associated work products. These work products may be consumed in the associated task (inputs, graphically represented by arrow green) or may be produced by that task (outputs, graphically represented by arrow red). The work products represented in yellow refers to work products belonging to the Business Modeling discipline.

The work products, represented in orange, despite being work products produced and consumed in this discipline/phase, do not belong directly to work products defined by RUP for this discipline. For these work products (in orange), a description below them indicates the discipline and the phase to which they belong; some of those do not have associated discipline because, in concrete, they do not belong to any.

The representation of the Fig.2 is based on information gathered in the TABLE II and TABLE IV. This representation refers to the Requirements discipline in the Inception and Elaboration phases. It presents the six activities belonging to this discipline, as well as its interconnected tasks. All these activities have associated tasks because all of them are within the scope of this study. Each task has its associated work products. These work products may be consumed in the associated task (inputs, graphically represented by arrow green) or may be produced by that same task (outputs, graphically represented by arrow red). The work products represented in yellow refers to work products belonging to the Requirements discipline.

TABLE III. WORK PRODUCTS OF THE TASKS OF THE BUSINESS MODELING DISCIPLINE

Discipline	Business Modeling																
Tasks	Assess Target Organization - ATO	Business Architectural Analysis - BAA	Business Operation Analysis	Business Operation Design	Business Use-Case Analysis	Capture a Common Business Vocabulary - CCBV	Construct Business Architectural Proof-of-Concept	Define Automation Requirements	Define Business System Context	Detail a Business Entity - DBE	Detail a Business Use Case	Detail a Business Worker	Find Business Actors and Use Cases - FBAUC	Identify Business Goals - IBG	Maintain Business Rules - MBR	Prioritize Business Use Cases	Review the Business Analysis Model - RBAM
Work Products																	
Business Analysis Model - BAM	OUT	I/O	IN	OUT					I/O								IN
Business Architectural Proof-of-Concept - BAP-C	IN *				IN *	OUT											
Business Architecture Document - BAD	OUT				IN	IN										I/O	
Business Deployment Model - BDM	OUT			OUT			IN										
Business Design Model - BDesM	OUT			OUT			IN										
Business Glossary - BGI						OUT	IN										
Business Goal - BG										IN				OUT			
Business Rule - BR															OUT		
Business Use Case Model - BUCM			I/O	IN	IN				I/O				OUT			I/O	IN
Business Vision - BV		IN				IN							IN	IN	IN	IN	OUT
Supplementary Business Specification - SBS					IN			IN		OUT		OUT					
Target-Organization Assessment - TOA	OUT						IN										IN

OUT	Output
IN	Input
I/O	Input/Output
IN *	The work product is na optional input of the associated task

TABLE IV. WORK PRODUCTS OF THE TASKS OF THE REQUIREMENTS DISCIPLINE

Disciplines	Requirements										
Tasks	Capture a Common Vocabulary - CCV	Detail a Use Case - DUC	Detail the Software Requirements - DSR	Develop Requirements Management Plan - DRMP	Develop Supplementary Specifications - DSS	Develop Vision - DV	Elicit Stakeholder Requests - ESR	Find Actors and Use Cases - FAUC	Manage Dependencies - MDep	Prioritize Use Cases - PUC	Review Requirements - RReq
Work Products											Structure the Use-Case Model - SUCM
Glossary - GI	OUT										OUT
Requirements Attributes - ReqA		OUT	OUT		OUT	OUT	OUT	OUT	OUT	OUT	OUT
Requirements Management Plan - RMP				OUT					I/O		
Software Requirement - SofR			OUT							OUT	IN
Software Requirements Specification - SRS			OUT								
Stakeholder Requests - SR					IN	IN	OUT	IN			
Storyboard - St							OUT				
Supplementary Specifications - SS					OUT						OUT
Use-Case Model - UCM								OUT	IN		I/O
Vision - Vi		IN				OUT			OUT		

OUT	Output
IN	Input
I/O	Input/Output
IN *	The work product is na optional input of the associated task

