Knowledge Management Governance in Software Development Process with GI-Tropos

Vu H. A. Nguyen
LouRIM-CEMIS
Université catholique de Louvain
Belgium
Email: vu.nguyenhuynh@uclouvain.be

Manuel Kolp
LouRIM-CEMIS
Université catholique de Louvain
Belgium
Email: manuel.kolp@uclouvain.be

Yves Wautelet
Faculty of Economics and Business
KULeuven
Belgium
Email: yves.wautelet@kuleuven.be

***Abstract*— Governing Knowledge Management in a Requirements-Driven Software Processes such as (GI-) Tropos or even waterfall, V, UP-based ones allows IT managers to propose rules for efficient handling of information and resources to cope with stakeholders' requirements. On the one hand, Knowledge Management Governance in software engineering has to ensure that software organization business processes determine organizational knowledge access conditions, quality maintenance, decision making processes and means of resolving knowledge management obstacles 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 straightforward inclusion and adaptation of knowledge management governance principles into the software development life cycle. To consolidate both perspectives, this paper proposes a generic framework allowing to drive Knowledge Management in the GI-Tropos software processes.**

***Index Terms*—Knowledge Management Governance, Software Process, GI-Tropos**

I. Introduction

Software Engineering [1] is a wide knowledge area. It requires various other types of data, information, skills and know-how during the software development and operation processes [2]. It is notably devoted to implement human activities and cope with socio-intentional problems through business modeling and requirements engineering techniques at the strategic level [3]. The software development life-cycle process itself is a structure of groups of activities in which communicating and collaborating tasks are required from any user and stakeholder. At the project and organizational level, each individual's knowledge can (or must) be shared as to what knowledge management activities they do.

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 organizations intellectual capital and performance.*' [4]. The foundations of KM include infrastructure, mechanisms, and technologies [5]. KM mechanisms are organizational or structural instruments that are used to enable KM systems. KM technologies are information technologies that can be used to promote KM. Both KM mechanisms and KM technologies are supported by the KM infrastructure. From an organizational perspective, KM infrastructure includes five major components: organization culture, organization structure, organization's information technology infrastructure, common knowledge, and physical environment [5].

Common dimensions of knowledge management governance under conceptualization are leadership, organizational structure, and relational mechanisms among stakeholders [6]. The goal of knowledge management governance is to ensure that KM processes deliver value to the identified stakeholders and minimize risks related to organizational knowledge loss [7]. Moreover, governing knowledge management sets the structural principles and rules within which all the components of a knowledge management system should be deployed in the broader organizational context. Unfortunately, little specific research has been completed on KM governance in software development process, including the problem of aligning both approaches. Indeed, most studies on knowledge management governance have rather focused on more wide-ranging fields than just software engineering life cycle.

Therefore, this paper proposes a generic framework to align knowledge management rules and constraints to software processes. The framework uses strategic modeling techniques to represent the organizational setting but also governance and management structures. We will also more specifically discuss the adoption of our framework within particular processes in order to align KM governance with requirements-driven software specification.

This paper is organized as follows. Section 2 overviews our proposed software development template called Governance I-Tropos (GI-Tropos) for governing requirements-driven software process. Section 3 illustrates knowledge management governance in requirements-driven software development process. Section 4 introduces a use case for validation. Finally, Section 5 concludes the paper and points out further work.

II. GI-Tropos

Figure 1 represents the GI-Tropos [8] process in a classical iterative perspective based on a series of phases illustrated in the horizontal dimension and a series of disciplines presented in the vertical dimension.

