Towards the measurement of Enterprise Information Systems agility to support EIS improving projects

Jannik Laval, Chantal Cherifi, Vincent Cheutet

To cite this version:

Jannik Laval, Chantal Cherifi, Vincent Cheutet. Towards the measurement of Enterprise Information Systems agility to support EIS improving projects. International Journal of Agile Systems and Management, 2018, 11 (3), pp.222-246. 10.1504/IJASM.2018.10015581 . hal-01847787

HAL Id: hal-01847787
https://hal.science/hal-01847787

Submitted on 20 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards the measurement of Enterprise Information Systems agility to support EIS improving projects

Jannik LAVAL*

University of Lyon, University Lumière Lyon 2, DISP lab EA4570
IUT Lumière Lyon 2, 160 boulevard de l'Université
69500 Bron, France
E-mail: jannik.laval@univ-lyon2.fr
*Corresponding author

Chantal CHERIFI

University of Lyon, University Lumière Lyon 2, DISP lab EA4570
IUT Lumière Lyon 2, 160 boulevard de l'Université
69500 Bron, France
E-mail: chantal.bonnercherifi@univ-lyon2.fr

Vincent CHEUTET

University of Lyon, INSA Lyon, DISP lab EA4570
Bât Léonard de Vinci, 21 avenue Jean Capelle
69100 Villeurbanne, France
E-mail: vincent.cheutet@insa-lyon.fr

Abstract Enterprise information systems (EIS) are directly implied in the global performance of an organisation. Nevertheless, their potential rigidity in comparison with the required fast evolution of the supported organisation remains an important open research question. The proposed research work aims to define and evaluate the agility of an EIS, in order to assist both software engineers and business managers in EIS improvement projects. In particular, a framework is proposed to structure the different existing metrics on agility according to the improvements needs and the intrinsic characteristics of an information system.

Keywords: Agility, enterprise information system, metrics, improvement project

Biographical notes: Jannik Laval is an associate professor at UniversitÃ© de Lyon, France, since 2015. He received the doctorate degree in computer science from the University Lille 1, France, in June 2011. He works on agility of information system, and more particularly on modularity and tools for these systems. Previously, he worked on modularity of software systems for robotics, and on software reengineering. He is co-chair of International Workshop on Smalltalk Technologies since 2014. He is a contributor of the Pharo platform, an innovative and open-source software development environment. He is the main maintainer of Phratch, a visual programming language.

Dr. Chantal Cherifi received her PhD in computer science from Corsica University, France, in 2011. She is working as an associate professor at the DISP laboratory, Lyon 2 University, France. Her research focuses on Information Systems Agility with SOA architectures, Smart Cities and Network Science. She serves as program

committee member in several international conferences. She has published about 30 international refereed conference and journal papers.

Vincent CHEUTET is currently full professor in Information Systems and Industrial Engineering, at the French engineering school INSA Lyon and deputy director of the DISP Laboratory. He obtained his cotutelle PhD in mechanical engineering in 2006 from INP Grenoble (France) and University of Genoa (Italy). He obtained his HDR in 2012 in information system management from University of Technology of Compiegne (France). His current research works focus on design, integration and management of agile information systems (such as ERP or PLM), knowledge management and organisation transformation management.

1 Introduction

In a context where the evolution capacity of an organisation is a key factor for its success story, the control of its Enterprise Information System (EIS) dynamics is of prior interest. A large literature has proven the impact of the EIS on the global performance of a company (Liu et al., 2013; Garcia-Alcaraz et al., 2017), and several studies showed positive correlations between business-EIS alignment and organizational performance since years (Papp, 1999; Trinh-Phuong et al., 2012; Wiengarten et al., 2013).

Nevertheless, the acceleration of the organisation changes implies a modification of how EIS evolution and control have to be tackled. Classically, once the alignment is built, the environment of the organisation evolves and changes, requiring an adjustment of the alignment and its evolution over time. The evolution of the alignment along time is a complex question (Vessey and Ward, 2013) that is ranked recurrently as one of the major preoccupations of managers. Indeed, this evolution has to be managed not only from an IT point of view but also from strategical and organisational perspectives.