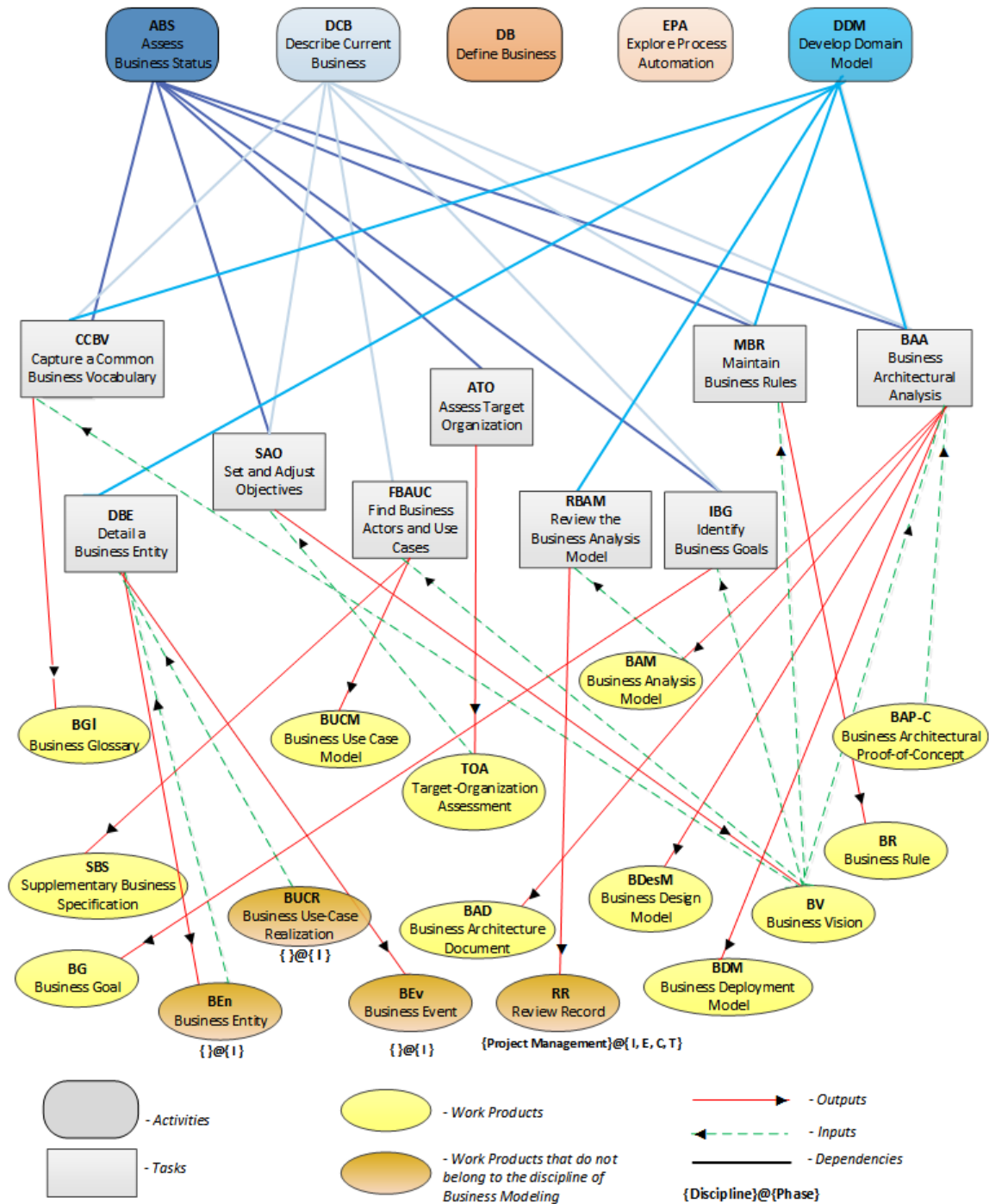


Fig. 2. Scheme Business Modeling@Inception

The work products represented in orange, despite being work products produced and consumed in this discipline|phase, do not belong directly to work products defined by RUP for this discipline. For these work products, represented in orange, a description below them indicates the discipline and the phase to

which they belong; some of them do not have discipline associated because, in concrete, they do not belong to any Fig. 2 depicts two different colored background areas: one represents the Inception phase and the other the Elaboration phase.