10.18293/SEKE2018-054

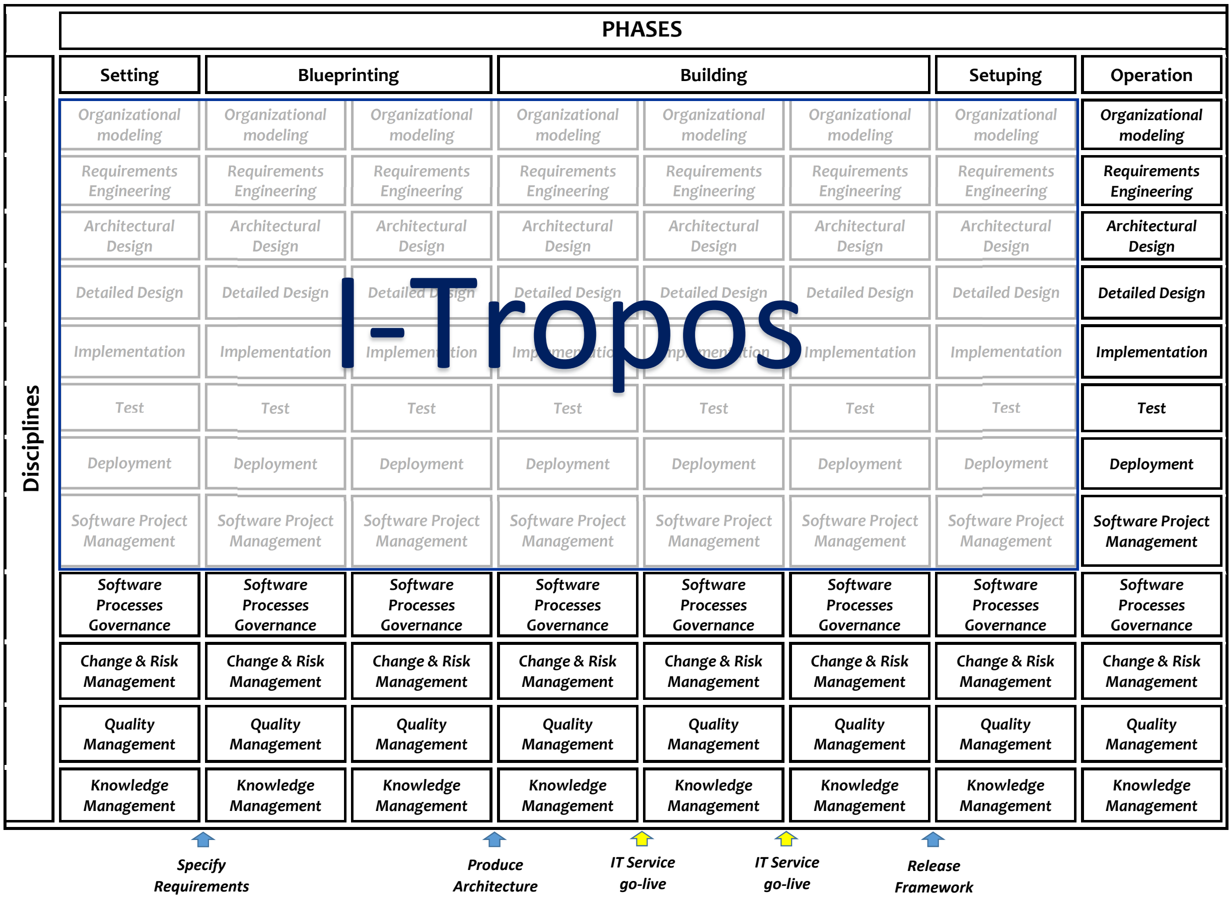

Fig. 1. GI-Tropos iterative process framework

GI-Tropos is an extension of Iterative Tropos (I-Tropos) [9], 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 [10] and deductive forward engineering (transformational) techniques, for aligning requirements-driven software processes with IT governance rules and principles. 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.

I-Tropos extends Tropos [11], a requirements-driven development methodology using the i* modeling framework [12] that supports iterative [13] and agent development [14]. 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 not only includes core disciplines (i.e., Organizational Modeling, Requirements Engineering, Architectural Design, Detailed Design, Implementation, Test and Deployment) but also supports disciplines to handle Risk Management, Time Management, Quality Management and Software Process Management [15]. 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 governance level.

From a systems development perspective, GI-Tropos improves and redefines the four following phases (Setting, Blueprinting, Building, Setuping) of I-Tropos plus a new one, Operation, to operate the system in the perspective of IT enterprise governance and management. GI-Tropos also includes all I-Tropos disciplines plus four new ones: Software Processes Governance, Change & Risk Management, Quality Management, Knowledge Management. These new disciplines ensure that software processes are evaluated, directed and monitored to meet stakeholders' requirements and achieve value added by aligning requirements-driven software processes with IT governance rules and constraints. They also enable identifying, analyzing and assessing changes and risks as well as developing strategies to manage them. Moreover, these disciplines ensure that quality expected and contracted with stakeholders is achieved throughout the system. Finally, they enable acquiring, storing and utilizing knowledge for such things as problem solving, dynamic and deep learning, strategic planning, decision making and business processes. GI-Tropos' disciplines are grouped 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.