In this research work, our target are the existing EIS in the considered organisations with the objective of improving the performance of those EIS. Hence, we consider the problem with an EIS re-engineering point of view, rather than with a software development perspective, with needs coming from business and not from technology. As a consequence, the control of the EIS dynamics may be not so mature and has to be improved.

More precisely, we adopt a Lean vision for improvement projects, following a methodology in four main steps (figure 1 placing the first three steps):

1. Diagnostic based on performance measurements at the current situation,
2. Vision definition on expected performance level agreement,
3. Path definition between the current situation and the vision one,
4. Path control to ensure the consistency between the realised actions and the initial path.

With this perspective, EIS agility is often considered as the major key characteristics in the uncertain and ever-changing business environment that is the current reality for companies (Rasouli et al., 2017; Tarafdar and Qrunfleh, 2017). Galliers (2006) claims that the dynamic nature of the environment implies to deal with agile information systems. He highlights the need for information system strategizing in order to reach alignment. Even if the definition of agility is ambivalent in scientific literature and industry, agility can be

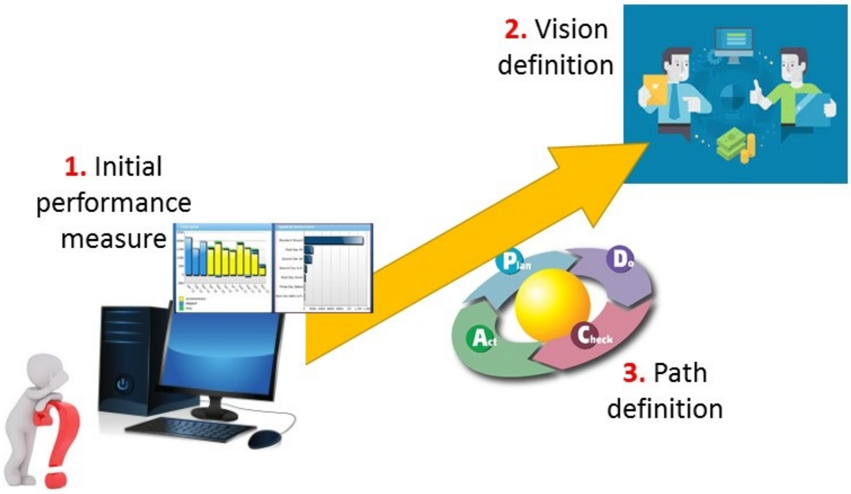

Figure 1 Improvement project global methodology

defined as: “the continual readiness of an entity to rapidly or inherently, pro-actively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment” (Conboy and Fitzgerald, 2004). Information System (IS) agility generally deals with the Information Technology (IT) problems that is to say agile IS development or IT for sensing, diagnosing and responding in real time to changes without focusing on enterprise organisation (Baskerville et al., 2011; Lowry and Wilson, 2016).

In this research work, we make the hypothesis that EIS agility can not be considered only as a characteristics of its evolution process but also as a core and intrinsic characteristics of the system. We hence increase the range of performance indicators that define and evaluate the behaviour of an EIS.

The research question under these conditions is: **how to measure agility as key performance indicator for an existing enterprise information system, with an improvement project perspective** ? This measure is required in the the global methodology at the first step to clearly and factually evaluate the initial situation and at the second step to define a shareable and reachable state. Metrics and measures are the key elements to achieve these objectives (Slayton, 2015).

In order to answer this question, we propose a framework that defines a set of potential metrics that can be used in order to evaluate the agility performance of an existing EIS. In fact, due to the diversity of EIS, of organisations and usage, a unique set of metrics cannot be defined to answer this research gap. As a consequence, the framework is designed to be modular. Indeed, according to the business needs, the EIS maturity and overall context, one can select a consistent subset of the framework to evaluate the performance and propose a vision.

The paper is structured as follow. Section 2 focuses on information system change management. In Section 3, the definition of agility is tackled. The proposed agility metrics framework is presented in Section 4. Finally, Section 5 is dedicated to discussions and perspectives.

