Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 3–10

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

An Organizational Modeling for Developing Smart Contracts on Blockchain-based Supply Chain Finance Systems

Vu Nguyen Huynh Anh [a,b,*]

[a]*Ho Chi Minh University of Banking, 36 Ton That Dam Street, District 1, 700000 Ho Chi Minh City, Vietnam*
[b]*LouRIM-CEMIS, Université catholique de Louvain, Place des Doyens 1, 1348 Ottignies-Louvain-la-Neuve, Belgium*

Abstract

Managing smart contracts in an Agile software development based on blockchain technology enables project managers to offer rules for software processes to meet the needs and expectations of stakeholders. On the one hand, smart contracts' faster cycle time, lower fraud, and lower fees and charges have ensured that the business logic satisfies the stakeholder's criteria. Agile approaches, on the other hand, are development processes that use incremental and iterative engineering methodologies to drive the system life cycle. Such methodologies are typically well-suited for incorporating and adapting management ideas early in the development life cycle. To reconcile both viewpoints, this work provides an organization modeling for constructing smart contracts on the blockchain utilizing organizational modeling techniques that allow all processes as well as the smart contracts management discipline in supply chain finance system development to be illustrated.

Keywords: Smart contract; Blockchain; Supply chain finance.

* Corresponding author.
E-mail address: vunha@hub.edu.vn / vu.nguyenhuynh@uclouvain.be

10.1016/j.procs.2024.06.139

1. Introduction

Supply Chain Finance (SCF) is one such well-liked method for improving a company's cash situation. Control, trust, and cost are a few issues that could prevent a provider from moving forward with a supply chain contract. If the level of transparency and execution speed throughout the entire supply can be greatly increased, these difficulties can be effectively handled [10], [17]. In this context, blockchain technology can offer a potential answer. Using a distributed ledger like a shared database, blockchain technology assists in cryptographically registering each transaction. These data blocks are linked together in a way that makes them permanent and unchangeable [13]. Additionally, an automated version of a contract (such as a smart contract) between a supplier and a customer or a customer and a bank increases system transparency [9]. To remove linked counterparty risk in this system, essentially no involvement of a neutral third party is necessary. Even value can be exchanged between peers via blockchain technology.

The paper uses strategic modeling organization requirements to model blockchain-based supply chain finance systems. In order to do this, we suggest an organization setting for creating smart contracts on a blockchain-based supply chain finance system. In this approach, the organizational setting and management structures are represented using strategic modeling tools. The implementation of this method inside certain processes to align the development processes with smart contracts on blockchain in the supply chain finance system.

The structure of this paper is as follows. Section 2 provides an overview of the research context, while Section 3 introduces the study technique and methods. The AgileSCF model, which is proposed in Section 4, is an agile methodology for creating smart contracts on blockchain-based supply chain finance systems. A case study is shown in Section 5 while the validation of this approach is shown in Section 6. Section 7 draws the paper's conclusion and suggests areas for future research.

2. Research Context

2.1. Organizational Modeling

Organizational modeling such as i* [20], [7] or KAOS [16], which focuses on investigating and analyzing the strategic aims, intentions, and interactions of each player in a network of actors, can be helpful in comprehending organizational problems at various levels. It serves as a tool for providing the "why" behind organizational decisions. Models that explicitly reflect and address goals can aid in understanding the "whys." [20].

The concept of intentional actor and intentional dependency is central to the i* framework. This paradigm is divided into two parts: Strategic Dependency (SD) and Strategic Rationale (SR). The SD model provides the deliberate level of a system, which includes a set of actors and their dependencies, such as resource dependency, task dependency, goal dependency, and quality dependency. The SR models are used in systems and organizations to express the rationale underlying processes, which can be clearly specified in terms of process aspects such as goals, attributes, tasks, and resources, as well as their interactions. It enables us to visualize the purposeful elements within an actor's boundary in order to develop the SD model and add reasoning capacity. The SD model's dependencies are related to purposeful elements at the actors' boundary. The SR model's elements are broken into three types of links: means-end, contribution, and task-decomposition.

Fig. 1 illustrates the main i* elements and their graphical representation used in the diagrams of the paper.

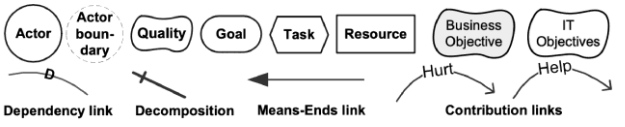

Fig. 1. i* Elements and their Graphical Representation.