III. KM Governance in GI-Tropos

This section describes knowledge management governance rules and constraints in a requirements-driven software processes. Figure 2 summarizes the alignment while Figures 3 and 4 illustrate the Strategic Dependency model and Strategic Rationale model for the Software Processes KM governance.

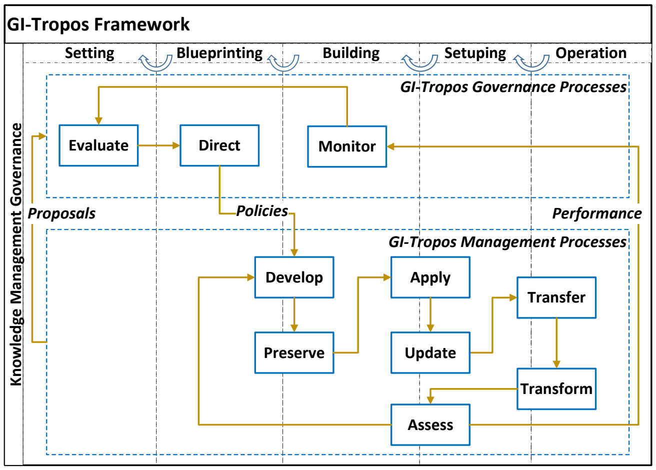

Fig. 2. GI-Tropos Knowledge Management Governance Alignment Model

Knowledge Management Governance processes aligned with requirements-driven software processes such as GI-Tropos, as depicted in Figure 2, can be summarized as follows:

- The ***Evaluate*** process ensures that stakeholders needs, conditions and options are evaluated to determine balanced, agreed-on organizational objectives to be achieved. It allows examining and judging current and future use of knowledge, considering internal and external pressures, evaluating continuously, considering current and future business needs and objectives: competitive advantage and specific strategies.
- The ***Direct*** process enables setting direction through prioritization and decision making. It assigns responsibility, directs preparation and implementation of KM plans and policies, sets directions for KM, establishes sound behavior in KM use through policies, properly plans transition of project to operational status, encourages culture of good KM governance, directs submission of proposals identifying needs.
- The ***Monitor*** process enables monitoring performance and compliance against agreed-on direction and objectives. It allows monitoring and measuring KM performance, assures that performance is in accordance with plans and business objectives, ensures that KM conforms with external obligations (regulatory, legislation, common law, and contractual) and internal work practices.
- The ***Develop*** process ensures that 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. It covers 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.
- The ***Preserve*** process ensures that 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.
- The ***Apply*** process enables knowledge to be used, enacted, executed, exploited properly.
- The ***Update*** process enables knowledge to evolve, improve, be maintained and refreshed.
- The ***Transfer*** process ensures that 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. It identifies KM requirements, acquires the technology, and implements it within the organization's current business processes.
- The ***Transform*** process ensures 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.
- The ***Assess*** process ensures that 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. It deals with the organizational strategy in assessing its needs and whether or not the current KM system still meets the objectives for which it was designed and the controls necessary to comply with regulatory requirements. It also covers the issue of an independent assessment of the effectiveness of KM system in its ability to meet business objectives and the organizational control processes by internal and external auditors

The Strategic Dependency model, as depicted in Figure 3, has three main actors depending on each other (Implementer, Knowledge Management Development Board, Knowledge Management Governance Board), resources (Organization Culture, Organization Structure, IT Infrastructure, Common Knowledge, and Physical Environment), goals (Develops Knowledge Management Structure, Continuous Implement Knowledge Management), qualities (Organization Strategy, Knowledge Management Quality), and tasks (Knowledge Management Modeling, Knowledge Management Operation).

