LOU
RIM

LOUVAIN
RESEARCH INSTITUTE
IN MANAGEMENT
AND ORGANIZATIONS

WORKING PAPER

2019/22

Process Specification of a Governance-Aware Software Development Method

Vu H. A. Nguyen,
Manuel Kolp,
Louvain Research Institute in Management and Organizations

Yves Wautelet,
KULeuven

LOUVAIN RESEARCH INSTITUTE IN MANAGEMENT AND ORGANIZATIONS

Louvain Research Institute in Management and Organizations
Working Paper Series
Editor: Prof. Valérie Swaen
(president-lourim@uclouvain.be)

Process Specification of a Governance-Aware Software Development Method

Vu H. A. Nguyen, Louvain Research Institute in Management and Organizations
Manuel Kolp, Louvain Research Institute in Management and Organizations
Yves Wautelet, KULeuven

Summary

Governing and managing software development in a Requirements-Driven Software Processes such as (GI-) Tropos or even waterfall, V, UP-based ones allow IT managers to propose rules for software processes to cope with stakeholders' requirements and expectations. On the one hand, software development governance has to ensure that software organization business processes meet the strategic requirements of the organization. On the other hand, requirements-driven software methods are development processes using high-level social-oriented models to drive the software life cycle both in terms of project management and deductive iterative engineering techniques. Typically, such methods are well-suited for the inclusion and adaptation of governance and management principles immediately into the software development life cycle. To consolidate both perspectives, this paper proposes a generic process specification using the Software and Systems Process Engineering Metamodel notation allowing to illustrate all disciplines and processes of software development life cycle's governance and management.

Keywords : IT Governance, Software Process, GI-Tropos, SPEM.

Corresponding author :
Vu H. A. Nguyen
CEMIS
Louvain Research Institute in Management and Organizations / Campus Louvain-la-Neuve
Université catholique de Louvain
Place des Doyens 1
B-1348 Louvain-la-Neuve, BELGIUM
Email : vu.nguyenhuynh@uclouvain.be

Process Specification of a Governance-Aware Software Development Method

Vu H. A. Nguyen [1], Manuel Kolp [1] and Yves Wautelet[2]
[1]*LouRIM-CEMIS, Université catholique de Louvain, Belgium*
[2]*KULeuven, Faculty of Economics and Business, Belgium*
{*vu.nguyenhuynh, manuel.kolp*}*@uclouvain.be, yves.wautelet@kuleuven.be*

Abstract: Governing and managing software development in a Requirements-Driven Software Processes such as (GI-) Tropos or even waterfall, V, UP-based ones allow IT managers to propose rules for software processes to cope with stakeholders' requirements and expectations. On the one hand, software development governance has to ensure that software organization business processes meet the strategic requirements of the organization. On the other hand, requirements-driven software methods are development processes using high-level social-oriented models to drive the software life cycle both in terms of project management and deductive iterative engineering techniques. Typically, such methods are well-suited for the inclusion and adaptation of governance and management principles immediately into the software development life cycle. To consolidate both perspectives, this paper proposes a generic process specification using the Software and Systems Process Engineering Metamodel notation allowing to illustrate all disciplines and processes of software development life cycle's governance and management.

Keywords: IT Governance, Software Process, GI-Tropos, SPEM.

1 Introduction

In today's software development methods, the management of requirements all along the life cycle and Information Technology (IT) Governance principles are closely intertwined [KF14]. IT Governance is defined as a framework that ensures the effective and efficient use of IT support and enables the achievement of its corporate strategies and objectives. IT Governance reflects the alignment of IT strategy with the organization strategy to offer value-added for business based on corporate governance objectives [Wei04]. Software engineering [Som10] follows development opportunities corresponding to the evolution of IT and supply software to encounter demands for progressively broader and sophisticated business applications [Ban09]. Software engineering is, additionally, through requirements engineering techniques, dedicated to support human activities and to deal with socio-intentional problems [WK16].

Software development is thus deliberated on the basis of the IT services it provides; it can thus be adapted adequately from the perspective of IT Governance. The goal of IT Governance in software engineering is to ensure that "*the results of a software organizations business processes meet the strategic requirements of the organization*" [CWY08]. However, few specific research has been completed on software processes governance since most governance works have focused on more wide-ranging fields than software engineering.

The purpose of this paper is to describe and formalize a generic process for governance and management rules and constraints in a software development process. In this paper, we overview an iterative process framework for a proper governing and managing requirements-driven software processes from strategic to operational levels. Then we will represent the use of the Software and Systems Process Engineering Metamodel (SPEM 2.0) notation within particular processes in order to illustrate all governance and management disciplines in a requirements-driven software life-cycle.

This paper is organized as follows. Section 2 overviews the proposed development template called Governing I-Tropos (GI-Tropos) for requirements-driven software process and IT governance alignment. Section 3 overviews the Software and Systems Process Engineering Meta-Model (SPEM 2.0) while section 4 illustrates GI-Tropos' disciplines specification using SPEM 2.0 notation. Finally, Section 5 concludes the paper and points out further work.

2 Governing I-Tropos

Iterative Tropos (I-Tropos) [Wau08, WK11, WK16, WKP18, WKP11] is an extension of Tropos [CKM02], a requirements-driven development methodology using i* modeling framework [YGMM11] that supports iterative [Kru03] and agent development [MKG02]. It is a development process using coarse-grained (i.e., high-level) and social-oriented requirement models to drive the software development both in terms of project management [PMI13] and deductive forward engineering (transformational) techniques.

The five phases of traditional Tropos (*Early Requirements*, *Late Requirements*, *Architectural Design*, *Detailed Design*, and *Implementation*) are considered as groups of iterations that are workflows with a minor milestone in order to comply with the most generic terminology. In I-Tropos, the Organizational Modeling and Requirements Engineering disciplines respectively correspond to Tropos' Early and Late Requirements phases. The Architectural and Detailed Design disciplines correspond to the same stages of the traditional Tropos process. I-Tropos also adds up Software Process Management discipline to handle Risk Management, Time Management, Quality Management and Software Process Management [Wau08]. The research method we have followed uses a bottom-up approach, I-Tropos was considered as a given and validated framework and has been enhanced with a (IT services) governance level. IT services are coarse-grained structures aligned with the core values of the organization, i.e., what (added) value it provides to the external world [WK16].

Governing I-Tropos (GI-Tropos) [NKWH17] is an extension of I-Tropos for aligning requirements-driven software processes with IT governance. This extension aims to enable governing and managing requirements-driven software processes to cope with stakeholders' requirements and expectations in the context of business aspects. Figure 1 represents the GI-Tropos process in a classical iterative perspective based on a series of disciplines illustrated in the vertical dimension and a series of phases illustrated in the horizontal dimension. Disciplines of GI-Tropos are grouped in and transversal to each phase. They can be deployed in several iterations by phase depending on each software project characteristics. Consequently, the disciplines of GI-Tropos can be repeated iteratively and the effort/workload spent on each discipline varies from one iteration to another.

From a systems development perspective, GI-Tropos has the four phases (*Setting*, *Blueprinting*, *Building*, *Setuping*) redefining and improving those of I-Tropos plus a new one, *Operation*, to operate the system in the perspective of IT enterprise governance and management. It also adds up core processes of IT governance (*Evaluate*, *Direct*, and *Monitor*), IT management (*Plan*, *Deploy*, *Deliver*, and *Assess*), knowledge management (*Discover*, *Refine*, *Share*, and *Estimate*), quality management (*Define*, *Measure*, and *Enhance*), change management (*Explore*, *Arrange*, *Implement*, and *Review*), and risk management (*Determine*, *Analyze*, *Organize*, and *Perform*). These disciplines' processes are shown in Figure 2.

In terms of disciplines, GI-Tropos includes all disciplines of I-Tropos plus four new ones; they are:

- ***Software Project and Process Governance***: Ensuring software project and software development process are evaluated, directed and monitored to meet stakeholders' requirements and to achieve value added by aligning requirements-driven software processes with IT governance rules and constraints;
- ***Knowledge Management***: Discovering knowledge management infrastructure, mechanisms, and technologies; refining tacit knowledge and explicit knowledge; sharing knowledge management services; estimating knowledge management performance.
- ***Change Management***: Implementing change roadmap, reviewing measurements and analytics;
- ***Risk Management***: Analyzing business goals and impact factors, qualifying risk factors;
- ***Quality Management***: Defining quality requirements, measuring quality assurance and quality control, enhancing quality achieved, ensuring that quality expected and contracted with stakeholders is achieved throughout the system;

These new disciplines ensure that software processes are governed and managed to cope with stakeholders' requirements and expectations.