2 Information system change management

2.1 EIS definition

From a system point of view, any organisation is defined by three layers namely operating, decision and information (Le Moigne, 1990). Consequently, more than a simple software tool, an information system also integrates the organisation layers. Shackelford et al. (2006) sketches out the conceptual territory occupied by each of the five computing disciplines: Computer Engineering, Computer Science, Software Engineering, Information Technology and Information Systems, as illustrated by Figure 2. In particular, they claim that "Information systems specialists focus on integrating information technology solutions and business processes, to meet the information needs of businesses and other enterprises, enabling them to achieve their objectives in an effective, efficient way".

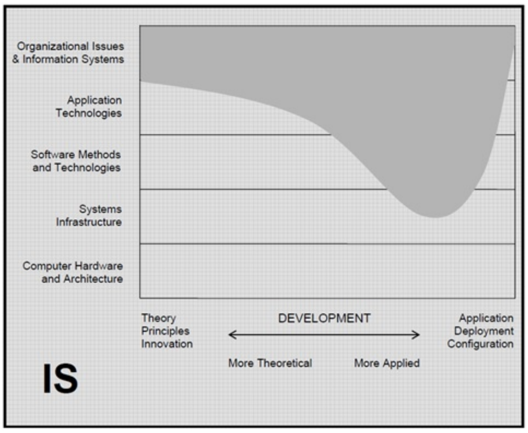
Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory Principles Innovation
DEVELOPMENT
Application Deployment Configuration
More Theoretical
More Applied
IS

Figure 2 Positioning of the information system discipline in all dimensions of computer domain (Shackelford et al., 2006)

Among all existing information systems, six particular types of EIS exist and are considered in our research work (Romero and Vernadat, 2016): Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Manufacturing Execution Systems (MES), Customer Relationship Management (CRM), Product Lifecycle Management (PLM) and Business Intelligence (BI).

2.2 Information system framework analysis

Some information system agility framework analysis found in the literature propose a practical approach to accompany entreprises in measuring and achieving agility.

Imache et al. (2012) consider different dimensions of the information system in order to differentiate the levels of analysis. The agility levels go through Business Processes, Organizational elements of the enterprise, Information, Resources that include people, Information Technology, Infrastructure, and finally Environment which deals with external factors of the enterprise. The goal is to evaluate the agility of an information system through a maturity grid. The result of the evaluation is the basis of recommendations to improve the agility of the information system.

In (Benaben et al., 2014), a mediation information system is proposed to provide a service-based platform, dedicated to initiating and supporting the interoperability of collaborative situations among potential partners. The third level of the approach tackles the system agility, i.e. the management of evolutions and changes required for the system, with a focus on the automated exploitation of incoming data to update situational models and the detection of an adaptation necessity and the characterisation of this necessity for adaptation.

Rdiouat et al. (2015) propose to evaluate the information system agility through business contribution, user orientation, operation excellence and innovation, and competitiveness, in a framework leveraging the balanced-scorecard approach. The agility is measured according to agility key success factors using agility evaluation criteria.

These propositions expose a clear understanding on how to measure agility for the dimensions under consideration. We claim that to go a step further, lowest technical levels i.e. Services and Software need to be explored as well, in order to enrich the panel.

2.3 IS alignment approaches

Most of the literature that tackles the problem of evolution of an EIS according to business needs, are dealing with the information system alignment problem. The design of the information system must be in accordance with the strategy and the supported business processes (Goepp and Millet, 2011).

According to Tarafdar and Qrunfleh (2009), there are three levels of alignment: strategic, tactical and project. Alignment at strategic level ensures that information technology plans are in accordance with business plans. It also ensures that the organisation plans to implement an information system supporting its strategic goals. The strategic alignment literature generally deals with this level. Alignment at the operational level is required for ensuring that the information system is successfully implemented, maintained and used, and in turn delivers the targeted business benefits. The project level is defined as the degree to which an information system project's deliverables match the organisation's information technology strategy and the project's objectives (Chan and Reich, 2007).