The Knowledge Management Governance Board decides on the processes and the environmental factors (risks, quality

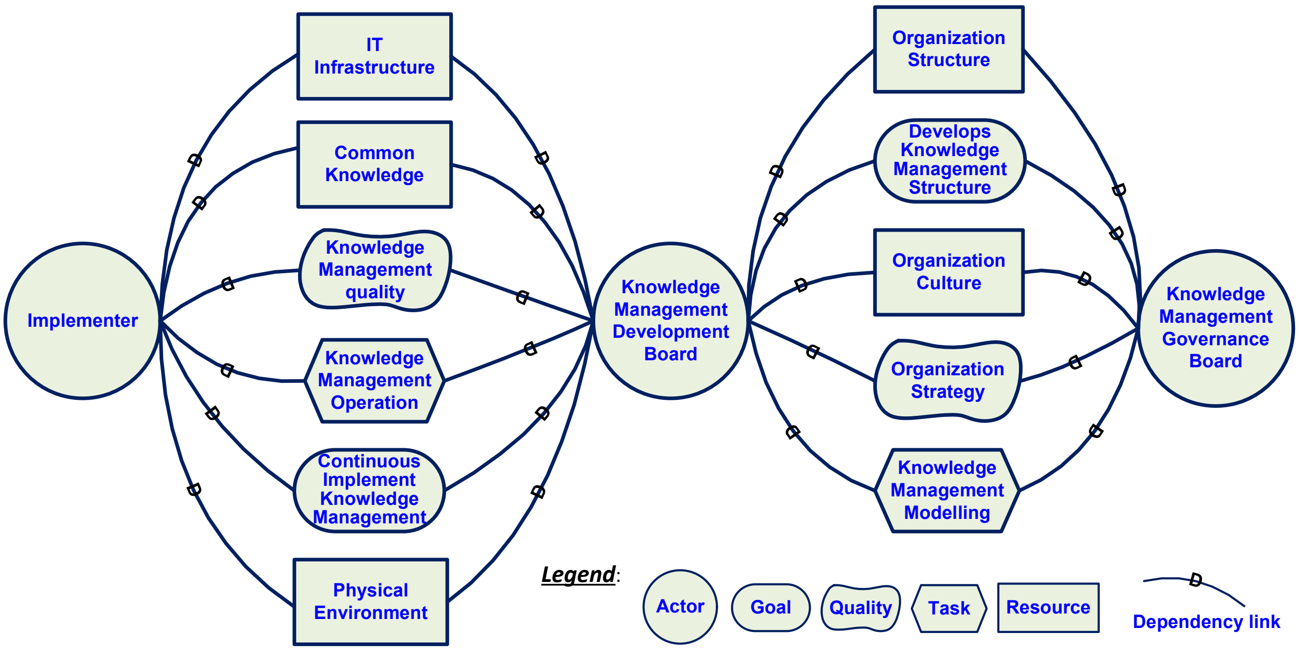
IT Infrastructure
Common Knowledge
Knowledge Management quality
Knowledge Management Operation
Continuous Implement Knowledge Management
Physical Environment
Implementer
Knowledge Management Development Board
Organization Structure
Develops Knowledge Management Structure
Organization Culture
Organization Strategy
Knowledge Management Modelling
Knowledge Management Governance Board
D
Legend:
Actor
Goal
Quality
Task
Resource
Dependency link