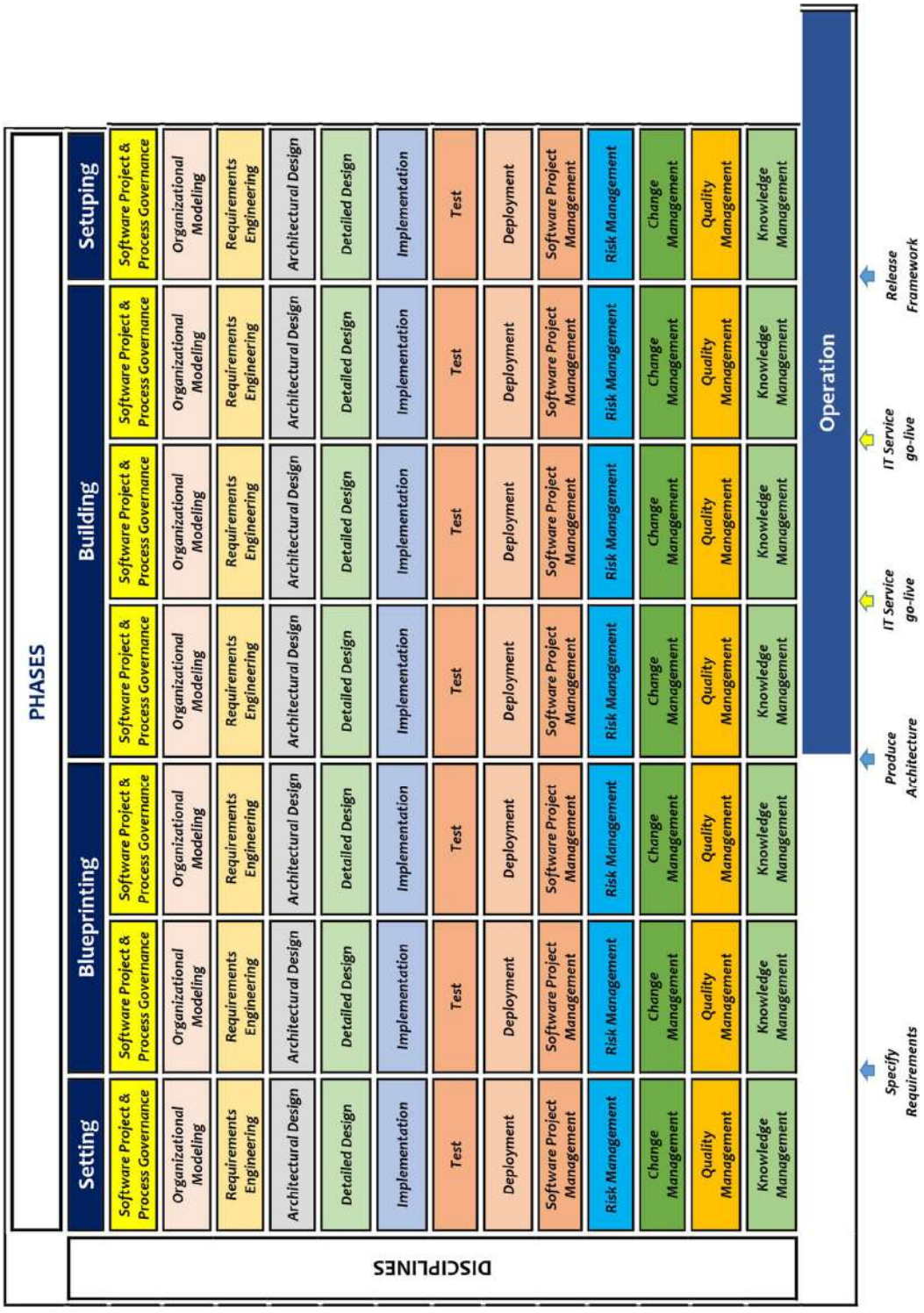

Figure 1: GI-Tropos iterative process framework

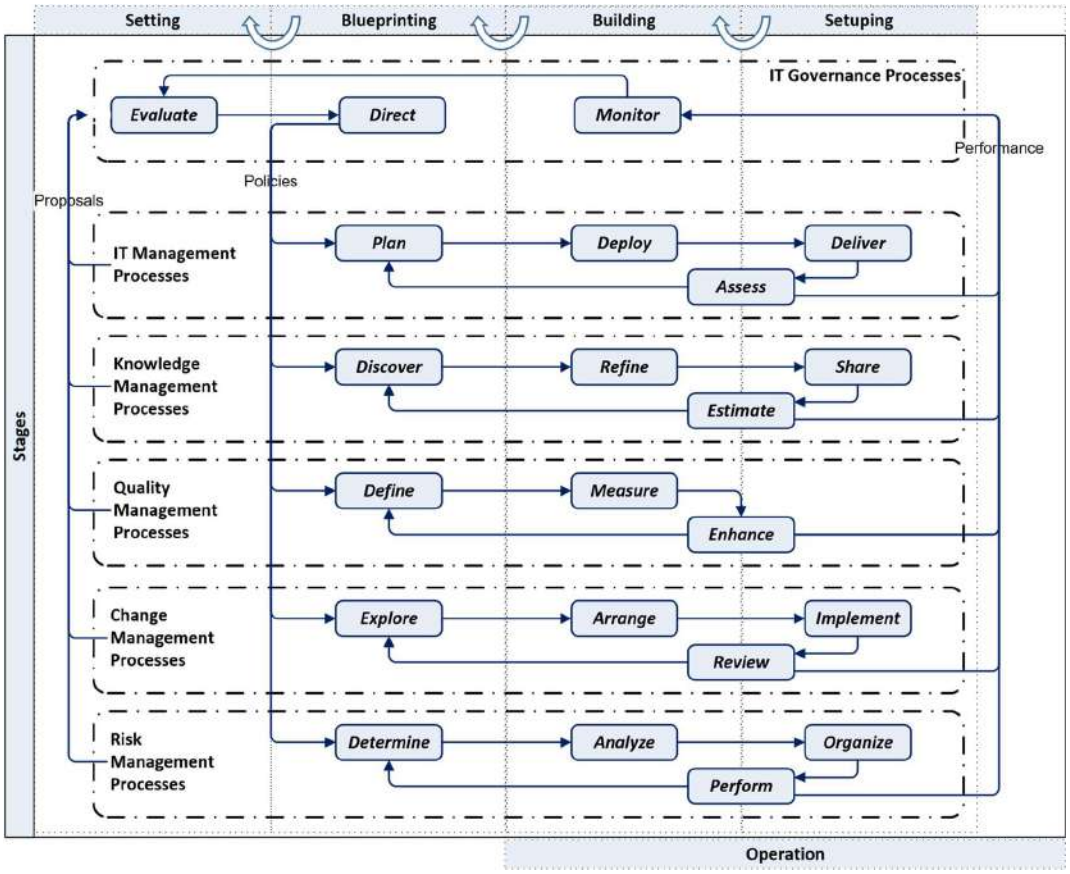

Figure 2: GI-Tropos Framework and Processes

3 SPEM 2.0

The Software and Systems Process Engineering Meta-model (SPEM 2.0) is a specification of the Object Management Group, Inc. (OMG) to describe software and systems development processes and their components [OMG08]. SPEM 2.0 is a meta-model that is used to explain a concrete software development process or a family of related system and software development processes.

The SPEM 2.0 represents a process based on the collaboration between active entities called *RoleDefinitions* that execute tasks called *Activities* on concrete entities called *WorkProducts*. The graphical representation for SPEM 2.0 concepts is illustrated by Unified Modeling Language (UML) stereotypes. Some relevant SPEM 2.0 concepts used in this paper are shown in Figure 3. The OMG [OMG08] defines these concepts as:

- Phase: A *Phase* is an important period within a software development process. A well-defined set of objectives is met during a phase. It ends with a major management checkpoint, a milestone;
- RoleDefinition: A *RoleDefinition* determines a set of related skills, competencies, behavior, and responsibilities of an individual or a set of individuals. *RoleDefinitions* are used by *TaskDefinitions* to specify who performs them along with clarifying a set of *WorkProductDefinitions* they are accountable for;
- Activity: An *Activity* describes something that one or more roles perform. *Activities* are related to timelines. It can be a group of nested process elements such as other *Activity Instances*, *Task Uses*, *Role Uses*, *Milestones*, etc.;

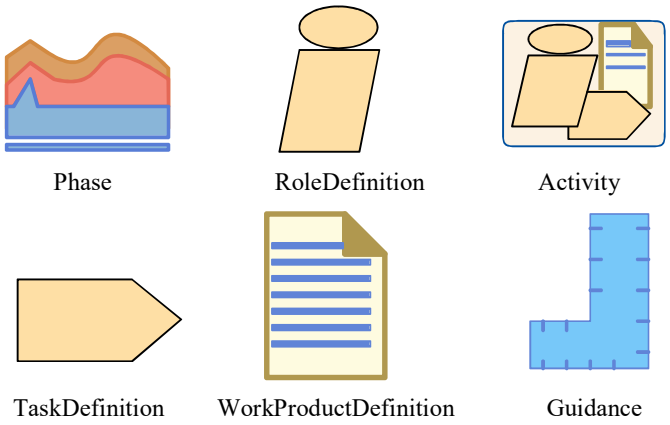

Figure 3: Some SPEM 2.0 Icons.

- TaskDefinition: A *TaskDefinition* determines an assignable unit of work being performed by *RoleDefinitions* instances. A *TaskDefinition* is related to input and output *WorkProducts*. A *TaskDefinition* explains complete step-by-step descriptions of all the work required to achieve its goal;
- WorkProductDefinition: *WorkProductDefinition* is an artifact of a process. It can be used, modified, or produced by *TaskDefinitions*. *Roles* use *WorkProducts* to carry out *Tasks* and construct *WorkProducts* in the course of performing *Tasks*;
- Guidance: *Guidance* provides descriptions and additional information related to other elements. It can be a *Guideline*, *Template*, *Checklist*, etc...

SPEM 2.0 considers the main elements of a software and systems development process as including of *RoleDefinitions*, the *WorkProducts* they are accountable for, and the *TaskDefinitions* that they carry out on the *WorkProducts*. These elements are often associated with *Guidance*.

4 Using SPEM 2.0 Notation to represent GI-Tropos' Disciplines Specification

As said, GI-Tropos is an extension of I-Tropos, an iterative requirements-driven software development process. The latter has already used SPEM to describe and formalize a generic process specification in [WKA06, FKWA06, KWF11]. The GI-Tropos' disciplines will be described in detail by using SPEM 2.0 notation in this section notably to remain fully compatible and consistent with I-Tropos. Firstly, each discipline is illustrated as a package that consists of *RoleDefinitions*, *Activities*, *WorkProducts*, and *Guidances*. Then, a WorkDefinition is presented through an activity diagram. After that, WorkDefinitions are split into a detailed workflow where each activity is presented corresponding to its order. Finally, the WorkProducts and their relationships are identified and illustrated.

GI-Tropos is made of disciplines repeated iteratively while the effort spent on each discipline is variable from one iteration to another. Each iteration belongs to one of the five GI-Tropos' phases (Setting, Prototyping, Building, Setuping, and Operation). The five phases are achieved sequentially and have different goals (achieved through the WorkDefinitions) and evaluated at milestones at the end of each phase. WorkDefinitions checking is performed at the milestones through the use of knowledge and achievement-oriented metrics. The phases and their WorkDefinitions are presented in Figure 4.

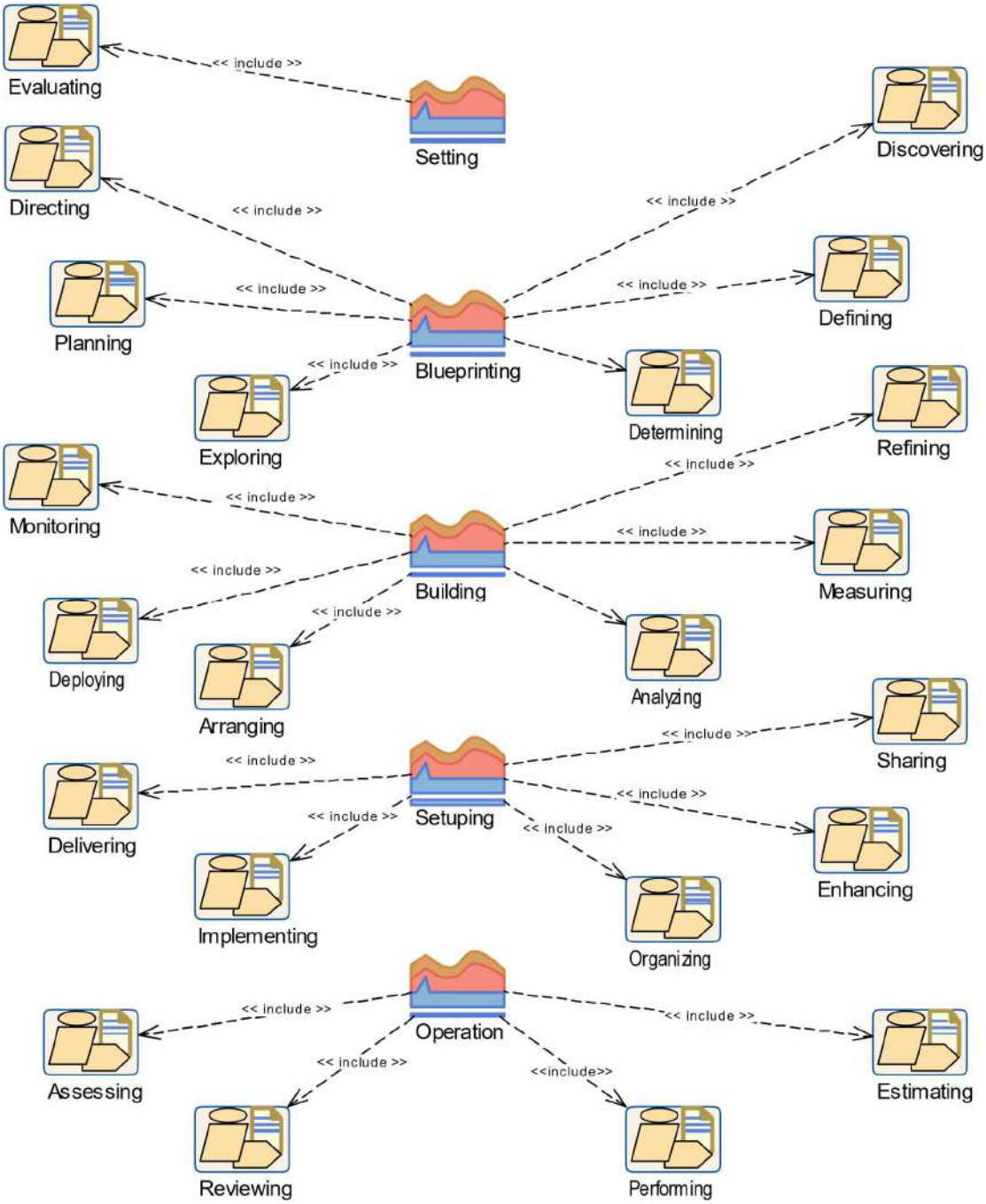

Figure 4: GI-Tropos phases and their Work Definitions.

4.1 Software Project and Process Governance

The Software Project and Process Governance discipline defines the governance mechanisms required for software development as well as their interfacing with project management. The purpose of the Software Project and Process Governance discipline is to ensure that enterprise objectives are achieved by evaluating stakeholder needs, conditions and options; setting direction through prioritization and decision making; and monitoring performance, compliance, and progress against agreed-on direction and objectives.

4.1.1 *Package*

In this section, we highlight that the Software Project and Process Governance discipline involves two *RoleDefinitions*, seven *WorkProducts* as shown in Figure 5.