2.2. Supply Chain Finance and Blockchain Technology

In relation to open account trade, which occurs when a buyer and a seller have previously conducted business together, SCF refers to the recently developed techniques and practices used by banks and other financial institutions to manage the capital invested into the supply chain and reduce the risk for the actors involved [10], [17]. Financial transactions that are connected to trade receivables finance and international commerce are mentioned by SCF. SCF is mostly used in open account trading, where the goods are shipped and delivered prior to the due date for payment. The buyer who has agreed to pay the invoice on a specific date is sent the products directly, together with any required paperwork.

Blockchain will store all the relevant data in a smart contract that can be viewed by everyone on the network and updated instantly. The transfer of ownership of products and money can be automated with the help of smart contracts. Through automation and network validation, third-party facilities like letters of credit (LCs) are no longer necessary. By doing away with the third parties and the fees they charge, will streamline the entire process and significantly cut expenses. Using smart contracts leads to faster cycle time, less fraud, and decreased fees and charges. [4]

Fig. 2 represents blockchain-based supply chain finance as a network of actors.

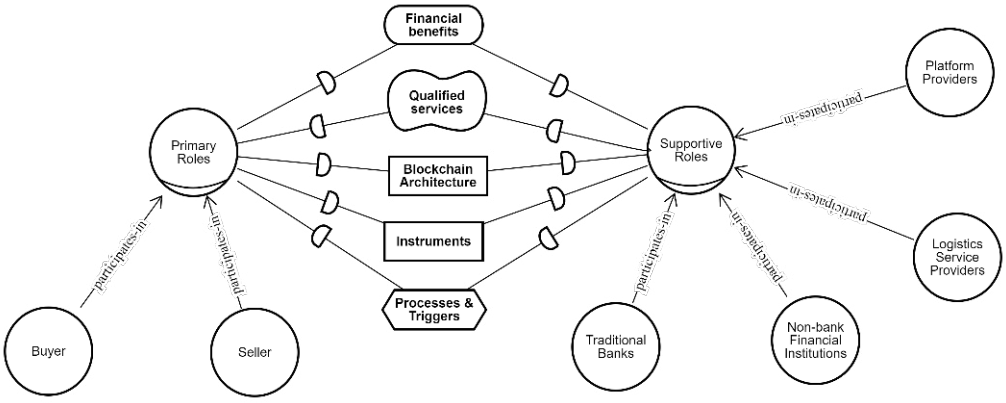

Fig. 2. Blockchain-based Supply Chain Finance as a strategic dependency diagram.

2.3. Agile Development

Agile is a term used in the *Manifesto for Agile Software Development* [5]. According to [19], the agile manifesto is "*a set of principles encapsulating the ideas underlying agile methods of software development*". Rather than a linear approach, the Agile methodology [1],[2], [14] is built on iterative and incremental development. As a result, instead of constructing a complete solution all at once, it develops the system incrementally. This system development life cycle contains five stages: *Planning*, *Analysis*, *Design*, *Building*, and *Testing*.

3. Research Approach and Methodology

3.1. Research Question

This section outlines the study question in enough detail to retain the focus of our research. We will undertake the following research inquiry after the preliminary investigation depending on the setting and objective of the research: ***How could organizational modeling for establishing smart contracts on blockchain-based supply chain finance systems be represented?***

By providing organizational modeling, project managers can offer guidelines for development processes that fulfill stakeholder requirements and expectations in terms of IT/Business alignment.

The next section depicts the steps involved in doing research.

3.2. Design Science

According to the research process described in this paper, which is based on the Design Science Research (DSR) methodology for information systems [18],[12], the initial start of the research process is the explanation of the goals of the solution: a suitable approach for developing smart contracts on blockchain-based supply chain finance systems in alignment with stakeholders' long-term strategy. The acquisition architecture steps used in our research were shown in Table 1.

Table 1. Design Science Research Methodology for Information Systems: Acquisition architecture's steps.