Fig. 3. GI-Tropos Knowledge Management Governance Strategic Dependency Model

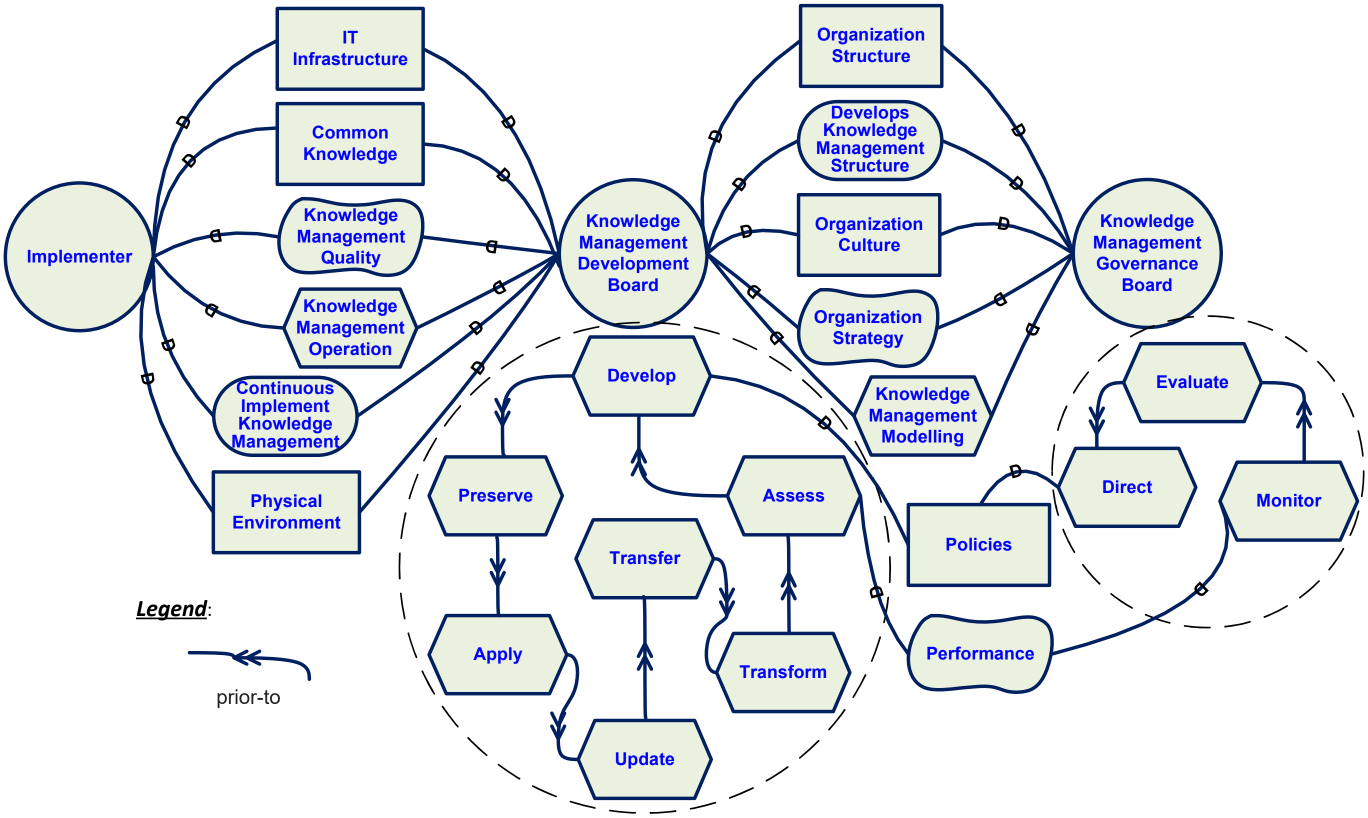
IT Infrastructure
Common Knowledge
Knowledge Management Quality
Knowledge Management Operation
Continuous Implement Knowledge Management
Physical Environment
Implementer
Knowledge Management Development Board
Organization Structure
Develops Knowledge Management Structure
Organization Culture
Organization Strategy
Knowledge Management Modelling
Knowledge Management Governance Board
Develop
Preserve
Assess
Transfer
Apply
Transform
Update
Policies
Performance
Evaluate
Direct
Monitor
D
Legend:
prior-to

Fig. 4. GI-Tropos Knowledge Management Governance Strategic Rationale Model

factors). The scope of the KM governance decisions relevant for GI-Tropos is thus only software processes. The Knowledge Management Development Board allows aligning requirements-driven software processes with KM governance. The Knowledge Management Development Board is thus a management board, not a governance one.

In the Strategic Rationale model, as depicted in Figure 4, the **Knowledge Management Governance Board** performs three tasks (Evaluate, Direct, and Monitor) corresponding to the three governance core processes (Evaluate, Direct, and Monitor) respectively. The **Knowledge Management Development Board** performs seven tasks (Develop, Preserve, Apply, Update, Transfer, Transform and Assess) corresponding to the seven management core processes (Develop, Preserve, Apply, Update, Transfer, Transform and Assess) respectively. The **Preserve** task depends on the **Direct** task based on the **Policies** resource and the **Monitor** task depends on the **Assess** task based on the **Performance** quality.

IV. Use Case

This section describes TransLogisTIC, a project used as a use case to illustrate the paper. This project developed a collaborative platform for outbound logistics. The purpose 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 project involved several complementary actors including 10 private companies and 5 universities and research labs on a 3 year basis and a 14 million euro budget.

The TransLogisTIC project was re-built with I-Tropos models for validation purpose in the context of this work. We focused on validating knowledge management governance in the software development process of an online collaborative logistic platform allowing the major outbound logistics stakeholders (chargers, carriers, infrastructure managers and final clients,...) to share information for a better optimization of the logistic chain. Firstly, we defined the knowledge areas of the software development process (I-Tropos). Then, we explained reasons why this knowledge was required. Table I represents the knowledge areas in I-Tropos based on the Guide to the Software Engineering Body of Knowledge (SWEBOK, IEEE Professional Practices Committee [16]) and explains these reasons. Finally, we aligned knowledge management governance rules and constraints with GI-Tropos's governance processes (Evaluate, Direct, Monitor) based on the workflow proposed in Figure 5.