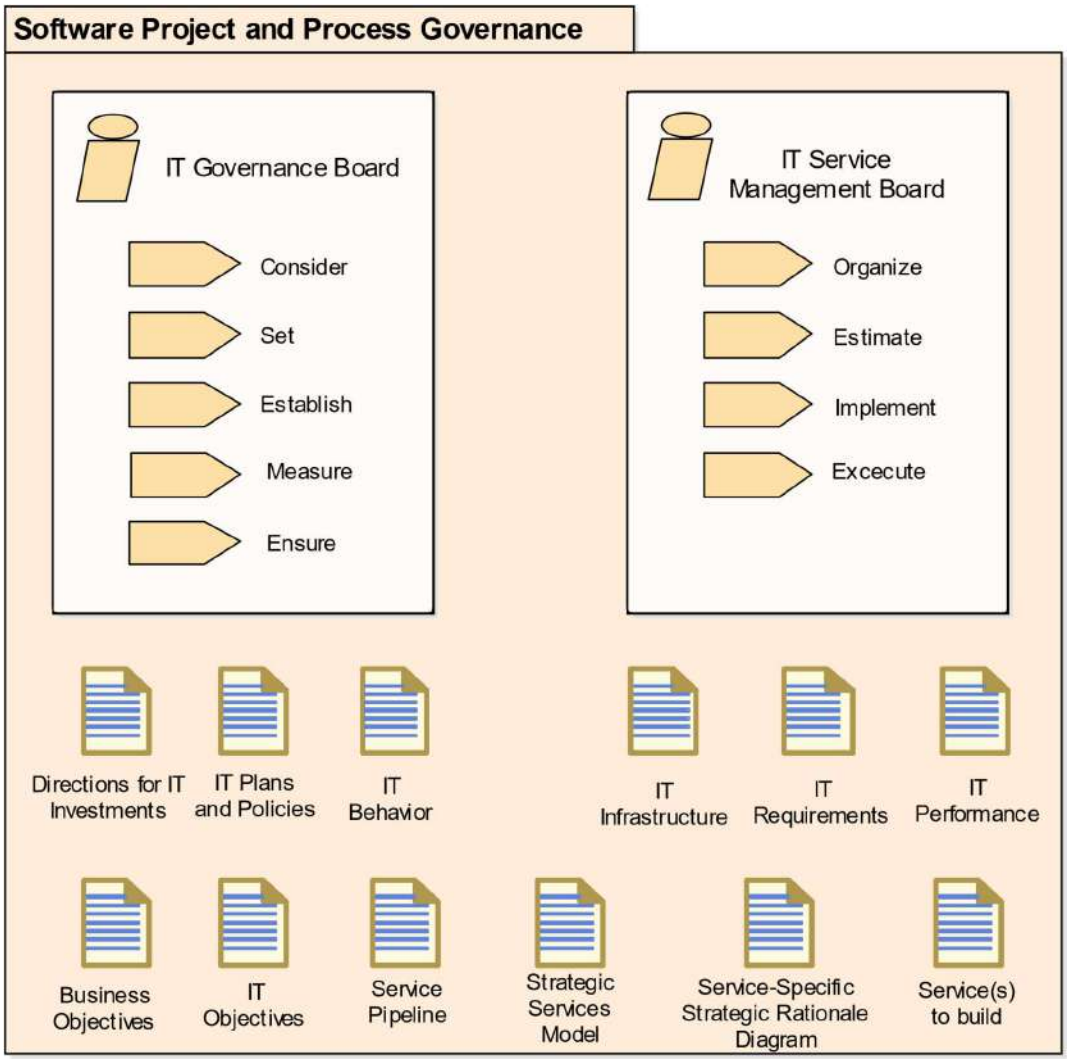

Figure 5: Software Project and Process Governance as a SPEM discipline.

4.1.2 *Work Definitions*

The WorkDefinitions performed during the Software Project and Process Governance discipline are presented in Figure 6 and Figure 7, for further refinements see Figure 8, Figure 9, Table 1 and Table 2.

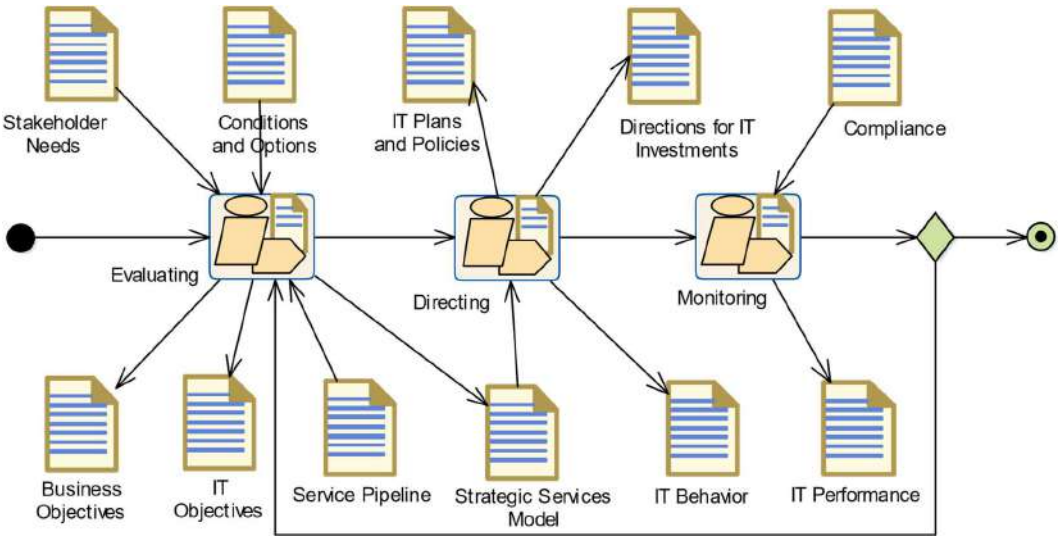

Figure 6: Software Project and Process Governance Work Definition: Governance Mechanism.

The WorkDefinitions performed during the Software Project and Process Governance discipline for interfacing with project management are presented in Figure 7, for further refinements see Figure 9, Table 2.

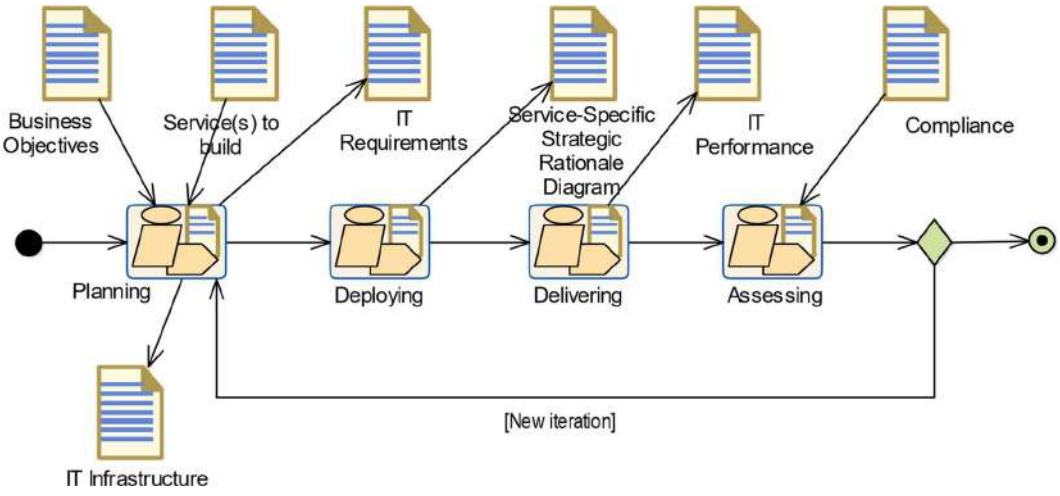

Figure 7: Software Project and Process Governance Work Definition: Interfacing with Project Management.

4.1.3 *Workflow*

The complete specification of the activities performed within software project and process governance is developed in Table 1, the workflow of these activities is presented in Figure 8.

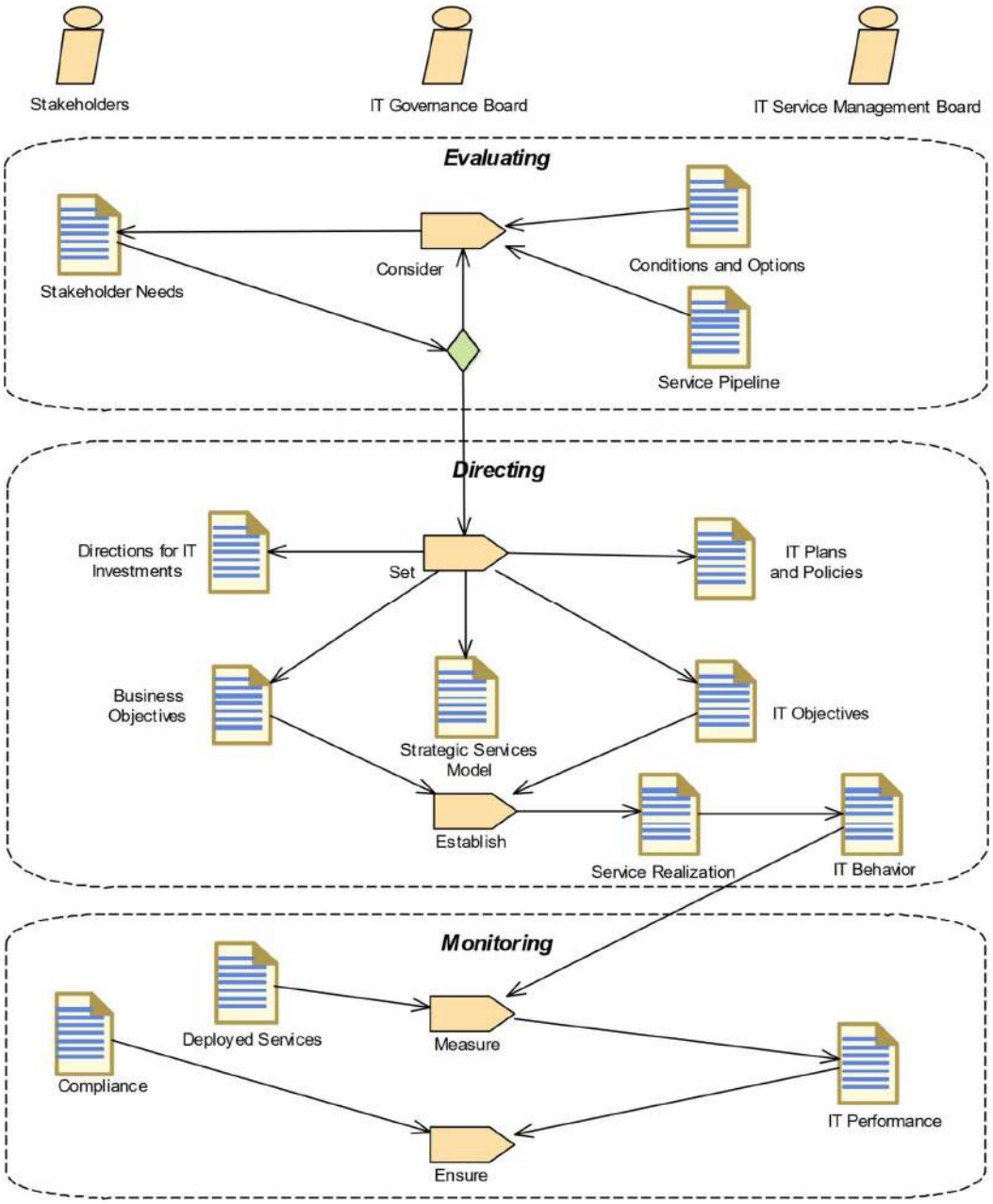

Figure 8: Software Project and Process Governance Workflow: Governance Mechanism.

The complete specification of the activities performed for interfacing with project management is developed in Table 2, the workflow of these activities is presented in Figure 9.