Steps	Description
1. Problem identification and motivation	The core problem is that a supplier might be dissuaded from pursuing a supply chain contract for a variety of reasons, including control, trust, and cost (Section 1). Due to the pervasiveness of blockchain technology and the pressure of competition for financial services organizations, open and adaptable software platforms are essential for process optimization and to provide proper IT integration;
2. Definition of the objectives for a solution	The objectives are to investigate how a supply chain finance system based on blockchain fits with an organization's long-term plan and to provide the organizational setting for its development;
3. Design and development	Our focus is mostly on the organizational setting in which supply chain finance systems based on blockchain are implemented. Using i* diagrams at different stages, we describe the blockchain-based supply chain finance system in this method as an organization made up of numerous dependent stakeholders. A comprehensive explanation of this effort is given in Section 4;
4. Demonstration	This step demonstrates how effectively operational supply chain finance processes may be aided by the supply chain finance system built on blockchain. A thorough example of this demonstration is provided in Section 5;
5. Evaluation	The organizational structure for the blockchain-based supply chain finance system will be evaluated in light of the long-term organizational strategy. The evaluation is found in Section 6;
6. Communication	In this paper, it is completed the task.

3.3. Research Approach and Method

The entire software development process is built up and validated in terms of blockchain technology and software engineering disciplines in this section, which outlines the research methodology. Fundamentally, AgileSCF broadens the agile paradigm by providing a management layer for smart contracts connected with the existing layers for software engineering. Strategic modeling techniques are used to express the organizational environment and management structures as direction for practitioners in this management layer, which is in line with the overall agile software development process.

4. Model

The generic model for describing AgileSCF is shown in this section. We depict the organizational setting for the creation of smart contracts on blockchain within the framework of agile modeling methodologies in this model. Additionally, it provides an overview of AgileSCF from both a structural and a software modeling perspective.

4.1. Overview of AgileSCF from a Software Modeling Perspective

The AgileSCF integrated paradigm for fusing smart contract development with Agile principles is explained in this section. This paradigm extends the Agile model. This extension aims to make it feasible to design an agile blockchain-based supply chain system that can satisfy stakeholders' demands and expectations about certain aspects of creating smart contracts. Fig. 3 displays the AgileSCF model. The AgileSCF model has three organizational decision levels: (i) *the strategic level*; (ii) *the tactical level*; and (iii) *the operational level*.

The long-term smart contract development strategy is recognized as the AgileSCF's strategic level objective at the strategic decision level. The top managers can benefit from this level. It considers the setting for controlling blockchain-based agile development processes. It helps the parties involved to come to a consensus on a thorough knowledge of the dependencies that smart contract development procedures should support before, during, and after development. The Strategic Dependency (SD) model of the i* modeling framework [7,[20] for acceptable representations of the tactical-level functional demands of smart contract development is made up of the actors and links that link them. By utilizing complex organizational concepts like tasks, goals, qualities, resources, and the actors who depend on them, middle managers who are in charge of achieving the strategic level's goals can profit from this level.

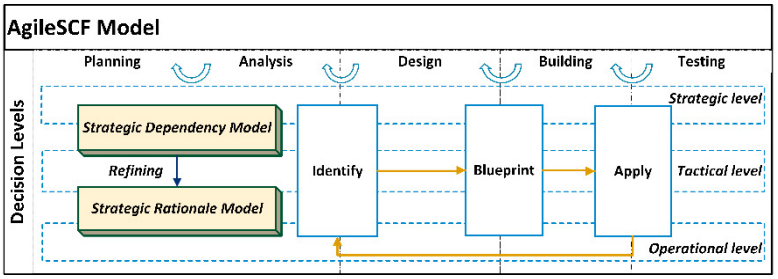

Fig. 3. The AgileSCF model.

Atomic tasks that are in charge of completing the tactically chosen sub-processes carry out each component necessary for the operational fulfillment of a smart contract. The managers and operators on the front lines can profit from this level. Following that, the tactical level models are transformed into a number of design models that are presented at this level from various angles. We recommend utilizing the Strategic Rationale (SR) model of the i* modeling framework for adequate descriptions of the functional and non-functional requirements of the services [7],[20].

4.2. Overview of AgileSCF from a Structural Perspective

Primarily, AgileSCF is structured similarly to a classic iterative approach, with a vertically presented discipline for building smart contracts and a horizontally depicted succession of phases. This discipline enables the identification of smart contract requirements, the blueprinting of smart contract assurance and control, the application of smart contract accomplishments, and the assurance that smart contract expectations and agreements with stakeholders are satisfied across the board. AgileSCF's discipline for producing smart contracts intersects with each phase. Depending on the characteristics of the software project, work may be put into each phase throughout several iterations. Therefore, it is possible to practice this discipline repeatedly, with each repetition requiring a different level of effort and workload.

