Aligning Requirements-driven Software Processes with IT Governance

Vu H. A. Nguyen[1], Manuel Kolp[1], Yves Wautelet[2] and Samedi Heng[1]
[1]*LouRIM-CEMIS, Université Catholique de Louvain, Belgium*
[2]*KULeuven, Faculty of Economics and Business, Belgium*

Keywords: IT Governance, Software Process, Requirements Engineering.

Abstract: Requirements Engineering is closely intertwined with Information Technology (IT) Governance. Aligning IT Governance principles with Requirements-Driven Software Processes allows then to propose governance and management rules for software development to cope with stakeholders' requirements and expectations. Typically, the goal of IT Governance in software engineering is to ensure that the results of a software organization business processes meet the strategic requirements of the organization. Requirements-driven software processes, such as (I-)Tropos, are development processes using high-level social-oriented models to drive the software life cycle both in terms of project management and forward engineering techniques. To consolidate both perspectives, this paper proposes a process framework called GI-Tropos to extend I-Tropos allowing to align requirements-driven software processes with IT governance.

1 INTRODUCTION

Requirements Engineering and, particularly, the management of requirements through the software engineering life-cycle is closely interconnected with Information Technology (IT) Governance practices and frameworks (Kalumbilo and Finkelstein, 2014) such as CobiT (Isaca, 2012). IT Governance is derived from the underlying corporate governance objectives and reflects the alignment of IT strategy with the organization strategy to provide value-added for business (Weill, 2004). Software engineering (Sommerville, 2010) follows development opportunities corresponding to the evolution of IT and supply software to encounter demands for progressively broader and sophisticated business applications (Bannerman, 2009). Software engineering is, additionally, through requirements engineering techniques, dedicated to support human activities and to deal with socio-intentional problems (Wautelet and Kolp, 2016).

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*" (Chulani et al., 2008). 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.

In software engineering, requirements engineering tasks (van Lamsweerde, 2009) are undertaken during the early phase of the system life-cycle that enables collecting and analyzing stakeholders' needs and expectations with the aim of adequately specifying the future system. Requirements engineering includes not only understanding the needs of stakeholders and the scope of the system-to-be, but also analyzing the strategic goals and relationships of each organizational actor at different levels. It is an instrument to answer the intentions behind an organization's decisions.

Goal oriented modeling (Dardenne et al., 1993) frameworks such as i* (Yu et al., 2011; Dalpiaz et al., 2016) or KAOS (van Lamsweerde and Letier, 2002) support the expression and reasoning of the intentional aspects of goals and relations of organizational actors. For instance, the i* framework has been used for requirements engineering, business process reengineering, organizational impact analysis, and software process modeling. It focuses on the notion of intentional actor and dependency. i* consists of two main models: Strategic Dependency (SD) and Strategic Rationale (SR). The SD model represents the intentional level of a system that contains actors and dependencies in terms of Resource, Task, Goal, and Quality dependums. SR models are used to discover the rationale behind the actors, which can be explicitly described also in the same terms of Goals, Qualities, Tasks, and Resources, and relationships among them. An extension of i*, the Strategic Service (SS) model (Wautelet et al., 2008) allows the stakeholders to have a higher comprehensive view of the soft-

Nguyen, V., Kolp, M., Wautelet, Y. and Heng, S.
Aligning Requirements-driven Software Processes with IT Governance.
DOI: 10.5220/0006431303380345
In *Proceedings of the 12th International Conference on Software Technologies (ICSOFT 2017)*, pages 338-345
ISBN: 978-989-758-262-2
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

ware system for adequate understanding of IT governance and management. It enables the stakeholders to share a common comprehensive view of what the governance and management processes should propose plus their dependency relationships.

The purpose of this paper is to align requirements-driven software processes with IT governance rules and constraints by using strategic modeling organization specifications. To this end, we propose GI-Tropos, an iterative process framework for a proper governance and management of requirements-driven software processes from strategic to operational levels. In this framework, we use strategic modeling techniques to represent the organizational setting and also governance and management structures. Then we will discuss the use of these models within particular processes in order to align IT governance and management with requirements-driven software lifecycle and disciplines.