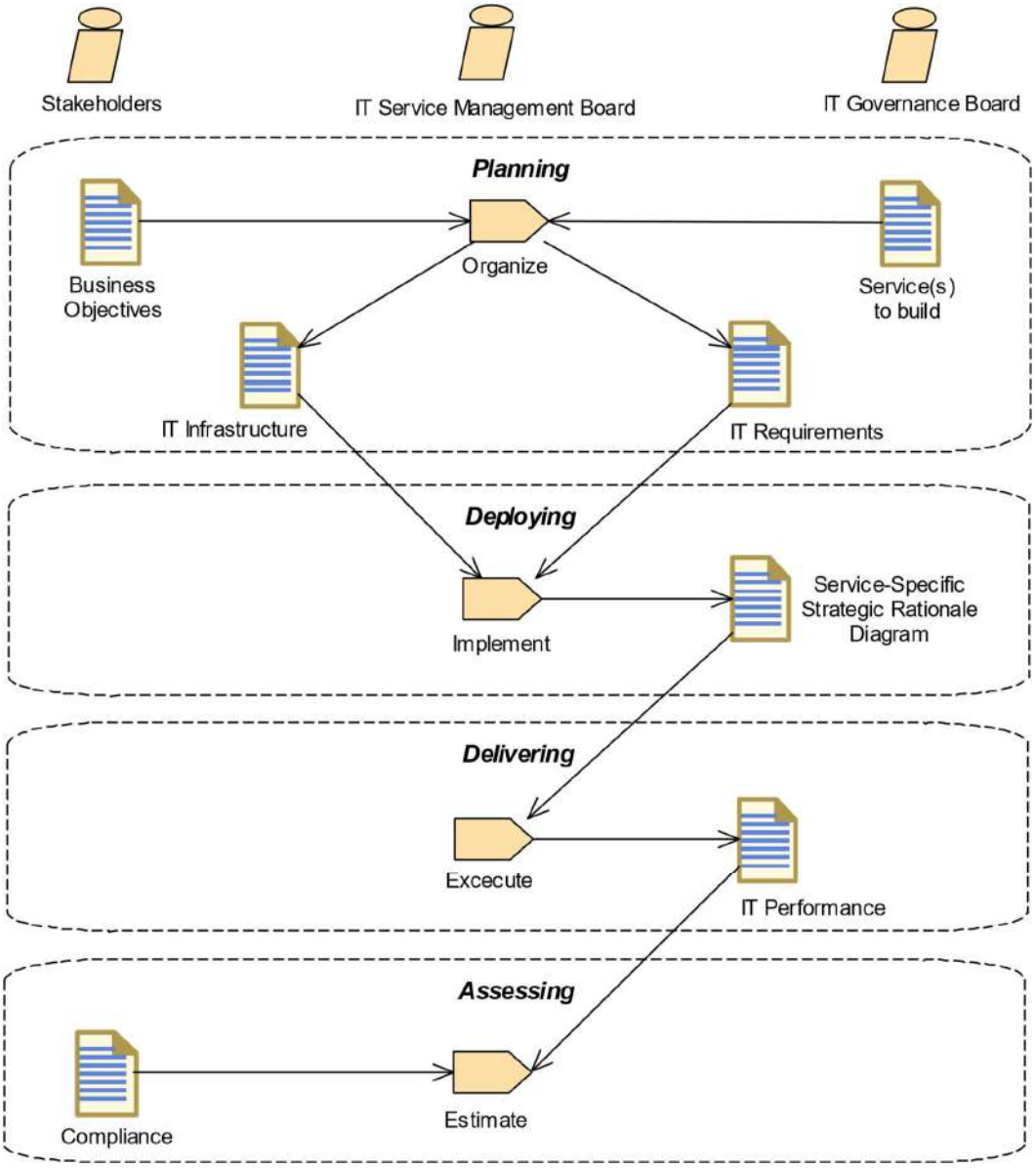

Figure 9: Software Project and Process Governance Workflow: Interfacing with Project Management.

Table 1: Software Project and Process Governance activities description: Governance Mechanism.

No	Activity name	Goal	RoleDefinition	Pre	Parent WorkDefinition	Output WorkProducts
1	Consider	Examining and judging current and future use of IT services include strategy proposals, supply arrangements; considering internal and external pressures (technological changes, economic trends, social trends, and political influences); evaluating continuously; considering current and future business needs and objectives: competitive advantage and specific strategies.	IT Governance Board	-	Evaluating	
2	Set	Assigning responsibility and directing preparation and implementation of software development-based IT services.	IT Governance Board	Activity 1	Directing	IT Plans and Policies Directions for IT Investments Business Objectives IT Objectives
3	Establish	Establishing sound behavior in IT services use through policies; properly plan transition of a project to operational status; encouraging the culture of good IT governance; direct submission of proposals identifying needs.	IT Governance Board	Activity 2	Directing	Service Realization IT Behavior
4	Measure	Monitoring and measuring software development-based IT services performance.	IT Governance Board	Activity 3	Monitoring	IT Performance
5	Ensure	Assuring that performance is in accordance with plans and business objectives; ensuring that software development-based IT services conforms with external obligations (regulatory, legislation, common law, and contractual); ensuring that software development-based IT services conforms with internal work practices.	IT Governance Board	Activity 4	Monitoring	

Table 2: Software Project and Process Governance activities description: Interfacing with Project Management.

No	Activity name	Goal	Role Definition	Pre	Parent WorkDefinition	Output WorkProducts
1	Organize	Covering the use of information and technology and how best it can be used in an organization to help achieve the organization's goals and objectives. It also highlights the organizational and infrastructural form IT is to take in order to achieve the optimal results and to generate the most benefits from the use of software development-based IT services.	IT Service Management Board	-	Planning	IT Requirements IT Infrastructure
2	Implement	Identifying software development-based IT services requirements, acquiring the technology, and implementing it within the enterprise's current business processes.	IT Service Management Board	Activity 1	Deploying	Service-Specific Strategic Rationale Model
3	Execute	Focusing on the delivery aspects of the software development-based IT services. It covers areas such as the execution of the software system within the IT system and its results, in addition to, the support processes that enable the effective and efficient execution of these IT systems.	IT Service Management Board	Activity 2	Delivering	IT Performance
4	Estimate	Dealing with an enterprise's strategy in assessing the needs of the enterprise and whether or not the current software development-based IT services still meets the objectives for which it was designed and the controls necessary to comply with regulatory requirements. Monitoring also covers the issue of an independent assessment of the effectiveness of software development-based IT services in its ability to meet business objectives and the enterprise's control processes by internal and external auditors.	IT Service Management Board	Activity 3	Assessing	

4.1.4 *Work Products*

The complete specification of the WorkProducts used as input or produced as output during the Software Project and Process Governance discipline is developed in Table 3 and Table 4, the relationships between these activities are presented in Figure 10 and Figure 11.

Table 3: Software Project and Process Governance Work Products description: Governance Mechanism.

WorkProduct name	Type	Description
IT Governance Model	WorkProduct	A model consists of WorkProducts, Guidances, Text Documents...
IT Plans and Policies	Text Document	Describing responsibility and directing preparation and implementation of software development-based IT services.
Directions for IT Investments	Text Document	Describing directions for software developmentbased IT services investments.
Business Objectives	WorkProduct	Describing business objectives of organization.
IT Objectives	WorkProduct	Describing IT objectives of organization.
Strategic Service Model	WorkProduct	Describing the services of IT in the organization at strategic level (long-term strategy).
IT Behavior	WorkProduct	Describing the behavior of using software development-based IT services.
IT Performance	WorkProduct	Describing the performance of using software development-based IT services.

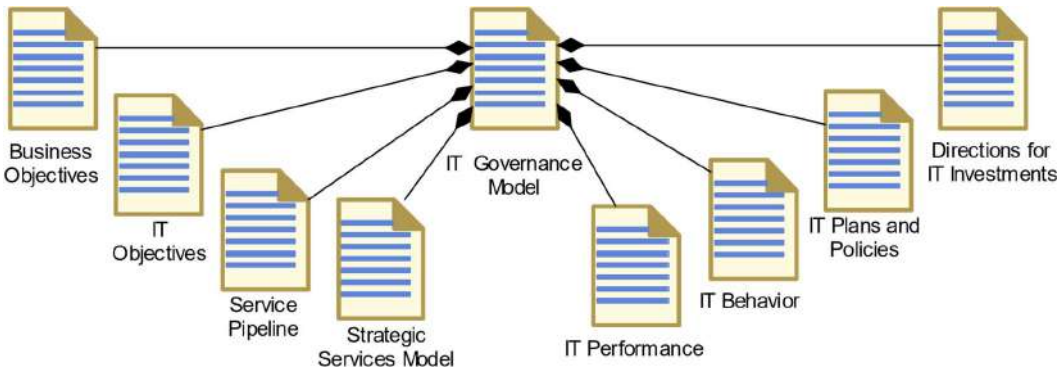

Figure 10: Software Project and Process Governance Work Products relationships: Governance Mechanism.

The complete specification of the WorkProducts used as input or produced as output during the Software Project and Process Governance discipline for interfacing with project management is developed in Table 4, the relationships between these activities are presented in Figure 11.

Table 4: Software Project and Process Governance Work Products description: Interfacing with Project Management.

WorkProduct name	Type	Description
IT Management Model	WorkProduct	A model consists of WorkProducts, Guidances, Text Documents...
IT Requirements	WorkProduct	Describing the requirements of software development-based IT services in the organization.
IT Infrastructure	WorkProduct	Describing the infrastructure of IT in the enterprise.
Service-Specific Strategic Rationale Model	WorkProduct	Describing the functional and non-functional requirements of the services of IT in the organization at rationale level.
IT Performance	WorkProduct	Describing the performance of using software development-based IT services in the organization.

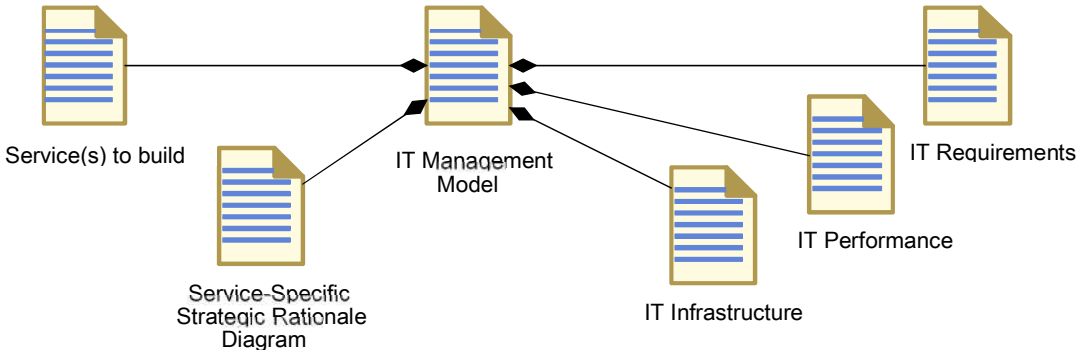

Figure 11: Software Project and Process Governance Work Products relationships: Interfacing with Project Management.

4.2 Knowledge Management

Knowledge management (KM) is defined as '*the effective learning processes associated with exploration, exploitation, and sharing of human knowledge (tacit and explicit) that use appropriate technology and cultural environments to enhance an organization's intellectual capital and performance.*' [Jas11]. The foundations of KM include infrastructure, mechanisms, and technologies [BFS15]. Knowledge Management discipline is to ensure that: KM infrastructure, KM mechanisms, and KM technologies are discovered; tacit knowledge and explicit knowledge are refined; KM services are; KM performance is estimated.

4.2.1 *Package*

In this section the Knowledge Management discipline involves one *RoleDefinition*, seven *WorkProducts* as shown in Figure 12.

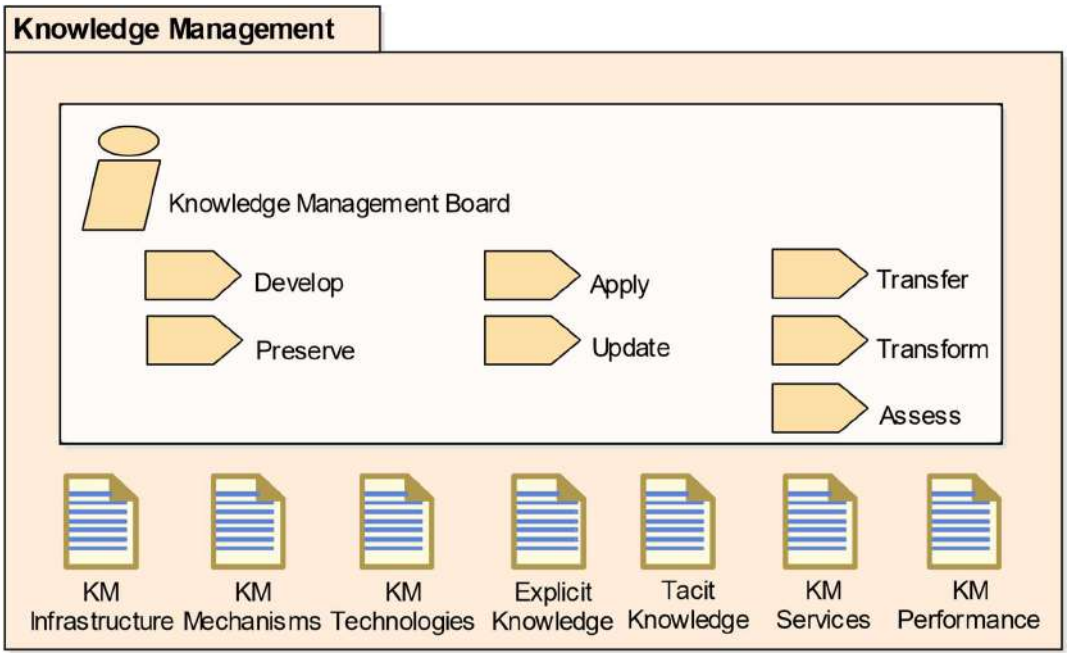

Figure 12: Knowledge Management as a SPEM discipline.

4.2.2 *Work Definitions*

The WorkDefinitions performed during the Knowledge Management discipline are presented. in Figure 13, for further refinements see Figure 14 and Table 5.