Three kinds of approaches are found in the literature regarding information system (re-)engineering: alignment building, alignment evaluation and alignment maintaining.

Alignment building approaches deal with the working out of alignment during information system design, that is to say agile information system design that are intrinsically aligned. Approaches like (Solaimani and Bouwman, 2012; Fritscher and Pigneur, 2015) formalize the models to be used to work out alignment, but they do not provide means to make evolve the alignment built.

Alignment measuring approaches like (Ahuja, 2012) enable to measure the fit between the information system and the supported business processes through evaluation of the related model elements. If the proposed metrics enable to assess the origin and kind of un-alignment, they do not cover all alignment levels, being mostly focused on tactical alignment.

Alignment maintaining approaches deal either with co-evolution model (Vessey and Ward, 2013). Most of these approaches only consider a "sticky" alignment because alignment is systematically rebuilt.

For all these reasons, the evolution of the alignment along time is a complex question that is ranked recurrently as one of the major preoccupations of managers, and few research works are facing successfully this issue.

As a conclusion, even if EIS are of prior importance for the global performance of an organisation, their intrinsic complex nature, being at the interfaces of many scientific fields (computer science, organisation, etc.), explains that controlling their evolution in a very uncertain environment remains an open question.

To overcome this difficulty, we propose to focus on agility performance of an EIS in order to provide a framework for helping decision maker in the improvement choices. The first step is to define what EIS agility can be and how it can be measured.

3 Agility definition and measurement

Agility follows a progressive and pragmatic philosophy. According to the Chambers Dictionary, the word agile is derived from the Latin "agere" that means "to act" or "whose movements are flexible and fast". The etymology of "Agile" term defines quick and slight movement.

Agility can then be addressed regarding different business competence areas, such as business agility, enterprise agility, agile organization, agile workforce, information technology agility, agile manufacturing, agile supply chains, agile software development, etc. Notably, the concept of agility is currently not exactly or uniformly defined in all those fields. Although the general objective should be the same, i.e. sustainable profitable business, the different disciplines address it from different points of view and at different levels. The disciplines also partially overlap.

To measure and assess agility, we use metrics already defined in the literature. The terms metric and performance indicator are widely used in very different areas like mathematics, management, trade links, etc.

In this research work, a metric is characterised by a semantics, related to the physical or virtual phenomena to measure, a zero value and a value scale (Böhme and Freiling, 2008). A performance indicator is based on a metric or a set of metrics focusing on a critical aspect of the global organisation performance.

Nevertheless, very few research works in the literature focus on information system agility and its measure. Since an enterprise information system is at the interface of computer science and production organisation, we focus on these two fields.

3.1 Agile manufacturing

In the context of production management, Kidd (1994) first defines agility as a timely and proactive adaptation of business elements to unexpected changes that cannot be anticipated. According to Gunasekaran (1998), Agile Manufacturing (AM) is the ability of surviving and prospering in a competitive environment of continuous and unpredictable change by reacting quickly and effectively to changing markets, driven by customer-defined products and services.

Since then, a number of literature reviews with different focuses have been realised on agile manufacturing (Sangari et al., 2015; Shaw et al., 2005; Chan et al., 2017; Fayezi et al., 2017). One can cite Charles et al. (2010), in which agility is defined as the ability of a system to be **flexible**, in a **reactive** and **efficient** manner. There are three main aspects to this vision: the system must be able to change its structure (flexibility) according to a relevant analysis of the situation and its requirements (efficiency) and this should be done in a hurry (reactiveness).

The most formal definition of agility, used in military logistics, allows us to better understand the different facets of agility in a synergistic combination of the following dimensions (Alberts and Hayes, 2003):

1. **Robustness**: the skill to maintain the efficiency through all the tasks, situations and conditions.
2. **Resilience**: the skill to improve or adjust itself against disruptive events from the environment.
3. **Reactivity**: the skill to react temporally to an environment change.
4. **Flexibility**: the skill to use a various set of means for success and the capacity to switch from one to other.
5. **Innovation**: the skill to generate novelty and the skill to make the "old" by using new means.
6. **Adaptation**: the skill to change business processes and the organisation.