In order to execute the software processes from the perspective of creating smart contracts, AgileSCF enhances the **Planning**, **Analysis**, **Design**, **Building**, and **Testing** phases of software development while augmenting those of the agile model. This alignment ensures that the operational environment is controlled, the needs and expectations of the stakeholders are understood, the system requirements are gathered, the project scope is defined, an initial risk assessment is made, and a baseline for the software system architecture is established during the **Planning**, **Analysis**, and **Design** phases. It involves establishing organization and procedures, specifying an information architecture, developing a model for technology planning, preparing for smart contracts, and developing a model for project management. The construction of the software system is fully compliant with the stakeholders' needs and expectations thanks to alignment throughout the **Building** phase. According to the results of the initial prototype, it requires continuously managing business objectives and requirements, designing and developing resources, and checking and analyzing smart contracts with various stakeholders. This alignment ensures that throughout the **Testing** phase, the equivalents of the provided software system will fully implement the requirements and expectations of the stakeholders.

AgileSCF is regarded as an all-encompassing process that can be verified by tailoring it to particular projects. Due to knowledge gained from working with several case studies, the technique needs to be refined even more. We do, in fact, assess the advantages and disadvantages of each case study to provide a generic process description. As a result,

the smart contract development paradigm is offered here in its current iteration and can be used as a stand-alone subprocess.

5. Case Study

In this section, the AgileSCF model is applied to a Supply Chain Finance application called DFinTrade. This application is an innovative solution built on the blockchain that creates a platform to assist banks and other financial institutions in managing the capital invested in the supply chain and lowering the risk for all stakeholders.

Fig. 4 depicts an SD diagram applied at the strategic level as well as at the tactical level of the AgileSCF.

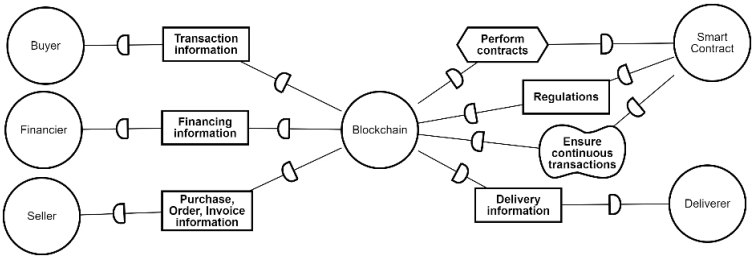

Fig. 4. DFinTrade as a strategic dependency diagram.

The SD diagram has six main actors (*Buyer*, *Seller*, *Financier*, *Blockchain*, *Smart Contract*, and *Deliverer*), resources (*Transaction information, Financing information*, *Purchase*, *Order, Invoice information, Delivery information*, and *Regulations*), and tasks (*Perform contracts*). This also illustrates the dependencies between actors.

In the SD diagram, the **Buyer** depends on the **Blockchain** to ensure the transaction information is secured. The **Seller** depends on the **Blockchain** to ensure the purchase, order, and invoice information is secured. The **Financier** depends on the **Blockchain** to ensure financing information is validated. The **Deliverer** depends on the **Blockchain** to ensure the delivery information is validated. The **Blockchain** depends on the **Smart Contract** to ensure the regulations are secured and continuous transactions quality achieved based on performing contracts.

Fig. 5 presents the SR diagram applied at the operational level of the AgileSCF. It is refined from the SD model.