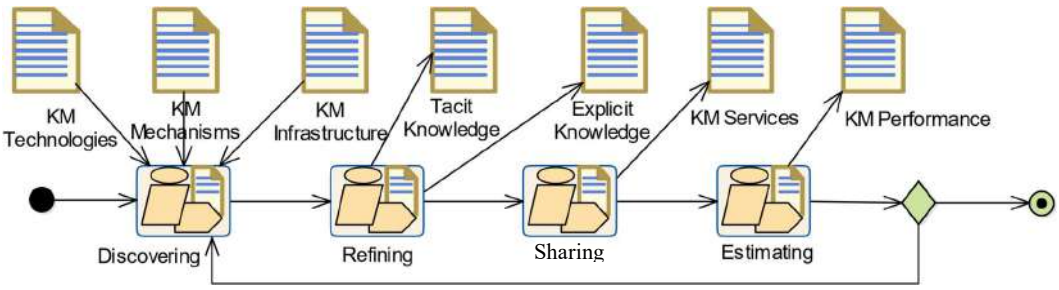

Figure 13: Knowledge Management Work Definitions.

4.2.3 *Workflow*

The complete specification of the activities performed during the Knowledge Management discipline is developed in Table 5, the workflow of these activities is presented in Figure 14.

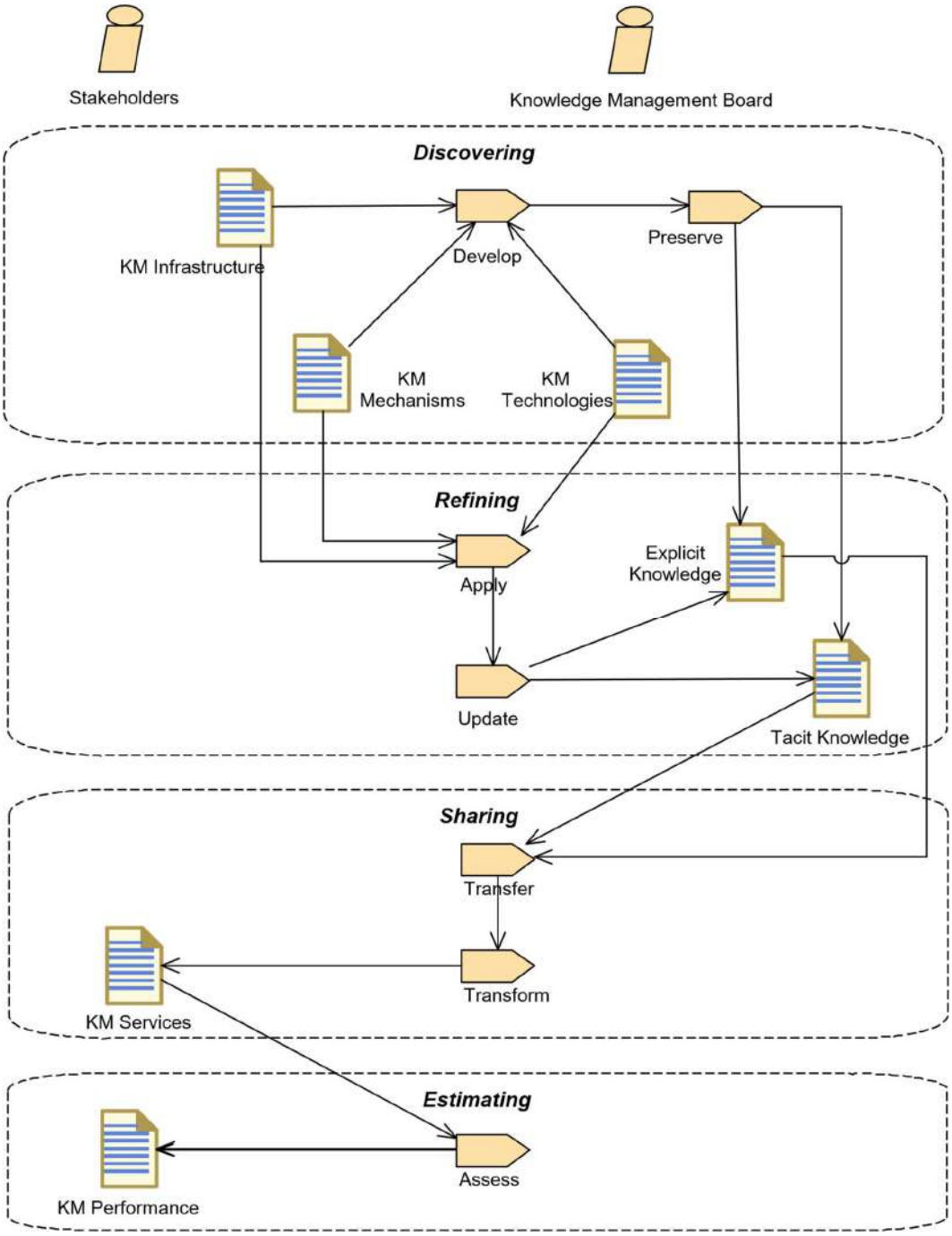

Figure 14: Knowledge Management Workflow.

Table 5: Knowledge Management activities description.

No	Activity name	Goal	RoleDefinition	Pre	Parent WorkDefinition	Output WorkProducts
1	Develop	Ensuring knowledge is acquired, captured, created, discovered correctly. It also plans activities in alignment with the direction set by the governance body to achieve the organizational objectives.	Knowledge Management Board		Discovering	
2	Preserve	Ensuring knowledge is stored, secured, conserved, retained. It also highlights the organizational and infrastructural form KM is to take in order to achieve the optimal results and to generate the most benefits from the use of KM.	Knowledge Management Board	Activity 1	Discovering	Explicit Knowledge Tacit Knowledge
3	Apply	Enabling knowledge to be used, enacted, executed, exploited properly.	Knowledge Management Board	Activity 2	Refining	
4	Update	Enabling knowledge to evolve, improve, be maintained and refreshed	Knowledge Management Board	Activity 3	Refining	Explicit Knowledge Tacit Knowledge
5	Transfer	Ensuring knowledge is communicated, deployed, disseminated, shared. It also deploys activities in alignment with the direction set by the governance body to achieve the organizational objectives.	Knowledge Management Board	Activity 4	Sharing	
6	Transform	Ensuring that knowledge is compiled, formalized, standardized, explicated. It also delivers activities in alignment with the direction set by the governance body to achieve the organizational objectives. It focuses on the delivery aspects of the KM system.	Knowledge Management Board	Activity 5	Sharing	KM Services
7	Assess	Ensuring knowledge is appraised, evaluated, validated, verified. It also assesses activities in alignment with the direction set by the governance body to achieve the organizational objectives.	Knowledge Management Board	Activity 6	Estimating	KM Performance

4.2.4 *Work Products*

The complete specification of the WorkProducts used as input or produced as output during the Knowledge Management discipline is developed in Table 6, the relationships between these activities are presented in Figure 15.

Table 6: Knowledge Management Work Products description.

WorkProduct name	Type	Description
KM Infrastructure	WorkProduct	Including organization culture, organization structure, information technology infrastructure, common knowledge, and physical environment.
KM Mechanisms	WorkProduct	Organizational or structural means used to promote knowledge management.
KM Technologies	WorkProduct	Information technologies that can be used to facilitate knowledge management.
Tacit Knowledge	WorkProduct	Knowledge that has not been documented.
Explicit Knowledge	WorkProduct	Knowledge that has been documented.
KM Services	WorkProduct	Describing the services of knowledge management in the enterprise.
KM Performance	WorkProduct	Describing the performance of managing knowledge in the enterprise.

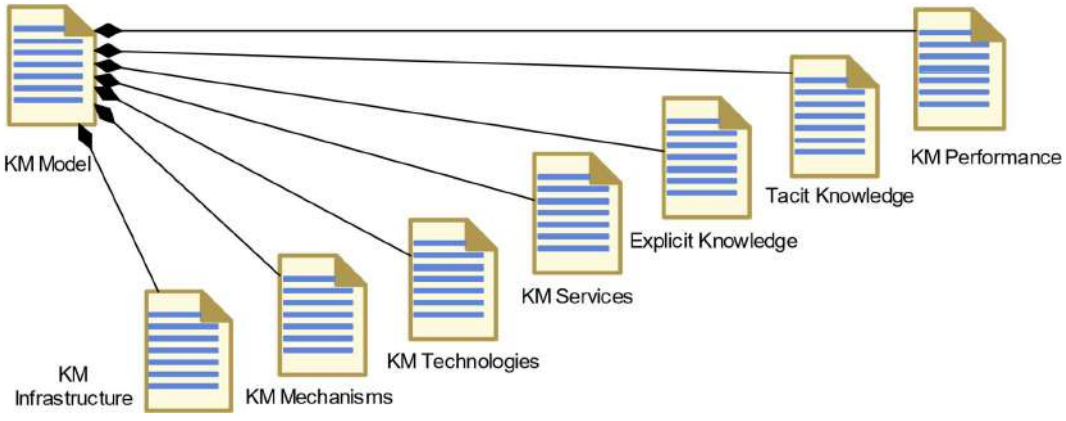

Figure 15: Knowledge Management Work Products relationships.

4.3 Change Management

Following [Ins13], "*Change management is a comprehensive, cyclic, and structured approach for transitioning individuals, groups, and organizations from a current state to a future state with intended business benefits*".

4.3.1 *Package*

In this section the Change Management discipline involves one *RoleDefinition* and six *WorkProducts* as shown in Figure 16.

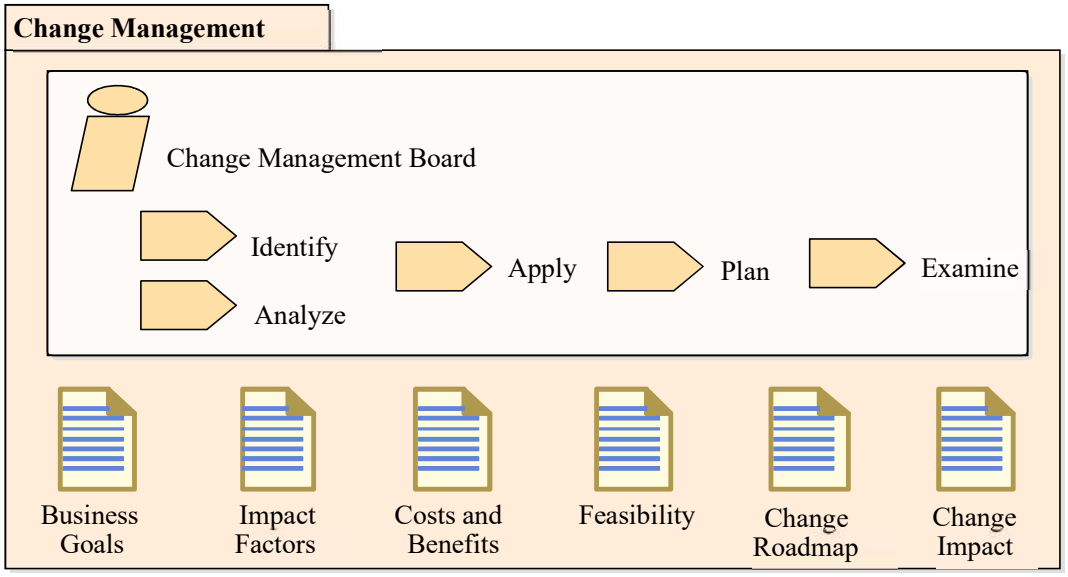

Figure 16: Change Management as a SPEM discipline.

4.3.2 *Work Definitions*

The WorkDefinitions performed during the Change Management discipline are presented in Figure 17, for further refinements see Figure 18 and Table 7.

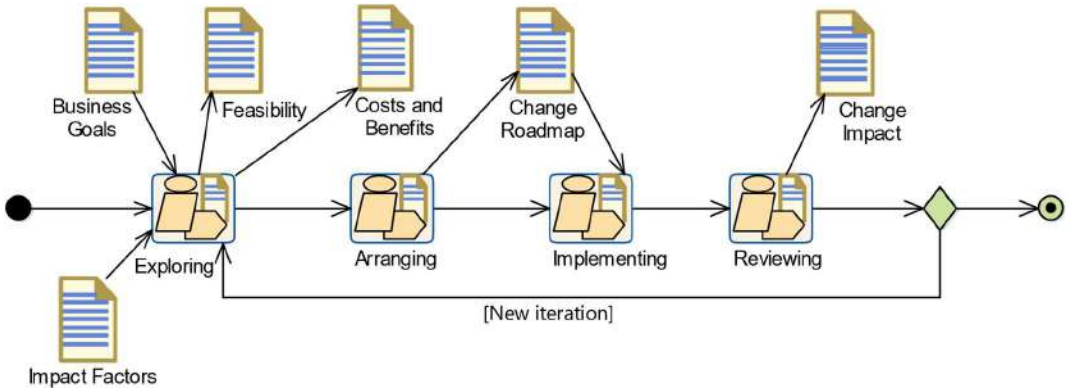

Figure 17: Change Management Work Definitions.