In this context, some authors have proposed metrics to characterise different facets of agility (Shaarabh et al., 2014; Routroy et al., 2015; Sangari et al., 2015; Lee et al., 2017). Due to the complexity of the concept, they all propose a set of metrics that are related to physical performances of the manufacturing system or the supply chain. Some of them can be automatically calculated from data coming from ERP, CRM, etc. but do not tackle the performance of the information system support to the physical one.

Despite the differences between all these definitions, we can conclude that they all agreed on the fact that agility is an intrinsic property of the system, based on flexibility and adaptation to the unpredictability of the environment.

3.2 Agility in software engineering

When it comes to software engineering, agility is an umbrella term for a set of methods and practices on software development processes, based on the values and principles expressed in the Agile Manifesto[1]. Solutions evolve using an empirical approach through collaboration between self-organizing and cross-functional teams utilizing the appropriate practices according to the context.

Agile approaches in software engineering are created to anticipate unpredictable events that can occur during all the project phases, concerning cost, time, quality and customer needs. These four criteria are addressed in agile approaches.

Among the existing agile methods and according to the 9th Annual state of agile survey (VersionOne, 2015), Scrum (Schwaber, 2004) takes over. It is used by 56% of the companies, while other methods are behind with little use. Scrum is a development team management methodology. It provides a set of practices based on sprints to regularize teamwork in software development, in order to continue to meet the objectives in line with customer requirements.

Scrum is an incremental and iterative approach. The main objective is that when the project evolves, the customer needs can change. It is particularly true in software engineering due to the strong evolution of technologies. In practice, it requires a strong involvement of the customer in the project. Regularly, every two to four weeks, the customer is invited to validate the status of the work, and each functionality validated is delivered.

This approach has multiple advantages:

- it avoids the tunnel effect, by having the validation of the customer at real time.
- it allows the customer to change the requirements as soon as possible, and reorder the importance of functionalities.
- it allows to reduce the risk of time increase, by updating the planning regularly.

In this context, a software metric is a compilation of measurements from the technical or functional properties of the software (Mordal-Manet et al., 2013).

In the field of software development, the use of standardization is the most popular method for ensuring the agility of the information system. It allows portability of applications, by giving the ability to move between different platforms in a hybrid frame without the need for a full or partial re-engineering. To implement this approach, it is necessary to set adaptability metrics, especially with the arrival of cloud architecture (Armbrust et al., 2010) which facilitates the sharing of platform changes. But the concern remains in the interpretation of these standards by the run-time engine.

It is possible to classify the software metrics in three categories: application maintenance, application quality and respect for development process. Sherehiy et al. (2007) claim that several different approaches dedicated to agility measurement can be found in the literature, from integrated agility index to AHP methodology.

Some authors propose metrics related to the software part of the project. Beck (2000) concentrate on coding metrics and a set of unit testing, while profiling is addressed in Summers (2002). Kupiainen et al. (2015) propose a literature review on metrics in agile software development in order to identify the most influential ones and what are the effects of using metrics during software development. Lenhard (2016) propose a methodology which validates and practically evaluates a framework for measuring portability by measuring adaptability . The limit of that kind of approaches is that it merely treats problems related just to the software layer. However, for a higher vision of information systems, it does not offer the possibility to extend it.

3.3 Definition: Agility in information system

Table 1 gives some definitions of agility that are partly applied on information systems. These definitions either focus on business processes agility or on information system development agility. A limitation is their theoretical and abstract aspect, preventing them to be efficiently deployed as a support in improvement projects. These proposed definitions of agility are not complete but rather complementary. Indeed, the information system is composed of software layers and high level layers related to the business entity.

Based on these definitions, we can identify two visions of information system agility as shown in Figure 3. The first vision is the agility based on the intrinsic capability of the information system to respond to the business. It is the ability of the information system to respond to the need of the business even if there are changes in the environment.

The second vision is the capability of the information system to evolve in order to integrate new needs and new technologies to stay competitive. This agility depends on the involvement of the project team. We name it the agility in the process of improvement.