This paper is organized as follows. Section 2 overviews the proposed development template called Governance I-Tropos (GI-Tropos) for requirements-driven software process and IT governance alignment. Section 3 proposes the meta-model while section 4 overviews the alignment between requirements-driven software development process and IT governance. Finally, Section 5 concludes the paper and points out further work.

2 REQUIREMENTS-DRIVEN DEVELOPMENT & GI-TROPOS

Requirements-driven development methodologies have been presenting great interest since they satisfy the increasing complexity and flexibility fo requirements for systems development (Wautelet and Kolp, 2011). Using the i* modeling framework, Tropos, a requirements-driven development methodology (Kolp and Mylopoulos, 2001), provides guidance for the development of the systems based on agent from early and late requirements analysis through architectural and detailed design to test and implementation. The methodology relates to a formal specification language called Formal Tropos (Fuxman et al., 2004) to capture the semantics of the subject domain to the graphical models, which can be validated, by adding constraints, invariants, pre- and post-conditions (Castro et al., 2002). Tropos consists of five phases: Early Requirements, Late Requirements, Architectural Design, Detailed Design and Implementation. These phases are not following the traditional sequence of requirements analysis, design, coding, integration, and test. Thus, a Tropos phase can actually be defined as a collection of activities that are all related to a major 'area of concern' – i.e., a discipline – in iterative development such as I-Tropos described just below.

Iterative-Tropos (I-Tropos) (Wautelet and Kolp, 2016) 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 (PMI, 2013) and forward engineering (transformational) techniques. I-Tropos is an extension of the Tropos methodology that supports iterative (Kruchten, 2003) and agent development (Kolp et al., 2002; Mylopoulos et al., 2002). With the aim of being compliant with the most generic terminology, traditional Tropos phases are considered as groups of iterations that are workflows with a minor milestone. 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 (Wautelet, 2008).

We propose, hereby an extension of I-Tropos named GI-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. From a systems development perspective, GI-Tropos takes up the five phases of I-Tropos for software engineering process (i.e., Setting, Blueprinting, Building, Setuping, and Operation) but also adds up core processes of governance (i.e., Evaluate, Direct, and Monitor) and management (i.e., Plan, Deploy, Deliver, and Assess).

Figure 1 illustrates the GI-Tropos process in a classical iterative perspective. It has been built based on a series of disciplines represented in the vertical dimension and a series of phases represented in the horizontal dimension. The GI-Tropos' disciplines are grouped in each phase. Several iterations can be deployed by phase depending on each software project characteristics. Therefore, GI-Tropos' disciplines can be repeated iteratively and the effort spent on each discipline can differ from one iteration to another.

Vertically, GI-Tropos includes all disciplines of I-Tropos (Wautelet, 2008) plus four new ones; they are:

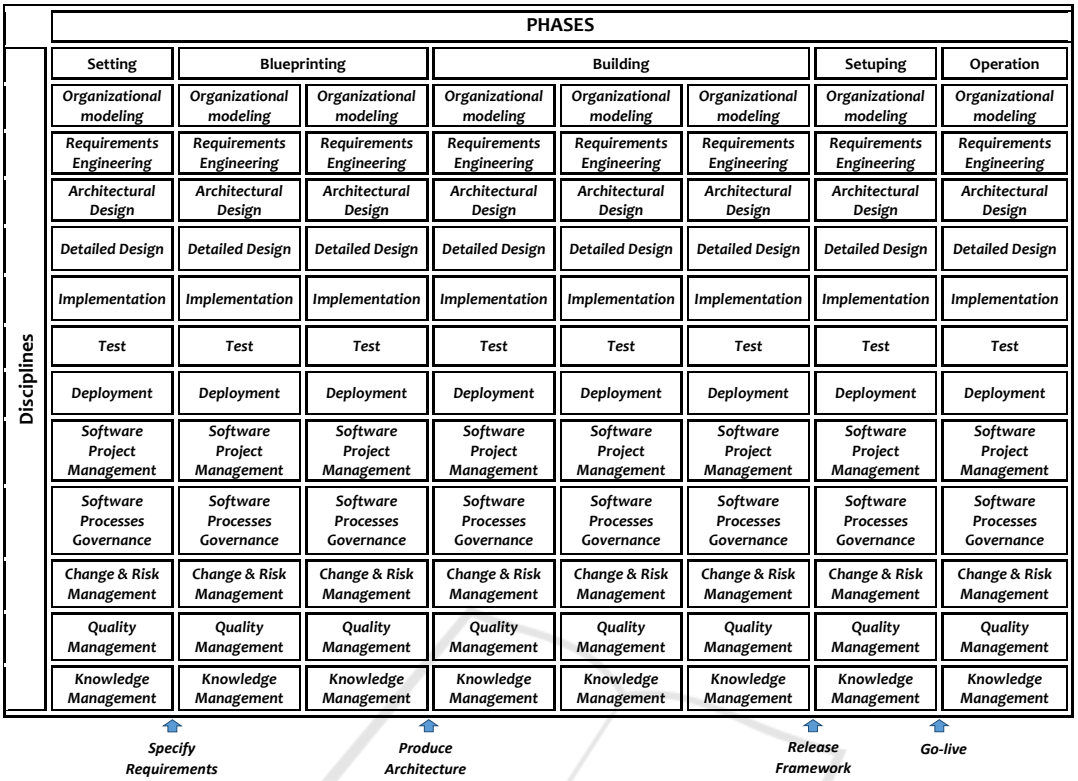

Figure 1: GI-Tropos iterative process framework.

- **Software Processes Governance:** Ensuring software processes 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;
- **Change & Risk Management:** Identifying, analyzing, assessing change and risk as well as developing strategies to manage them;
- **Quality Management:** Ensuring that quality expected and contracted with stakeholders is achieved throughout the system;
- **Knowledge Management:** Acquiring, storing and utilizing knowledge for such things as problem solving, dynamic learning, strategic planning, decision making and business processes.

Horizontally, GI-Tropos has the four following phases 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.

3 META-MODEL

The SS model in Figures 2 - 3 is used to represent all governance and management processes as services. This model has three main actors (Operator, IT Service Management Board, and IT Governance Board) and core services of governance (Evaluate, Direct, and Monitor) and management (Plan, Deploy, Deliver, and Assess). Moreover, it also illustrates threats (Risk and Resource) and quality expectations (Setting and Maintenance, Benefit Delivery, and Stakeholders Transparency) explained as follows:

- **Setting and Maintenance** represents quality expectations from analyzing and articulating requirements for governance of IT enterprise, to put in place and maintain enabling structures, principles, processes and practices, defining clear responsibilities and authorities.
- **Benefit Delivery** represents quality expectations from securing optimal value from IT enabled activities, services and assets, optimizing the contribution and value of IT to the business.
- **Stakeholders Transparency** represents quality expectations from communicating effectively and timely to stakeholders, transparently measuring

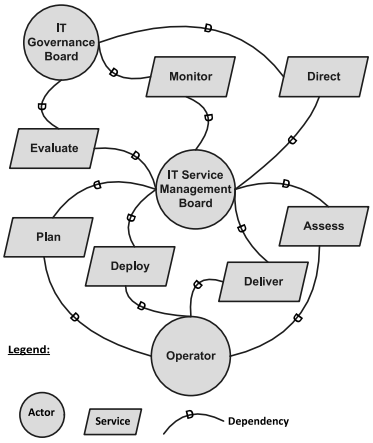

Figure 2: GI-Tropos Strategic Service model (a).

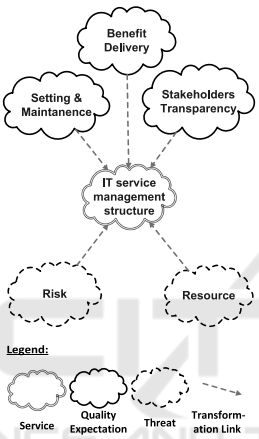

Figure 3: GI-Tropos Strategic Service model (b).

and reporting IT performance and conformance, for instance, stakeholder approval of goals and metrics.