4.3.3 *Workflow*

The complete specification of the activities performed during the Change Management discipline is developed in Table 7, the workflow of these activities is presented in Figure 18.

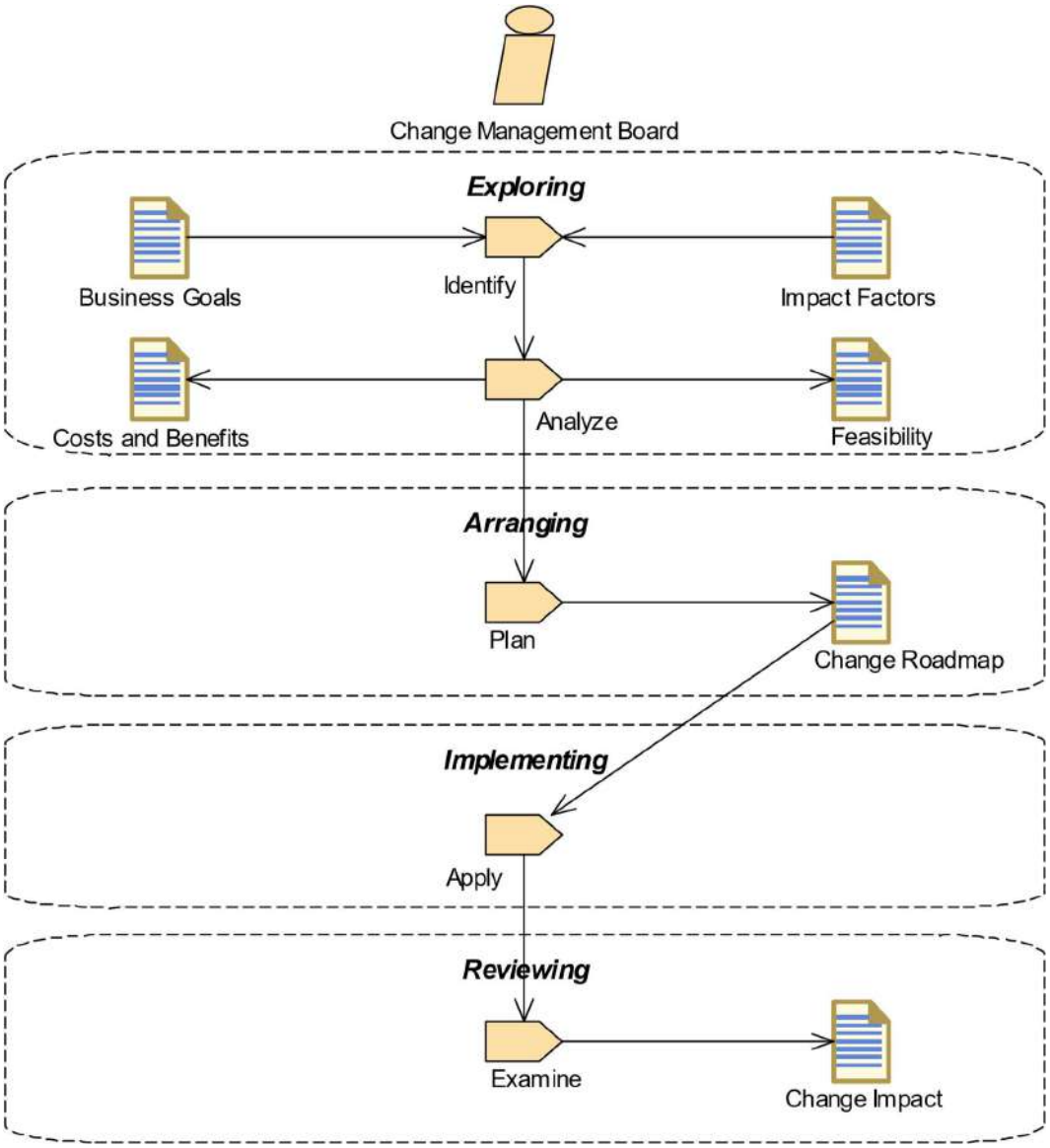

Figure 18: Change Management Workflow.

Table 7: Change Management activities description.

No	Activity name	Goal	RoleDefinition	Pre	Parent WorkDefinition	Output WorkProducts
1	Identify	Identifying business goals and impact factors	Change Management Board		Exploring	Business Goals Impact Factors
2	Analyze	Analyzing costs and benefits, feasibility	Change Management Board	Activity 1	Exploring	Costs and Benefits Feasibility
3	Plan	Planning change roadmap	Change Management Board	Activity 2	Arranging	Change Roadmap
4	Apply	Applying change roadmap	Change Management Board	Activity 3	Implementing	
5	Examine	Examining the change impacts	Change Management Board	Activity 4	Reviewing	Change Impact

4.3.4 *Work Products*

The complete specification of the WorkProducts used as input or produced as output during the Change Management discipline is developed in Table 8, the relationships between these activities are presented in Figure 19.

Table 8: Change Management Work Products description.

WorkProduct name	Type	Description
Business Goals	WorkProduct	Describing what is expected to accomplish over a specific period of time.
Impact Factors	WorkProduct	Comprising of factors which impact the success and approach of operations.
Cost and Benefits	WorkProduct	Describing cost and benefits of change.
Feasibility	WorkProduct	Describing feasibility of change.
Change Roadmap	WorkProduct	Describing the roadmap for building change management capability.
Change impact	WorkProduct	Describing change impacts that will influence on software engineering organizations.

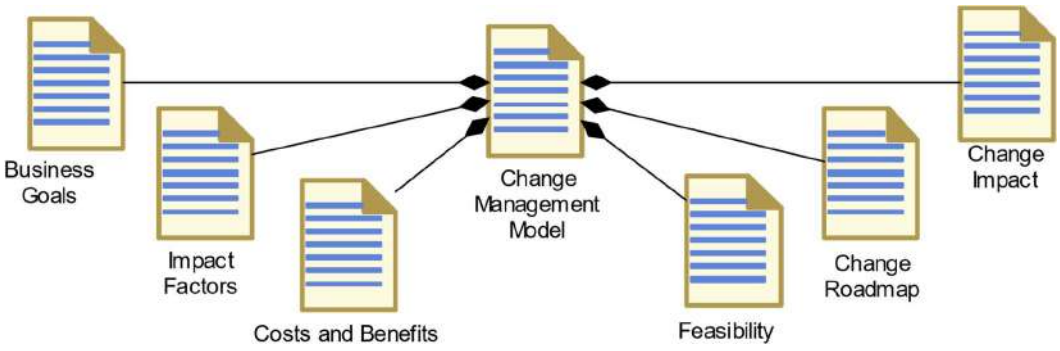

Figure 19: Change Management Work Products relationships.

4.4 Risk Management

Risk management is recognized as a process consisting of well-defined steps of identification, analysis, response, monitoring and control [BBW11]. Risk management concerns all activities that are performed to reduce the uncertainties associated with certain tasks, or events. Change and Risk Management discipline is to ensure that: business goals and impact factors are analyzed, risk factors are qualified, change roadmap is implemented, measurements and analytics are reviewed.

4.4.1 *Package*

In this section the Risk Management discipline involves one *RoleDefinition* and four *WorkProducts* as shown in Figure 20.

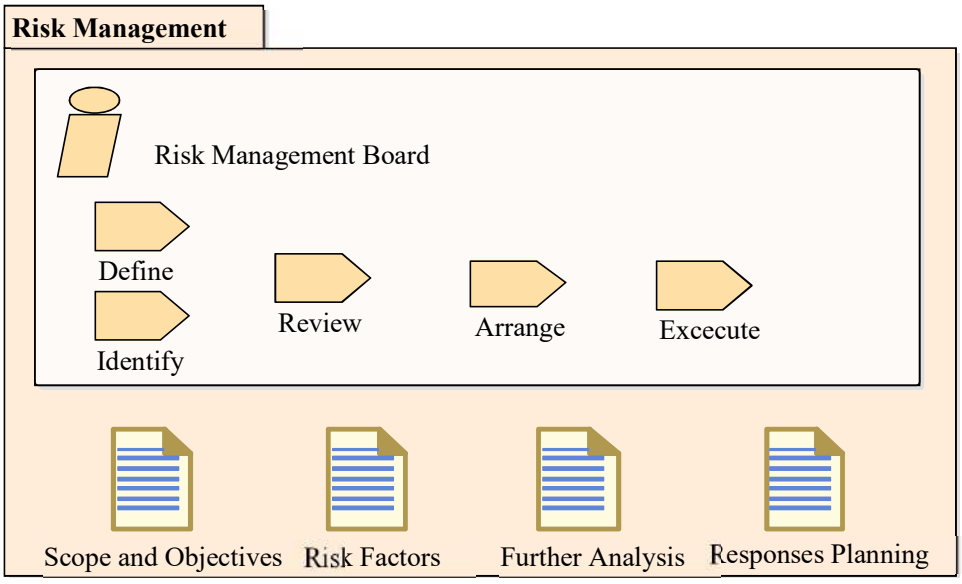

Figure 20: Risk Management as a SPEM discipline.

4.4.2 *Work Definitions*

The WorkDefinitions performed during the Risk Management discipline are presented in Figure 21, for further refinements see Figure 22 and Table 9.

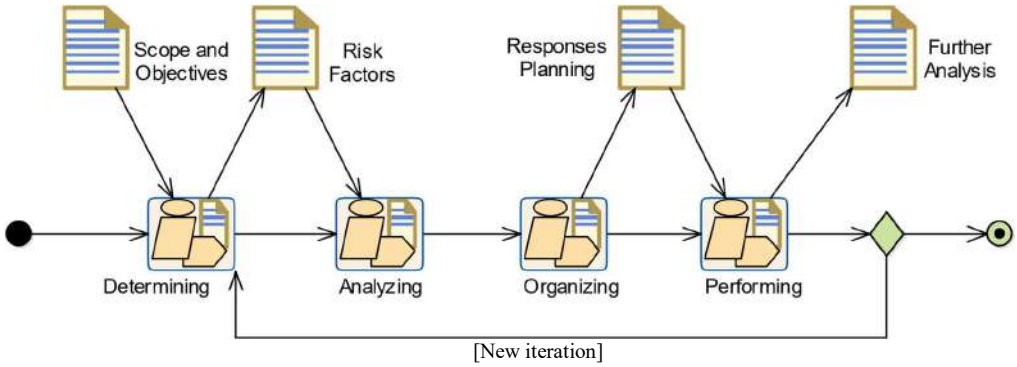

Figure 21: Risk Management Work Definitions.

4.4.3 *Workflow*

The complete specification of the activities performed during the Risk Management discipline is developed in Table 9, the workflow of these activities is presented in Figure 22.

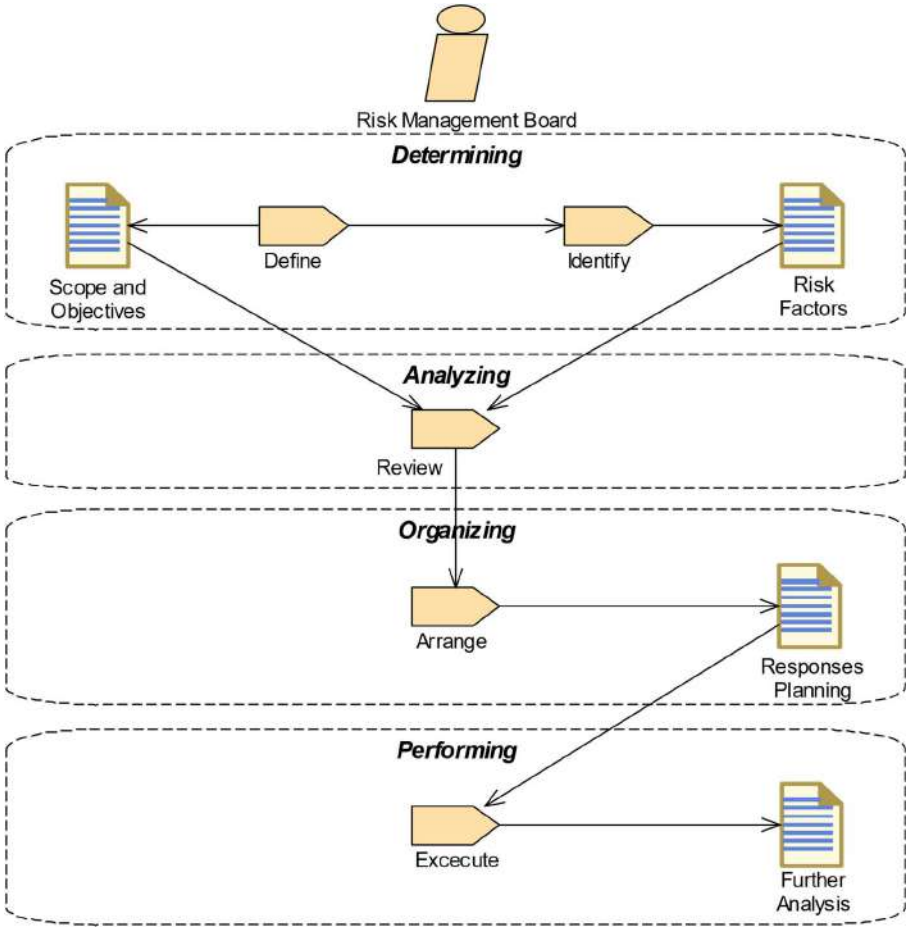

Figure 22: Risk Management Workflow.