Including the two visions, we propose the following definition for information system agility: **"An Information System is agile from the moment it appropriates a set of processes and strategies that involves incrementally the user, in order to acquire the ability to adapt flexibly to the business changes and continue to achieve goals, by**

Author	Definition
Dove (2005)	*"Agile systems are concerned with response ability - for both reactive and proactive response needs and opportunities - when these are unpredictable, uncertain, and likely to change."*
Rouse (2007)	*"Agile information systems should provide sufficient flexibility to respond to business disruptions, opportunities, threats, or unforeseen events"*
Conboy (2009)	*"Continual desire for a method of the Information System Department (ISD) to quickly create or intrinsically change, pro-actively or re-actively embrace change and learn the change while contributing to the value perceived by the customer (economy, quality and simplicity) through its collective components and relationships with the environment."*
Maurer (2010)	*"IS agility as the ability of IT artefacts, of information stored within those artefacts, and of the underlying processes that support and maintain the artefacts and information to quickly adapt to changing business needs."* IS agility has three main dimensions (technical infrastructure agility, IS process agility and human characteristics), each one of them being broken down into 3-4 first-order dimensions.
Evelson (2011)	*"Approach that combines processes, methodologies, organizational structure, tools and technologies that enables strategic decision-makers to be tactical and operational for more flexibility and more responsive to the rapid pace of change in the business and regulatory requirements"*
Poonacha and Bhattacharya (2012)	*"[...] Agile has been described as the response of a company to the most competitive and uncertain environment. The uncertainty [...] manifests itself in three forms: Supply uncertainty [...] uncertain demand [...] and operational uncertainty"*
Benaben et al. (2014)	" $agility = (detection + adaptation) \times (reactiveness + efficiency)$ "
Yang et al. (2016)	*"agility cycle" (Sense - Diagnose - Select - Execute) as a means to illustrate that agility is not a one-off effort, rather, it is continuous readiness for change. We also introduce the concept of "synchronicity" to highlight the time dimension of agility.*

Table 1 Some definitions for agility in the literature.

decreasing response time, despite the pressure and the turbulent in the unpredictable environment."

According to this definition, we can define how to measure information system agility taking into account all the system dimensions.

3.4 Metrics on the process of information system improvement

Several research works have propose framework to globally evaluate a performance of an EIS as support of improvement discussion. One can cite for instance (Cho et al., 2016) in which agility measurement is one metrics used by to evaluate the trustworthiness of a

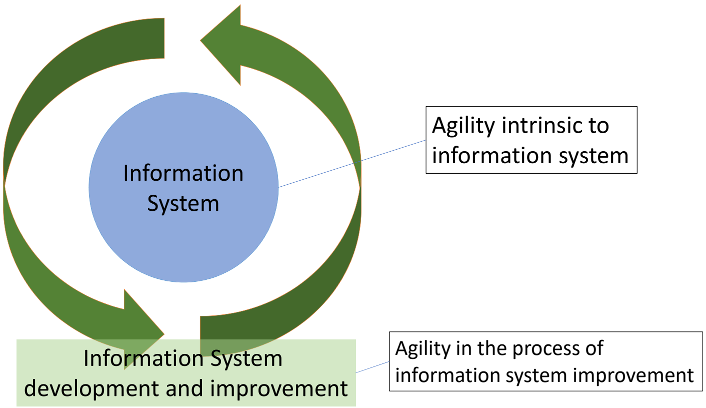
Information System
Agility intrinsic to information system
Information System development and improvement
Agility in the process of information system improvement

Figure 3 Two visions of metrics for Information System Agility

in their system, inside the TRAM (Trust, Resilience and Agility Metrics) framework. As an other example, DeLone and McLean (2016) propose a set of measure to evaluate the information system success inside an organisation. Knabke and Olbrich (2013) analyse agility for Business Intelligence systems. Anyway these frameworks only consider the global system and it will be very difficult to define an EIS improvement path with these high-level metrics only.