- **Risk** represents threats from IT related risks.
- **Resource** represents threats from inadequate and insufficient people, processes and technology to support enterprise objectives.

This model allows all stakeholders to share a common aggregate view of the services including their dependency relationships. The **IT Service Management Board** is the responsible actor for governance services (Evaluate, Direct, Monitor) while the **Operator** is the responsible actor for management services (Plan, Deploy, Deliver, Assess). Each service represents a dependency graph summarized as follows:

- The **Evaluate** service represents the processes that examine and judge current and future use of IT, include strategy proposals, supply arrangements, consider internal and external pressures, evaluate continuously, consider current and future business needs and objectives: competitive advantage and specific strategies;
- The **Direct** service represents the processes that assign responsibility, direct preparation and implementation of IT plans and policies, set directions for IT investments, establish sound behavior in IT use through policies, properly plan transition of project to operational status, encourage culture of good IT governance, direct submission of proposals identifying needs;
- The **Monitor** service represents the processes that monitor and measure IT performance, assure that performance is in accordance with plans and business objectives, ensure that IT conforms with external obligations (regulatory, legislation, common law, and contractual), ensure that IT conforms with internal work practices;
- The **Plan** service represents the processes that cover 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 IT;
- The **Deploy** service represents the processes that identify IT requirements, acquire the technology, and implement it within the enterprise's current business processes;
- The **Deliver** service represents the processes that focus on the delivery aspects of the information technology. 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;
- The **Assess** service represents the processes that deal with an enterprise's strategy in assessing the needs of the enterprise and whether or not the current IT 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 IT system in its ability to meet business objectives and the enterprise's control processes by internal and external auditors.

Figure 6 presents the SR model of GI-Tropos. It is refined from the SS and the SD models. The **IT Governance Board** performs three tasks (Evaluate, Direct, and Monitor) correspond to the three core services of governance (Evaluate, Direct, and Monitor) respectively. The **IT Service Management Board**

performs four tasks (Plan, Deploy, Deliver and Assess) correspond to the four core services of governance (Plan, Deploy, Deliver and Assess) respectively. The **Plan** 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.

Figure 4 illustrates the main meta-classes and relationships of the GI-Tropos process in the context of requirements-driven software processes governance and management. The meta-classes are:

- **Service** represents the core processes of governance (Evaluate, Direct, and Monitor) and management (Plan, Deploy, Deliver, and Assess);
- **Dependency** represents all dependencies;
- **Actor** represents all actors;
- **Role** represents all roles;
- **Resource** represents all resources;
- **Task** represents all tasks;
- **Goal** represents all goals;
- **Quality** represents all qualities;
- **Threat** represents all threats;
- **Quality Expectation** represents all quality expectations;
- **Iteration** represents all iteration plans;
- **Phase** represents the five phases of the iterative software engineering process: Setting, Blueprinting, Building, Setuping, and Operation.

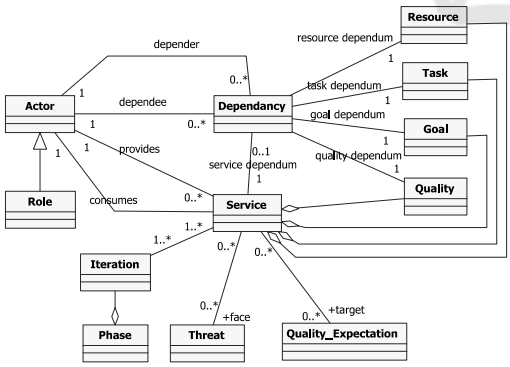

Figure 4: GI-Tropos meta-model.

The main actor function is to provide services achievement through an iteration plan. The actor has to perform a number of governance/management tasks to achieve a goal/quality with the resource dependency. Those tasks include the evaluation of factors such as risk, the limit of resource, setting and maintenance, benefit delivery, stakeholders transparency.