The workflow performed during the alignment of requirements-driven software processes with knowledge management governance rules and constraints is presented in Figure 5. These tasks are assumed by the Knowledge Management Governance Board in interaction with the Stakeholders and Knowledge Management Development Board. They are sorted in three groups (Evaluating, Directing, and Monitoring) corresponding to the three core governance processes (Evaluate, Direct, and Monitor) respectively. Each task is summarized as follow:

- ***Consider***: Examine and judge current and future use of KM include strategy proposals; survey and categorize, analyze KM activities, elicit, codify and organize; consider the foundations of KM (KM infrastructure, KM mechanisms, and KM technologies); evaluate continuously; consider current and future business needs and objectives: competitive advantage and specific strategies; appraise and evaluate value of knowledge and KM.

TABLE I
Knowledge needed in I-Tropos

I-Tropos phase	Knowledge areas	Reason knowledge is needed
Setting	• Software Requirements • Software Engineering Management • Software Engineering Process • Software Engineering Models and Methods • Software Quality • Software Engineering Professional Practice • Software Engineering Economics • Computing Foundations • Mathematical Foundations • Engineering Foundations	• To justify behind the generalized model, such as forces and trade-offs, success factors associated with practices; • To establish objectives, constraints, alternatives; • To evaluate product and alternatives; • To resolve risks. • To justify behind the task plan, including risk assessments, contingency plans, management goals and criteria. • To justify behind the system, including development goals and criteria, alternatives evaluated, and their evaluation.
Blueprinting	• Software Design	• To design systems and processes definitions.
Building	• Software Construction • Software Testing	• To implement systems and processes definitions.
Setuping	• Software Configuration Management	• To validate systems and processes definitions.
Operation	• Software Maintenance	• To make sure the commitments.

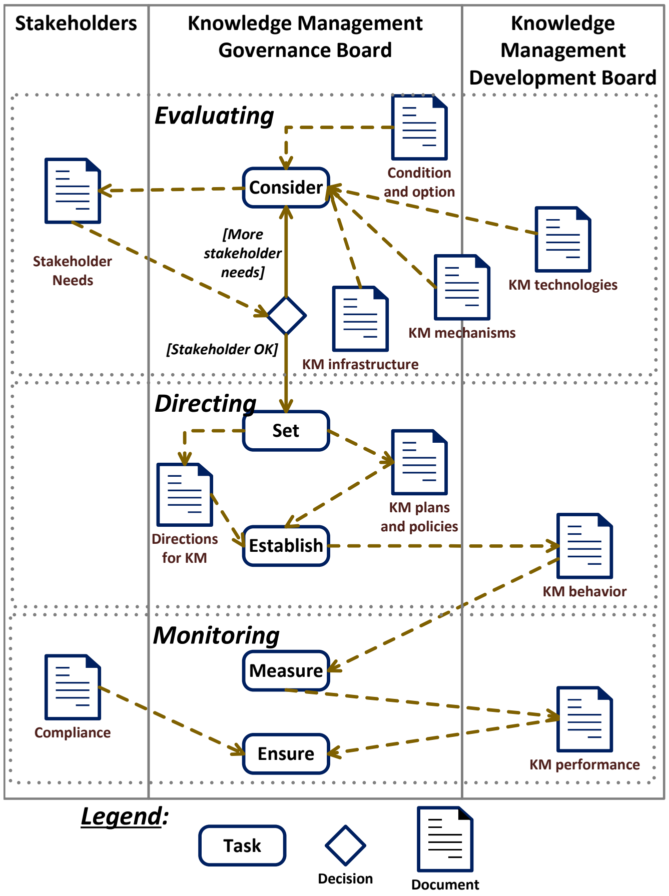

Fig. 5. GI-Tropos Knowledge Management Governance Workflow

- ***Set***: Assign responsibility and direct preparation and implementation of KM plans and policies; set directions for KM.
- ***Establish***: Establish sound behavior in knowledge use through policies; properly plan transition of project to operational status; encourage culture of good KM governance; direct submission of proposals identifying needs.
- ***Measure***: Monitor and measure KM performance.
- ***Ensure***: Assure that performance is in accordance with plans and business objectives; ensure that KM conforms with external obligations (regulatory, legislation, common law, and contractual) and internal work practices; handle, use, control, leverage, distribute and automate knowledge.