Tseng and Lin (2011) present a conceptual model of an agile enterprise representing the relationship between agility drivers, capabilities and providers. Based on this model, an agility development method for dealing with the interface and alignment between these three pillars is proposed. It is implemented with the Quality Function Deployment relationship matrix and fuzzy logic in order to cope with the uncertainty and the complexity of imprecise or vague definition of agile attributes and matrix relationships. In Couto et al. (2015), a conceptual model to analyze the interdependence between the governance of the information system and enterprise agility is proposed. The model is based on the identification of critical success factors enabling agility and on the identification of the information system governance practices that contribute to reach those factors.

Another approach is the combination between management methods and development methodologies as in Erickson et al. (2005), where the authors deal with metrics related to Extreme Programming in terms of agile modelling (Ambler, 2002). The model leverages a set of existing metrics in different contexts in order to bring the information system in its entirety from the software layer to the high-level layer to a level of agility.

To conclude, a notion of agility that combines intrinsic and extrinsic qualities of an information system can be proposed, with a multi-layered definition.

The literature provides a set of metrics of agility assessment but they are not sufficient. The idea is to use them according to each context to propose a model based on the existing metrics but by projecting them on a new structure adapted to the environment. The analysis of existing references shows that there are few links between the metrics of software and business process in current approaches. Moreover we can deduce from the literature that not one metric will be sufficient to measure all the dimensions of IS agility and we so need a framework to structure the selected and developed metrics.

4 Agility metrics framework

We introduce an analysis model to give a concrete vision of agility reached by an information system. In the literature, we can mention the approach developed by Imache et al. (2012) that is line with ours. The authors consider different dimensions of the information system in order to differentiate levels of analysis. The agility levels go through Business Processes, Organizational elements of the enterprise, Information stored and manipulated, Resources that include people, Information technology, Infrastructure, and finally Environment which deals with external factors of the enterprise.

The proposed model treats and describes special metrics dedicated to evaluate agility in the information system. It is inspired by the definition of agility cited in Section 2. To find answers regarding the agility of information systems, we propose a model that goes around on those aspects. Consequently, we choose a square plan which allows to visualize the objectives of the metrics. The projection on the plan allows external analysis to observe the relationship with the environment, but also internally observing correlation between these metrics.

This is an analysis model comprising metrics for evaluating agility according to 2-dimensional. The choice of two dimensions come to facilitate the judgement through the metric, because a relationship between two strangers is more relevant than several dimensions at once. system is able to adapt to the environment changes. Hence, these metrics lead us to discover the steps that must be implemented for the continuity of productivity. It shows how to reduce response time and keeping an eye on customer expectations in order to avoid the “tunnel effect”.

The first dimension of the framework is related to the levels of analysis. The idea is to differentiate the software part from the business part, in order to locate the exact source of agility problems. The second dimension deals with metrics depending on the situation we want to achieve.

Thus, evaluating the agility of an information system will be through the cross between the two dimensions, depending on the situation we want to evaluate.

4.1 Levels definition

We propose a 4-scales for the levels (figure 4):

- **Software** (SW): The software level is controlled by a set of development rules. This level represents the evaluation of the software development process and software source code quality. The source code quality can be measured based on ISO 25010 (ISO 25010, 2011) that provides a model to measure the quality of the software.

- **Services** (SV): For this level, a service is considered as a functionality in the work-flow of the information system. An agile system should be developed with service oriented architecture (SOA) that contains a set of services that can be used independently. But it is not real in all information system. So, this level represents the evaluation of the functionality of an information system to answer to the need of users.

- **Business Process** (BP): This level represents all possible trade flows and business work-flows. This level represents the association of all services needed to complete a business task, and how it can be changed depending on the environment.
- **Information System** (IS): This level includes the entire set of business management tools for information system. It represents the whole system and its capability to change a part of it without breaking other parts.