Figure 5 depicts the SD model of GI-Tropos. It has three main actors (Operator, IT Service Management Board, IT Governance Board), resources (Organizational structures, IT infrastructure), goals (Implementing IT management structure, Continuous operating IT services), qualities (Organization strategies, IT services quality), and tasks (Business processes modeling, IT development & operations). Moreover, it also illustrates the dependencies between actors.

In the Strategic Dependency model, the **IT Service Management Board** depends on the **IT Governance Board** to implement IT management structure based on organizational structures. The **IT Governance Board** depends on the **IT Service Management Board** to perform business processes modeling to achieve organization strategies. The **Operator** depends on the **IT Service Management Board** to operate IT services continuously based on the IT infrastructure. The **IT Service Management Board** depends on the **Operator** to perform IT development and operations to achieve IT services quality.

4 ALIGNMENT

This section summarizes aligning requirements-driven software processes with IT governance rules and constraints. In our GI-Tropos proposal, Disciplines have now to ensure the following:

- ***Organizational Modeling*** has to ensure that stakeholders agreed on direction and objectives; define project roadmap and resource constraints; identify delivery timeline and risks.
- ***Requirements Engineering*** has to identify solutions and analyze requirements to ensure that they correspond to business processes, applications, information/data, infrastructure, services; coordinate with affected stakeholders the review of feasible option and proposed solutions.
- ***Architectural Design*** has to develop and document architectural designs using agreed-on and appropriate phased or agile development techniques; ensure alignment with the IT strategy and enterprise architecture; review and update the designs when significant issues occur during detailed design or building phases or as the solution evolves; ensure that stakeholders actively participate in the design and approve each version.
- ***Detailed Design*** has to develop, document and elaborate detailed designs gradually using agreed-on and appropriate phased or agile development techniques, addressing all components; ensure

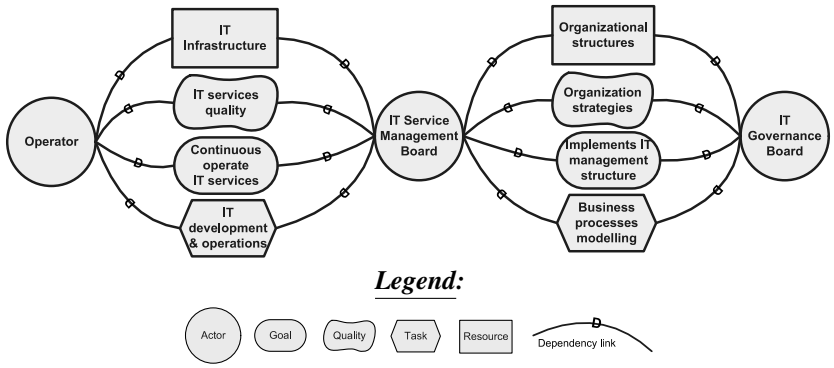

Figure 5: GI-Tropos Strategic Dependency model.

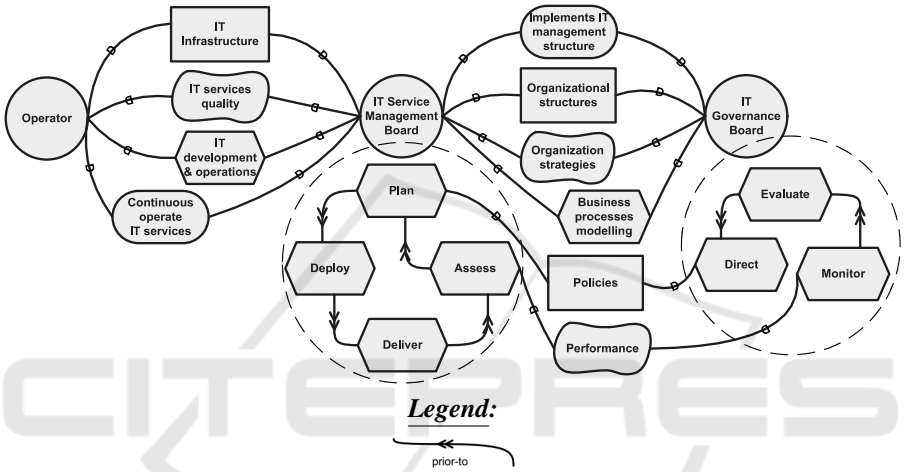

Figure 6: GI-Tropos Strategic Rationale model.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

that the detailed design includes internal and external service-level agreements and operational-level agreements.