V. Conclusion

For defining software systems methods and software project methodologies, governance can be viewed as evaluating, directing and monitoring software processes all along the life cycle. Governing knowledge management in requirements-driven software processes such as GI-Tropos enables coping with stakeholders' requirements and expectations. This paper aimed at specifying integration and alignment of knowledge management governance rules and constraints to requirements-driven software processes based on the software processes knowledge areas needed by governance.

The paper presents a new identification of critical moments in the software development life cycle for knowledge management governance since the main alignment objective was to deliver an efficient KM approach that meets stakeholders' needs and expectations. On the one hand, the strengths of GI-Tropos are to systematically offer structure and direction through the whole software processes governance and enable tailoring the process to the project needs. On the other hand, GI-Tropos also points out how to establish knowledge management governance rules to the software processes. Knowledge management can be governed in software development processes by a proper alignment performed on the software processes knowledge areas that need to be handled and KM governance processes themselves. Our proposed alignment specifies how to carry out these KM processes in the context of a collaborative software development life cycle.

Further work points to other additional practices that need to be integrated in this alignment to propose a complete framework taking into consideration, for instance, IT management, project management and agile practices [17]–[19] for managing the day-to-day activities and reacting to changing requirements and feedback. In addition, a CASE tool should be developed to help designing and implementing all the processes defined in this paper.

References

[1] I. Sommerville, *Software Engineering*, 9th ed. USA: Addison-Wesley Publishing Company, 2010.

[2] K. C. Desouza, "Barriers to effective use of knowledge management systems in software engineering," *Commun. ACM*, vol. 46, no. 1, pp. 99–101, Jan. 2003.

[3] Y. Wautelet and M. Kolp, "Business and model-driven development of BDI multi-agent systems," *Neurocomputing*, vol. 182, pp. 304–321, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2015.12.022

[4] A. Jashapara, *Knowledge Management: An Integrated Approach.*, 2nd ed. Prentice-Hall, 2011.

[5] I. Becerra-Fernandez and R. Sabherwal, *Knowledge Management: Systems and Processes*. Routledge, 2015.

[6] A. Schroeder, D. Pauleen, and S. Huff, "Governance and leadership of knowledge management," in *Leadership in the Digital Enterprise: Issues and Challenges*. Idea Group Publishing, 2010, pp. 46–61.

[7] S. Zyngier, "Kowledge management governance," in *The Encyclopaedia of Knowledge Management*. Idea Group Publishing, 2005, pp. 373–380.

[8] V. H. A. Nguyen, M. Kolp, Y. Wautelet, and S. 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*, 2017, pp. 338–345.

[9] Y. Wautelet, M. Kolp, and S. 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*, 2011, pp. 121–135.

[10] PMI, *A Guide To The Project Management Body Of Knowledge*. Project Management Institute, 2013.

[11] J. Castro, M. Kolp, and J. Mylopoulos, "Towards requirements-driven information systems engineering: the Tropos project," *Inf. Syst.*, vol. 27, no. 6, pp. 365–389, 2002.

[12] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, *Social Modeling for Requirements Engineering*. The MIT Press, 2011.

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

[14] J. Mylopoulos, M. Kolp, and P. Giorgini, "Agent-oriented software development," in *Hellenic Conference on Artificial Intelligence*. Springer Berlin Heidelberg, 2002, pp. 3–17.

[15] Y. Wautelet, "A goal-driven project management framework for multi-agent software development: The case of I-Tropos," Ph.D. dissertation, Universite catholique de Louvain, 2008.

[16] I. C. Society, P. Bourque, and R. E. Fairley, *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*, 3rd ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 2014.

[17] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, 1st ed. IBM Press, 2012.

[18] P. Kruchten, "Contextualizing agile software development," *Journal of Software: Evolution and Process*, vol. 25, no. 4, pp. 351–361, 2013. [Online]. Available: http://dx.doi.org/10.1002/smr.572

[19] A. J. H. d. O. Luna, P. Kruchten, and H. P. d. Moura, "Agile governance theory: conceptual development," *CoRR*, vol. abs/1505.06701, 2015. [Online]. Available: http://arxiv.org/abs/1505.06701