Table 9: Risk Management activities description.

No	Activity name	Goal	RoleDefinition	Pre	Parent WorkDefinition	Output WorkProducts
1	Define	Defining scope and objectives of risk management	Risk Management Board		Determining	Scope and Objectives
2	Identify	Identifying risk factors	Risk Management Board	Activity 1	Determining	Risk Factors
3	Review	Reviewing risk factors	Risk Management Board	Activity 2	Analyzing	
4	Arrange	Arranging responses planning	Risk Management Board	Activity 3	Organizing	Responses Planning
5	Execute	Executing further analysis	Risk Management Board	Activity 4	Performing	Further Analysis

4.4.4 *Work Products*

The complete specification of the WorkProducts used as input or produced as output during the Risk Management discipline is developed in Table 10, the relationships between these activities are presented in Figure 23.

Table 10: Risk Management Work Products description.

WorkProduct name	Type	Description
Scope and Objectives	WorkProduct	Describing scope and objectives of risk management.
Risk Factors	WorkProduct	Describing the uncertain conditions and influences that will be negative effects on the cost, duration and quality.
Responses Planning	WorkProduct	Describing the plan for performing risk management.
Further Analysis	WorkProduct	Describing further analysis that would be developed applied to software engineering organizations

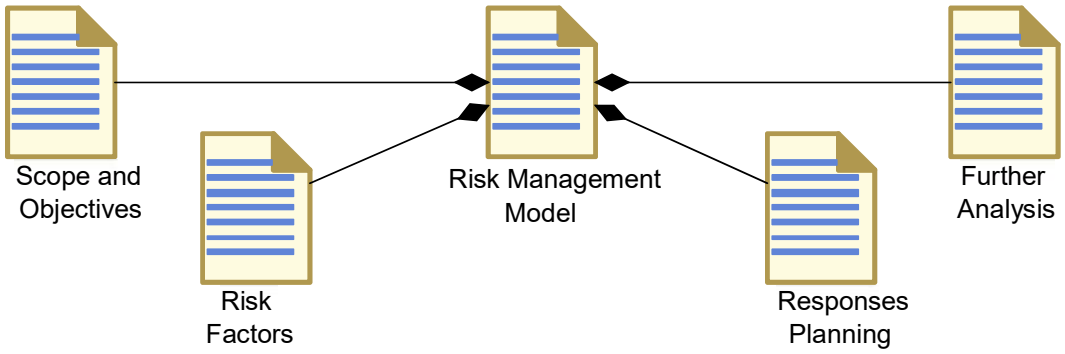

Figure 23: Risk Management Work Products relationships.

4.5 **Quality Management**

Quality management is to ensure that a product or service is in agreement. There are four main components of quality management: quality planning, quality assurance, quality control, and quality improvement [Ros14]. Quality management discipline ensures that: quality requirements are defined, quality assurance and quality control are measured, quality achieved is enhanced, quality expected and contracted with stakeholders is achieved throughout the system.

4.5.1 *Package*

In this section the Quality Management discipline involves two *RoleDefinitions*, four *WorkProducts* and a *Guidance* as shown in Figure 24.

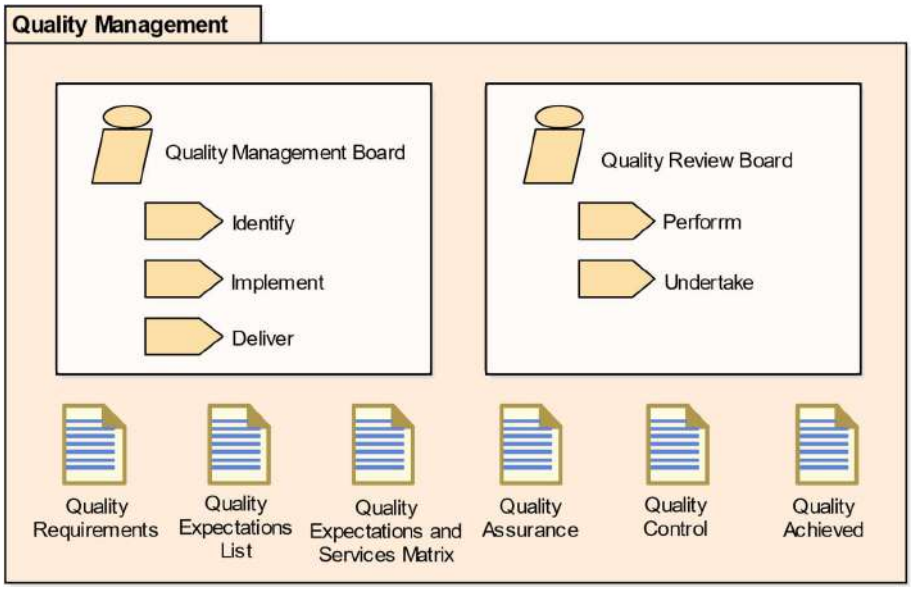

Figure 24: Quality Management as a SPEM discipline.

4.5.2 *Work Definitions*

The WorkDefinitions performed during the Quality Management discipline are presented. in Figure 25, for further refinements see Figure 26 and Table 11.

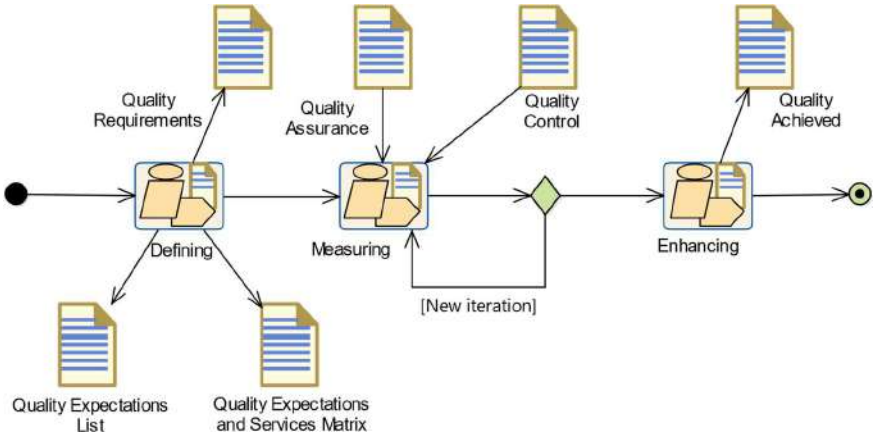

Figure 25: Quality Management Work Definitions.

4.5.3 *Workflow*

The complete specification of the activities performed during the Quality Management discipline is developed in Table 11, the workflow of these activities is presented in Figure 26.

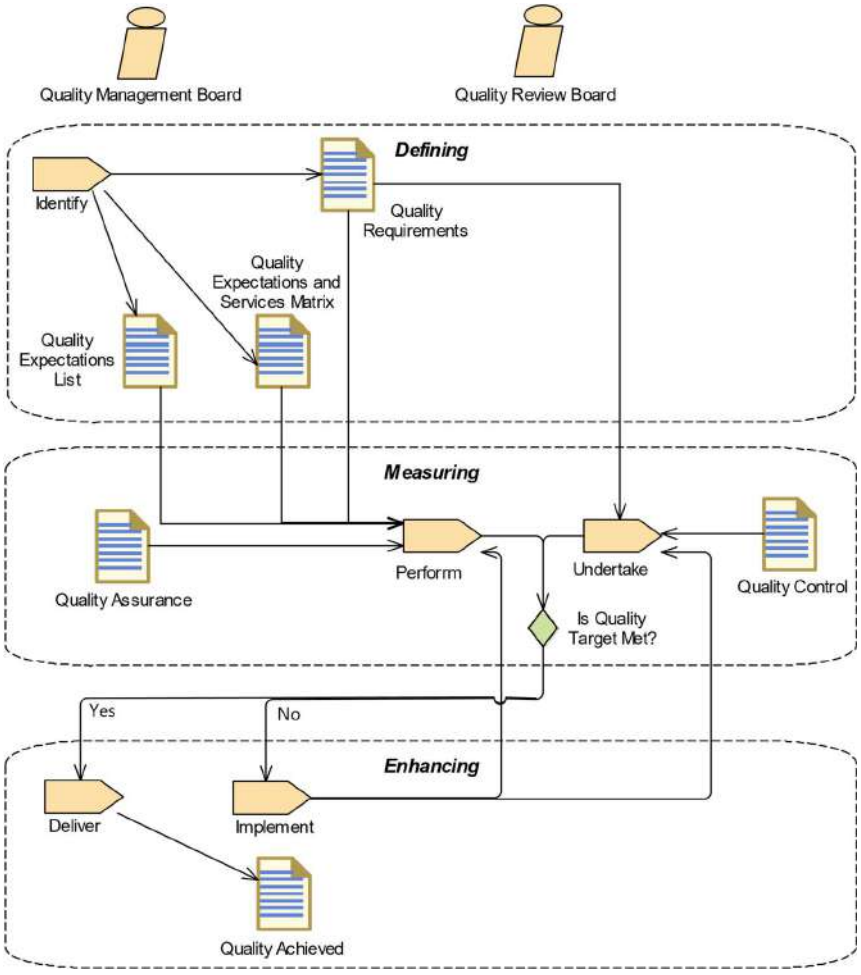

Figure 26: Quality Management Workflow.

Table 11: Quality Management activities description.

No	Activity name	Goal	RoleDefinition	Pre	Parent WorkDefinition	Output WorkProducts
1	Identify	Ensuring the quality targets identified within the quality plan	Quality Management Board		Defining	Quality Requirements Quality Expectations List Quality Expectations and Service Matrix
2	Perform	Ensuring the level of quality of the deliverables and processes	Quality Review Board	Activity 1	Measuring	
3	Undertake	Ensuring compliance to Quality Assurance Standards	Quality Review Board	Activity 1	Measuring	
4	Deliver	Comparing the level of quality achieved to the quality standards set	Quality Management Board	Activity 2 Activity 3	Enhancing	Quality Achieved
5	Implement	Implementing quality improvement actions as necessary	Quality Management Board	Activity 2 Activity 3	Enhancing	

4.5.4 *Work Products*

The complete specification of the WorkProducts used as input or produced as output during the Quality Management discipline is developed in Table 12, the relationships between these activities are presented in Figure 27.

Table 12: Quality Management Work Products description.

WorkProduct name	Type	Description
Quality Requirements	WorkProduct	Describing the processes and metrics that will be used. Quality Requirements need to be agreed with relevant stakeholders to ensure that their expectations for quality are correctly identified.
Quality Expectations List	WorkProduct	Describing the list of Quality Expectations of software development-based IT services in the organization.
Quality Expectations and Services Matrix	WorkProduct	Describing the matrix of Quality Expectations and IT services in the organization.
Quality Assurance	WorkProduct	Validating the use of procedures and standards in consistency, and ensuring the staff have the proper knowledge, skills and attitudes to fulfill their project roles and responsibilities in a competent manner.
Quality Control	WorkProduct	Consisting of inspection, testing, and measurement. Quality Control verifies that the deliverables conform to specification and meet stakeholder expectations in a proper purpose.
Quality Achieved	WorkProduct	Describing the way that quality assurance and quality control are used to drive improvements in efficiency and effectiveness.

Figure 27: Quality Management Work Products relationships.

4.6 WorkProducts dependency

The dependencies among the GI-Tropos process WorkProducts is illustrated in Figure 28. For instance, IT Governance Direction depends on Enterprise Objectives.