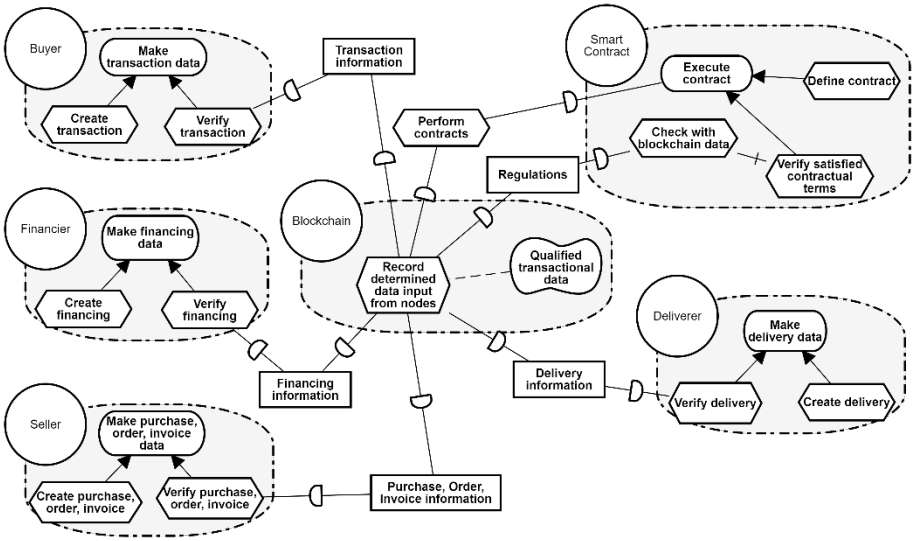

Fig. 5. DFinTrade as a strategic rationale diagram.

In the SR diagram, the **Buyer** performs two specific tasks (*Create transaction*, and *Verify transaction*) to make transaction data. The **Seller** performs two specific tasks (*Create purchase, order, invoice*, and *Verify purchase, order,*

invoice) to make purchase, order, invoice data. The **Financier** performs two specific tasks (*Create financing* and *Verify financing*) to make financing data. The **Deliverer** performs two specific tasks (*Create delivery* and *Verify delivery*) to make delivery data. The **Smart Contract** performs three specific tasks (*Define contract*, *Verify satisfied contractual terms*, and *Check with blockchain data*) to execute contracts. The **Blockchain** performs contracts by recording determined data input from nodes based on smart contracts, regulations, purchases, orders, invoices, financing, delivery, and transaction information to qualified transactional data.

We describe a strategic target as a long-term goal as a result of the organization's strategic-level strategies. The Strategic Alignment Model (SAM) from [11] distinguished two types of strategies: business strategy and IT strategy. To further the deconstruction of the strategic objectives, the strategies of the DFinTrade application based on the Non-Functional Requirements (NFR) tree [6] are shown, along with a graphical depiction of the business objectives and IT objectives (at the strategic level). DFinTrade's business strategy is centered on the following key goal: *Provide high-quality Supply Chain Finance services*. *Delivering smart contracts to support Supply Chain Finance services* is the major focus of DFinTrade's IT strategy. Fig. 6 represents the impact of the smart contracts development on the business and IT strategies.

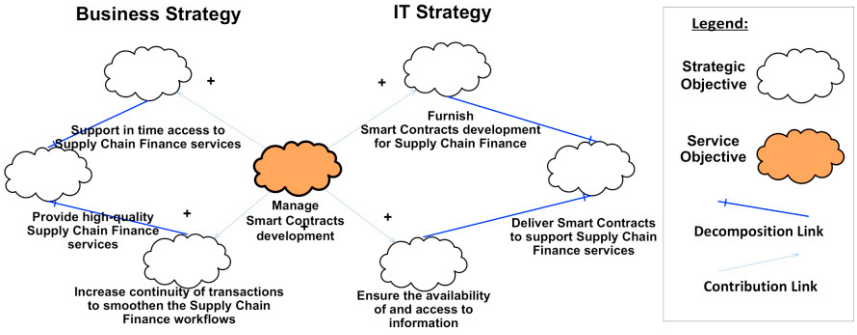

Fig. 6. IT and Business Strategies NFR Decomposition Model and Smart Contracts Development Contributions.

6. Validation

The application of the smart contracts development discipline along with the AgileSCF' phases within the DFinTrade project is illustrated as follows:

- In the **Planning** phase, the requirements of the DFinTrade application, its functions, and its range are described. At this phase, smart contracts are identified depending on the strategic business/IT objectives for the application. Based on the primary business objective of the DFinTrade project's business strategy, "*Provide high-quality Supply Chain Finance services*", the business objectives of this application have been determined. The key IT objectives of the DFinTrade project's IT strategy, "*Deliver Smart Contracts to Support Supply Chain Finance services*", are also used to identify the IT goals of this application.
- In the **Analysis** phase, the DFinTrade application is examined. The discovered smart contracts are utilized to enable the creation of the application requirements, such as the tactically-leveled smart contracts quality concept of operations and smart contracts management needs.
- In the **Design** phase, the DFinTrade application is designed. The blueprinting of smart contracts is presented. Through a realistic mock-up, the **Design** phase prototypes and further assesses the choices made for the management of smart contracts. The business/IT objectives are set during this phase. The application's business objectives are the "*Support in time access to Supply Chain Finance service*" and the "*Increase continuity of transactions to smoothen the Supply Chain Finance workflows*". The primary objectives of the application are the "*Furnish Smart Contracts Development for Supply Chain Finance*" and the "*Ensure the availability of and access to information*". At the tactical level, the smart contracts' specifications and their quality expectations are listed.
- In the **Building** phase, the DFinTrade application is acquired, created, coded, or in some other way built. Smart contracts are implemented at this stage. The **Building** phase fully carries out the management decisions made for smart contracts. The tactical realization of the business/IT objectives occurs in this phase. At the operational

level, the following concepts are identified: smart contracts assurance, smart contracts requirements, quality expectations list, and smart contracts matrix.