- ***Implementation*** has to develop solution components progressively in accordance with detailed designs following development methods and documentation standards quality assurance requirements, and approval standards; ensure that all controlled requirements are addressed; install and configure solutions and integrate with business process activities; implement control, security and auditability measures during configuration; protect resources and ensure availability and data integrity during integration of hardware and infrastructural software; update the services catalog to reflect the new solutions.
- ***Test*** has to establish a test plan and required environments to test the individual and integrated solution components; execute testing continually during development, including control testing, in accordance with the defined test plan and development practices in the appropriate environment; engage business process owners and end users in the test team; identify, log and prioritize errors and issues identified during testing.
- ***Deployment*** has to track the status of individual requirements including all rejected requirements throughout the project life-cycle and manage the approval of changes to requirements; develop and execute a plan for the maintenance of the solution and infrastructure components; include periodic reviews against business needs and operational requirements; define and agree on new or changed IT services and service level options; document new or changed service definitions and service level options to be updated in the services portfolio.
- **Software Project Management** has to provide a structure of processes to direct and control the governance of the IT project including risk, quality, cost and time management.
- **Software Processes Governance** has to ensure IT governance rules and constraints alignment and mainly control the business process described in

Figure 7 explained below.

Similarly, GI-Tropos phases have also to consider new dimensions. In the Setting and Blueprinting phases, this alignment ensures controlling the operational environment, comprehending the stakeholders' requirements and expectations, gathering system requirements, defining the project scope, assessing an initial risk, and establishing an initial baseline for the software system architecture. It includes describing an information architecture, forming framework for technology planning, defining organization and processes, describing development investment framework, managing human resources, developing quality management system, developing project management framework. It also measures, estimates and minimizes development risks and plans for compliance.

In the Building phase, the alignment ensures implementing software system counterparts totally the stakeholders' requirements and expectations. It consists of managing business goals and requirements continuously, designing and developing resource, validating and measuring quality, measuring development and ongoing costs, estimating value, measuring and reviewing risk with different stakeholders based on initial prototyping result. It also manages projects based on alignment between goals and software engineering concerns.

In the Setuping phase, this alignment ensures delivering software system counterparts totally implementing the stakeholders' requirements and expectations. It comprises monitoring and managing effort and other metrics to enable control and future planning, managing applications and information to maximize usage and flexibility, prioritizing risks, tracking actual values of effort, encountering compliance needs.It also manages projects based on alignment between goals and software engineering concerns.

The workflow of tasks performed during the alignment of requirements-driven software processes with IT governance rules and constraints is presented in Figure 7. These tasks are assumed by the IT Governance Board in interaction with the Stakeholder and IT Service Management Board. They are sorted in three groups (Evaluating, Directing, and Monitoring) correspond to the three core service of governance (Evaluate, Direct, and Monitor) respectively. Each task is summarized as follow:

- ***Consider:*** Examine and judge current and future use of IT 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.

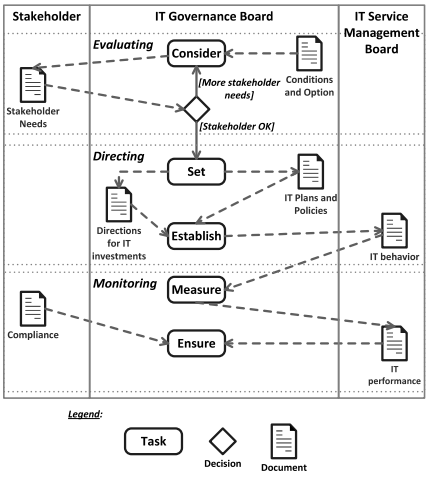

Figure 7: IT Governance Workflow.