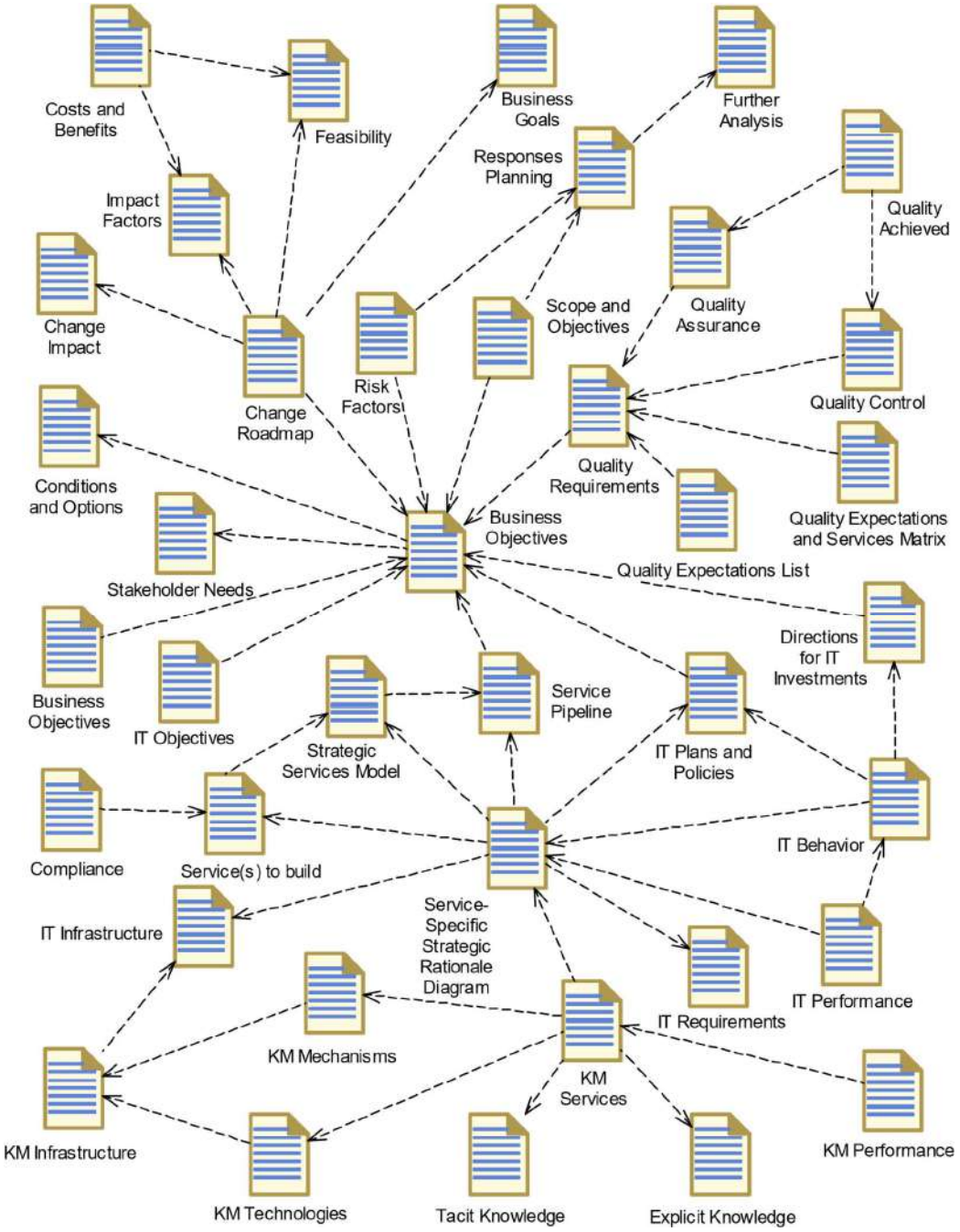

Figure 28: GI-Tropos WorkProducts Dependency.

4.7 Process roles

The different roles implied in the GI-Tropos process, as well as their hierarchy, is presented as Figure 29 and is described in Table 13. For instance, the IT Governance Board is a specialization of Board of Governance and Management.

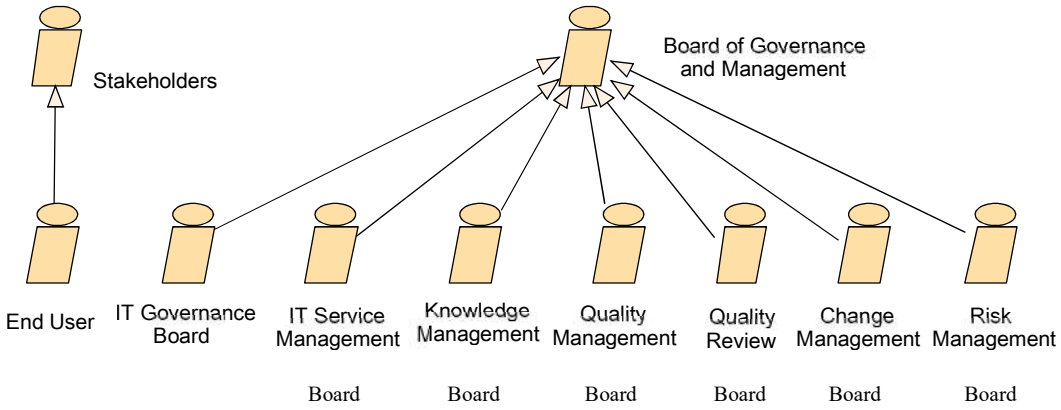

Figure 29: GI-Tropos process roles.

Table 13: GI-Tropos process roles description.

Role	Description
Stakeholder	Represents interest groups whose needs must be satisfied by the software processes. This role may be played by anyone who is (or potentially will be) materially affected by the outcome of software development process.
IT Governance Board	Represents interest groups is suggested to be in charge of governing the software development process.
IT Service Management Board	Represents interest groups is suggested to be in charge of managing the IT services in software development process.
Knowledge Management Board	Represents interest groups is suggested to be in charge of managing the knowledge in software development process.
Quality Management Board	Represents interest groups is suggested to be in charge of managing the quality in software development process.
Quality Review Board	Represents interest groups is suggested to be in charge of reviewing the qualities in software development process.
Change Management Board	Represents interest groups is suggested to be in charge of managing the changes in software development process.
Risk Management Board	Represents interest groups is suggested to be in charge of managing the risks in software development process.

5 Conclusion and Future Work

Today's software processes have to match with their operational and organizational environment. The management of requirements through the software development life-cycle is closely interconnected with IT Governance practices and frameworks. Illustrating all disciplines and processes of the governance and management rules and constraints in a software development process allows them to cope with stakeholders' requirements and expectations. Contributions of this paper consist of a complete specification of a governing iterative software development process called GI-Tropos using the SPEM 2.0 notation. This process includes all I-Tropos disciplines plus four new ones (*Software Project and Process Governance*, *Change Management*, *Risk Management*, *Quality Management*, *Knowledge Management*). Moreover, GI-Tropos also improves and redefines all phases of I-Tropos (*Setting*, *Blueprinting*, *Building*, *Setuping*) plus a new one, *Operation*, to operate the system in the perspective of IT enterprise governance and management.

The disciplines of GI-Tropos should now to be validated on use case: current work is undergoing on a collaborative platform for outbound logistics called in the TransLogisTIC project. This project was to propose combined, performing and complete transportations in the Walloon Region with transport by rail particularly promoted in accordance with the European policy. The TransLogisTIC project was re-built with I-Tropos models for validation purposes in the context of this work. We focus on validating governance and management rules and constraints in the software development process of an online collaborative logistics platform allowing the major outbound logistics stakeholders (chargers, carriers, infrastructure managers, and final clients...) to share information for better optimization of the logistics chain.

Further works point to other additional practices that need to be integrated into GI-Tropos to propose a completed software processes governance framework taking into consideration, for instance, project management and agile practices [AL12, Kru13] for managing the day-to-day activities and reacting to changing requirements and feedback. Moreover, mapping processes that integrate relevant practices (IT governance, IT management, software project and program management and software development processes) to GI-Tropos should also be performed. In addition, a CASE tool should be developed to help to design and to implement all the models defined in this paper.

References

[AL12] Scott W. Ambler and Mark Lines. *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, 1st edition, 2012.

[Ban09] P. L. Bannerman. Software development governance: A meta-management perspective. In *2009 ICSE Workshop on Software Development Governance*, pages 3–8, May 2009.

[BBW11] Karel Bakker, Albert Boonstra, and Hans Wortmann. Risk management affecting IS/IT project success through communicative action. 42:75 – 90, 04 2011.

[BFS15] Irma Becerra-Fernandez and Rajiv Sabherwal. *Knowledge Management: Systems and Processes*. Routledge, 2015.

[CKM02] Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards requirements-driven information systems engineering: the Tropos project. *Inf. Syst.*, 27(6):365–389, 2002.

[CWY08] Sunita Chulani, Clay Williams, and Avi Yaeli. Software development governance and its concerns. In *Proceedings of the 1st International Workshop on Software Development Governance*, SDG '08, pages 3–6, New York, NY, USA, 2008. ACM.

[FKWA06] Stéphane Faulkner, Manuel Kolp, Yves Wautelet, and Youssef Achbany. A formal description language for multi-agent architectures. In *Agent-Oriented Information Systems IV, 8th International Bi-Conference Workshop, AOIS 2006, Hakodate, Japan, May 9, 2006 and Luxembourg, Luxembourg, June 6, 2006, Revised Selected Papers*, pages 143–163, 2006.

[Ins13] Project Management Institute. *Managing Change in Organizations: A Practice Guide*. PMI, 2013.

[Jas11] Ashok Jashapara. *Knowledge Management: An Integrated Approach.* Prentice-Hall, 2nd edition, 2011.

[KF14] Monde Kalumbilo and Anthony Finkelstein. Linking strategy, governance, and performance in software engineering. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE 2014, pages 107–110, New York, NY, USA, 2014. ACM.

[Kru03] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, 3rd edition, 2003.

[Kru13] Philippe Kruchten. Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4):351–361, 2013.

[KWF11] Manuel Kolp, Yves Wautelet, and Stéphane Faulkner. *Social-Centric Design of MultiAgent Architectures*, pages 425 – 456. The MIT Press, 2011.

[MKG02] John Mylopoulos, Manuel Kolp, and Paolo Giorgini. Agent-oriented software development. In *Hellenic Conference on Artificial Intelligence*, pages 3–17. Springer Berlin Heidelberg, 2002.

[NKWH17] Vu H. A. Nguyen, Manuel Kolp, Yves Wautelet, and Samedi Heng. Aligning Requirements -driven Software Processes with IT Governance. In *ICSOFT 2017 Proceedings of the 12th International Conference on Software and Data Technologies, Madrid, Spain, 24-26 July, 2017*, pages 338–345, 2017.

[OMG08] OMG. *Software and Systems Process Engineering Meta-Model Specification Version 2.0*, 2008.

[PMI13] PMI. *A Guide to The Project Management Body of Knowledge*. Project Management Institute, 2013.

[Ros14] Kenneth H. Rose. *Project Quality Management: Why, What and How, Second Edition*. J. Ross Publishing, 2014.

[Som10] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition, 2010.

[Wau08] Yves Wautelet. *A goal-driven project management framework for multi-agent software development: The case of I-Tropos*. PhD thesis, Université catholique de Louvain, 2008.

[Wei04] Ross J.W. Weill, P. *IT governance: How Top Performers Manage IT Decision Rights for Superior Results, Watertown, MA: Harvard Business School Press*. 2004.

[WK11] Yves Wautelet and Manuel Kolp. Goal driven iterative software project management. In *ICSOFT 2011 - Proceedings of the 6th International Conference on Software and Data Technologies, Volume 2, Seville, Spain, 18-21 July, 2011*, pages 44–53, 2011.

[WK16] Yves Wautelet and Manuel Kolp. Business and model-driven development of BDI multi-agent systems. *Neurocomputing*, 182:304–321, 2016.

[WKA06] Yves Wautelet, Manuel Kolp, and Youssef Achbany. S-Tropos : An iterative spemcentric software project management process. *Louvain School of Management, Université catholique de Louvain, Louvain-la-Neuve, Belgium*, 2006.

[WKP11] Yves Wautelet, Manuel Kolp, and Stephan Poelmans. Requirements-driven iterative project planning. In *Software and Data Technologies - 6th International Conference, ICSOFT 2011, Seville, Spain, July 18-21, 2011. Revised Selected Papers*, pages 121– 135, 2011.

[WKP18] Yves Wautelet, Manuel Kolp, and Loris Penserini. Service-driven iterative software project management with I-Tropos. *J. UCS*, 24(7):975–1011, 2018.

[YGMM11] Eric Yu, Paolo Giorgini, Neil Maiden, and John Mylopoulos. *Social Modeling for Requirements Engineering*. The MIT Press, 2011.