- In the **Testing** phase, the DFinTrade application is installed and tested. The business/IT objectives are put into practice and validated. At the operational level, smart contracts requirements and smart contracts control are implemented. The implementation of smart contracts results in the delivery of the desired quality.

7. Conclusion

To emphasize the fusion and alignment of agile approaches with smart contract development, this study proposes a model with a meta-level specification. With the help of organizational requirements from strategic modeling, we created AgileSCF, a development template focused on applying agile concepts to software development based on smart contracts. This template's main objective has been to develop smart contracts on the blockchain effectively while meeting stakeholder demands and expectations for IT/Business alignment. One of the benefits of the agile model is that it provides systematic organization and guidance across the whole software development process, enabling incremental and iterative engineering approaches to drive the system life cycle. However, the agile model does not explain how to set development standards for smart contracts, but the AgileSCF model incorporates software development concepts that can be applied to smart contracts, particularly in Supply Chain Finance. Furthermore, the AgileSCF model specifies the phases of a collaborative software development environment.

Future research indicates that other practices, such as project management and agile methods [3],[15], must be included in this model in order to present a complete template that takes into consideration, for example, managing day-to-day activities and reacting to changing requirements and input. Additionally, a CASE tool should be developed to aid in the design and implementation of all of the models outlined in this study.

References

[1] Ambler, S., Nalbone, J., Vizdos, M. (2005*) Enterprise Unified Process: Extending the Rational Unified Process*. Prentice Hall.

[2] Ambler, S. (2007) "Agile software development at scale. In: IFIP Central and East European Conference on Software Engineering Techniques", 1–12, Springer.

[3] Ambler, S., Lines, M. (2012) *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, IBM Press.

[4] Bambara, J. et al. (2018) *Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions*, McGraw-Hill Education.

[5] Beck, K. et al. (2001) *Manifesto for agile software development*.

[6] Chung, L. et al. (2000) *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers.

[7] Dalpiaz, F., Franch, X., Horkoff, J. (2016) *iStar 2.0 language guide*, CoRR, abs/1605.07767.

[8] Dingsøyr, T. et al. (2012) *A decade of agile methodologies: Towards explaining agile software development*, Elsevier.

[9] Du, Mingxiao et al. (2020) "Supply chain finance innovation using blockchain." *IEEE Transactions on Engineering Management* **67**(**4**): 1045–1058

[10] Gelsomino, L. M. et al. (2010) "Supply chain finance: a literature review." *International Journal of Physical Distribution & Logistics Management* **13**(**2**): 133-142

[11] Henderson, J. C. and Venkatraman, N. (1999) "Strategic alignment: Leveraging information technology for transforming organizations." *IBM Syst. J.* **38**(**2**): 472–484.

[12] Hevner, A. et al. (2004) "Design Sciene in Information Systems Research." *Management Information Systems Quarterly* **28**(**1**): 75–105.

[13] Hofmann, E. et al. (2017) *Supply chain finance and blockchain technology: the case of reverse securitization*, Springer.

[14] Kruchten, P. (2004) *The Rational Unified Process: An Introduction*. Addison-Wesley.

[15] Kruchten, P. (2013) "Contextualizing agile software development." *Journal of Software: Evolution and Process* **25**(**4**): 351—361.

[16] Lamsweerde, A., 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: 325–340.

[17] Moritz Leon Gomm (2010) "Supply chain finance: applying finance theory to supply chain management to enhance finance in supply chains." *International Journal of Logistics Research and Applications* **13**(**2**), 133-142.

[18] Peffers, K. et al. (2008) "A Design Science Research Methodology for Information Systems Research." *Journal of Management Information Systems* **24**(**3**): 45–77.

[19] Sommerville, Ian (2016) *Software engineering* 10th Edition, Pearson Education, England.

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