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

5 CONCLUSION

In today's software engineering and IT project methods, the management of requirements all along the life cycle and IT governance principles are closely intertwined. Aligning them allows governance and management rules for software development to cope with stakeholders' requirements and expectations. Contributions of this paper consist of a framework including a meta-level specification to emphasize integration and alignment of requirements-driven software processes with IT governance rules and constraints. Using strategic modeling organization specifications, we have proposed an extension of I-Tropos, called GI-Tropos, customized on the application of IT governance principles to software processes driven by requirements. The main objective of this extension has been to deliver an efficient governance for software

development that meets stakeholders' needs and expectations. The strengths of I-Tropos is to offer systematically structure and direction through the whole software development process and enable tailoring the process to the project needs. However, I-Tropos does not point out how to establish governance rules to the software processes while IT governance contains the principles of governance to apply in the software processes. The GI-Tropos extension also delivers a sequence of phases for a collaborative software development environment.

Further works point to other additional practices that need to be integrated in this extension to propose a completed aligned framework taking into consideration, for instance, project management and agile practices (Ambler and Lines, 2012; Kruchten, 2013) 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 designing and implementing all the models defined in this paper.

REFERENCES

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

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

Castro, J., Kolp, M., and Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: the Tropos project. *Inf. Syst.*, 27(6):365–389.

Chulani, S., Williams, C., and Yaeli, A. (2008). 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. ACM.

Dalpiaz, F., Franch, X., and Horkoff, J. (2016). istar 2.0 language guide. *CoRR*, abs/1605.07767.

Dardenne, A., van Lamsweerde, A., and Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50.

Fuxman, A., Liu, L., Mylopoulos, J., Pistore, M., Roveri, M., and Traverso, P. (2004). Specifying and analyzing early requirements in tropos. *Requirements Engineering*, 9(2):132–150.

Isaca (2012). *Cobit 5*. ISA.

Kalumbilo, M. and Finkelstein, A. (2014). 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. ACM.

Kolp, M., Giorgini, P., and Mylopoulos, J. (2002). Organizational multi-agent architectures: a mobile robot example. In *First I. Joint Conf. on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy*, pages 94–95.

Kolp, M. and Mylopoulos, J. (2001). Software architectures as organizational structures. In *Proc. ASERC Workshop on The Role of Software Architectures in the Construction, Evolution, and Reuse of Software Systems, Edmonton, Canada*.

Kruchten, P. (2003). *The Rational Unified Process: An Introduction*. Addison-Wesley, 3rd edition.

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

Mylopoulos, J., Kolp, M., and Giorgini, P. (2002). Agent-oriented software development. In *Hellenic Conference on Artificial Intelligence*, pages 3–17. Springer Berlin Heidelberg.

PMI (2013). *A Guide To The Project Management Body Of Knowledge*. Project Management Institute.

Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition.

van Lamsweerde, A. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley Publishing, 1st edition.

van Lamsweerde, A. and Letier, E. (2002). From object orientation to goal orientation: A paradigm shift for requirements engineering. In *Radical Innovations of Software and Systems Engineering in the Future, 9th International Workshop, RISSEF 2002, Venice, Italy, October 7-11, 2002, Revised Papers*, pages 325–340.

Wautelet, Y. (2008). *A goal-driven project management framework for multiagent software development: The case of I-Tropos*. PhD thesis, Universite catholique de Louvain.

Wautelet, Y., Achbany, Y., and Kolp, M. (2008). A service-oriented framework for MAS modeling. In *ICEIS 2008 - Proc. of the 10th Int. Conf. on Enterprise Information Systems, Volume ISAS-1, Barcelona, Spain, June 12-16, 2008*, pages 120–128.

Wautelet, Y. and Kolp, M. (2011). 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.

Wautelet, Y. and Kolp, M. (2016). Business and model-driven development of BDI multi-agent systems. *Neurocomputing*, 182:304–321.

Weill, P., R. J. (2004). *IT governance: How Top Performers Manage IT Decision Rights for Superior Results, Watertown, MA: Harvard Business School Press*.

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