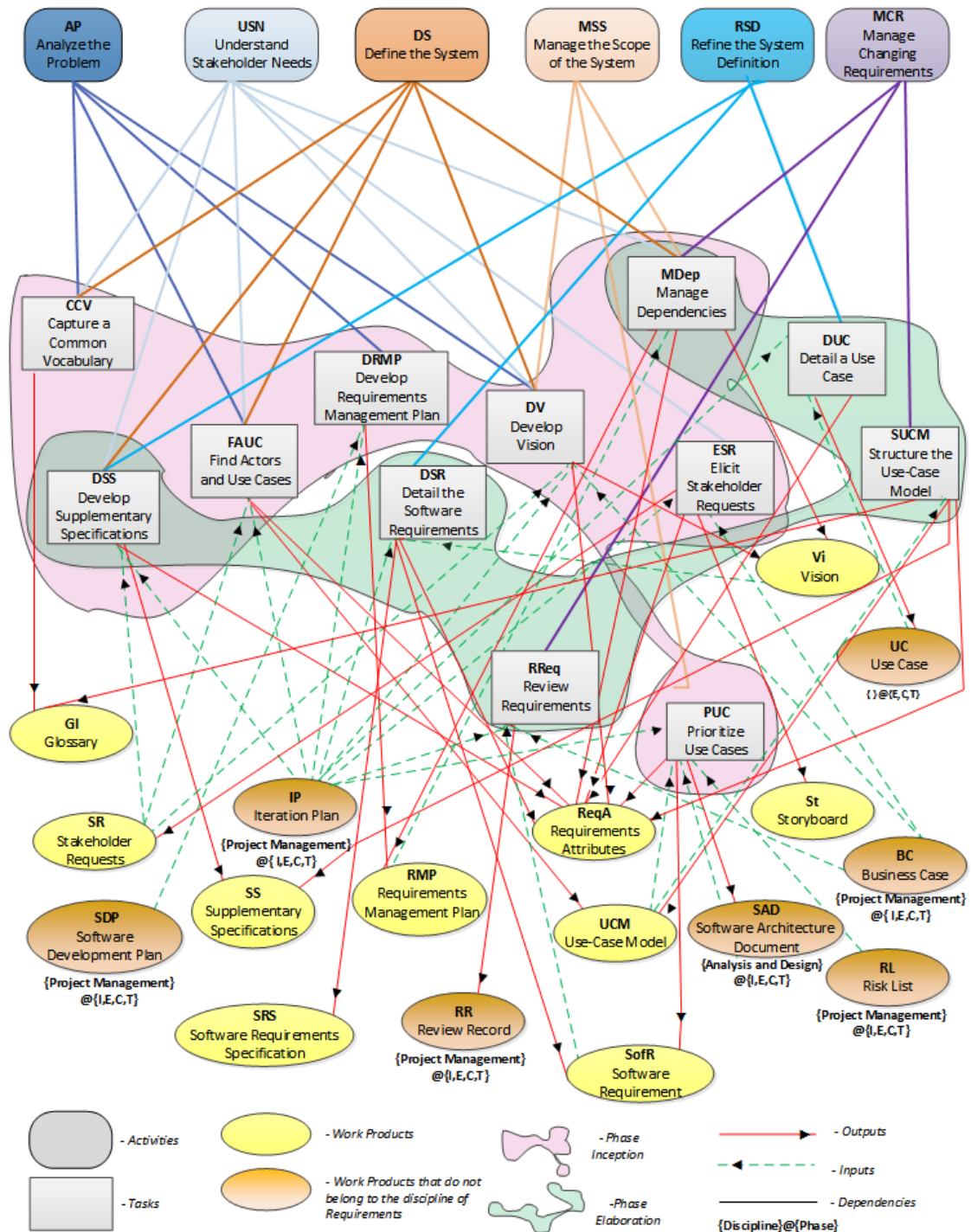


Fig. 3. Scheme Requirements@Inception, Elaboration

These colored background areas allow perceiving that two tasks are handled in both phases, thus verifying that there are interdependencies between the phases. The tables built facilitate the identification of interdependencies, not only among activities, tasks, phases and the roles, but also among tasks and work products, of the disciplines under consideration.

These tables, as well as the graphical representations allow analyzing the traceability of various elements of RUP, as well as easily identifying all the existing interdependencies between those elements. This becomes particularly useful since it allows knowing in detail how the various elements of the RUP process are related.

The information provided in these tables and representations, improve the practitioner's capacity in dealing with the impact of changes and in supporting better development decisions.

This systematization of the interdependencies is also useful to compare a particular method/process model with the RUP since it allows knowing in detail how the RUP is organized. The study of traceability and of the interdependencies between the various elements of the RUP may be extended to all disciplines and phases that compose this process. The expansion of the study will allow detailing how the various elements are related throughout the whole RUP process.

IV. CONCLUSIONS

RUP is a process that provides best practices and guidelines for successful software development. However, this does not provide any information that enables for easy identification of

traceability and existing interdependencies between the various elements that constitute it. Throughout the software development, this can become a problem since there is no explicit native information on RUP documentation on the interrelation of RUP elements.

Our work produced several tables and graphical representations in order to highlight how the various RUP elements are related. These tables and graphical representations allow, from the initial phases of development, an easier identification of the various interdependencies and the traceability among elements, as well as to provide a deeper knowledge about the organization of RUP. This is quite advantageous since it is possible to avoid unconscious decisions during the development process as well as to detect early potential problems due to the existing interdependencies.

ACKNOWLEDGMENT

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

REFERENCES

- [1] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, "An Industrial Survey of Requirements Interdependencies in Software Product Release Planning," Proc. of the Fifth International Symposium on Requirements Engineering, IEEE, 2001, pp. 84-91.
- [2] B. Regnell, B. Paech, A. Aurum, C. Wohlin, A. Dutoit, and J. Natt och Dag, "Requirements Mean Decisions! – Research issues for understanding and supporting decision-making in Requirements Engineering," Proc. First Swedish Conference on Software Engineering Research and Practice (SERP'01), 2001.
- [3] Å. G. Dahlstedt, and A. Persson, "Requirements Interdependencies– State of the Art and Future Challenges," In Aybülke Aurum, & Claes Wohlin, Engineering and Managing Software Requirements, Springer, 2005, pp. 95-116. Springer.
- [4] Å. G. Dahlstedt, and A. Persson, "Requirements Interdependencies - Moulding the State of Research into a Research Agenda," Proc. of the Ninth International Workshop on Requirements Engineering: Foundation for Software Quality, 2003, pp. 55-64.
- [5] M. Heindl, and S. Biffl, "A Case Study on Value-based Requirements Tracing," Proc. of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering, ACM, 2005, pp. 60-69.
- [6] P. Kruchten, "Tutorial: Introduction to the Rational Unified Process," Proc. of the 24th International Conference on Software Engineering (ICSE '02), ACM, 2002, p. 703.
- [7] E.C. Genvigir, Um modelo para rastreabilidade de requisitos de software baseado em generalização de elos e atributos, São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2009.
- [8] X. Zou, R. Settini, and J. Cleland-Huang, "Improving automated requirements trace retrieval: a study of term-based enhancement methods," Empirical Software Engineering, vol. 15, no. 2, 2010, pp. 119-146.
- [9] P. Sánchez, D. Alonso, F. Rosique, B. Álvarez, and J. A. Pastor, "Introducing Safety Requirements Traceability Support in Model-Driven Development of Robotic Applications," IEEE Transactions on Computers, vol.60, no. 8, 2011.
- [10] B. Ramesh and M. Jarke, "Toward Reference Models for Requirements Traceability," IEEE Transactions on Software Engineering, vol. 27, no. 1, 2001, pp. 58-93.
- [11] S. Winkler and J. V. Pilgrim, "A survey of traceability in requirements engineering and model-driven development," Software and Systems Modeling, vol. 9, no. 4, 2010, pp.529-565. Springer.
- [12] G. Spanoudakis and A. Zisman, "Software Traceability: A Roadmap," Handbook of Software Engineering and Knowledge Engineering, vol. III S K Chang, Eds. World Scientific Publishing, 2005, pp. 395-428.
- [13] R. Dörmges and K. Pohl, "Adapting Traceability Environments to Project-Specific Needs," Communications of the ACM, vol. 41, no. 12, 1998, pp. 54-62.
- [14] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, and Y. Shaham-Gafni, "Model traceability," IBM Systems Journal, vol. 45 no.3, 2006.
- [15] IBM, Rational Method Composer, version.7.1.