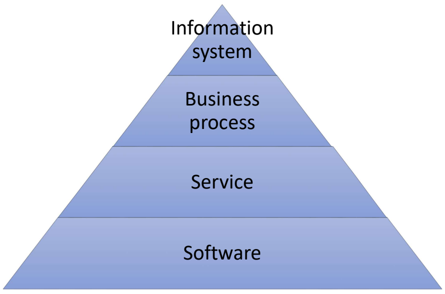

Figure 4 The four levels to measure the agility of an information system

4.2 Dimensions of agility

From our definition of information system agility, we propose to decompose the framework into five dimensions. These dimensions are listed in the first line of Table 2:

- **Appropriation Process**: It represents the flexibility desired in the system to monitor changes in the environment. This translates its inner ability to adapt appropriate behaviour in new situations without seeing the need to introduce outside interventions.
- **Adaptation to Changes**: It presents all the operations to be set up by the company using its resources to adapt to external changes.
- **Response Time reduction**: For being agile, the system has to successfully reduce the response time. Hence, it must decrease the customer wait time to access services.
- **Objectives achievement continuation**: The strong feature of agility is the ability to continuously achieve the objectives despite environmental perturbations.
- **Customer involvement**: The agile method is a user-centred approach and the customer involvement should be taken into account throughout the product lifecycle.

4.3 Metrics classification methodology

A proposition of metrics insertion in the framework is presented in Table 2. For each level and each dimension, we defined a category of metrics. In this section we will detailed theses categories and give some examples of metrics to help to implement the model.

Metric	Appropriation process	Adaptation to changes	Response time reduction	Objectives achievement continuation	Customer involvement
Information System	IS traceability Knowledge capitalization Anticipation Risk management	Evolution capacity Added value prioritization	IS response time Services availability	Bug rates Customer satisfaction	IS training Customer feedback Change acceptance
Business Process	Re-setup Process traceability	Gain evaluation Maintainability reduce Time to Market	System performance Frequency	Evaluation of Goals (AS-IS) Positive Operating Quality control	Continuous Delivery Customer Support
Services	Use flexibility Parametrisation flexibility	Extensibility Loose coupling Recomposition	Service performance Profiling service	Functional testing Scalability	Meeting frequency Transparency
Software	Use flexibility Parametrization flexibility	Integration Portability Modularization	Refactoring Code optimization Code profiling	Unit test Coverage test Scalability	Meeting frequency Validation of functionalities Usage of functionalities

Table 2 Proposition of metrics insertion in the framework for IS agility measurement.

4.3.1 Software Level

Appropriation process

- Use flexibility: It is the capability for a developer to identify and use the API (Application Programming interface) of the software source code. It means that the API methods are (i) easily accessible, (ii) well documented and (iii) useful. (i) Easily accessible means that the developer can find the APIs and differentiate them from the methods of the software-core. Depending on the language used, it can be identified by a tag, or by a type of methods. A public method is accessible by external programs. The accessibility is facilitated if a standardization of the structure is applied. (ii) Well documented is that the documentation of the API is enough to understand it and to use it. A simple metrics to measure the documentation of the API is to measure the number of lines of comments for the API. (iii) Useful is the ability to provide a benefit using this API. A possible measure of the usefulness is to analyse the number of caller of an API method. A lots of callers means that the API is useful. A refinement can be to differentiate the caller inside the software core (they use the method as a core method) and outside where they use the API as an interface as we expect to measure.

- Parametrization flexibility: It is the ability for a software to be parametrized. In Object Oriented Systems, we named it extendibility. It is the possibility of a class to be a specification of a more general class. We expect that a method can evolve independently of of the type of data sent by the caller. Metrics here are the number of generic function or the number of virtual method.

Adapting to change

- Integration: it assures that the software application is enough stable to be proposed to the final user. It implies that the integration process is well known and well controlled. It can be measured by multiple ways: the integration frequency, the number of manual operation to integrate a change, the existence of a rollback process in case of problem during the integration, the number of integration tests (ISO 15745, 2003).

- Portability: This is the ability of a software to be installed and used on multiple environments, or hybrid environments. It can be measured by the number of operation to make the system portable. Depending on the language used to develop the source code, the library used to access external resources, and the structure of the relation between the source code and the library. For example, using languages based on virtual machine, such as Java or Smalltalk, is more platform independent than a C-like source code that is compiled for a specific type of processor. Moreover, the library used by the software application is a constraint to portability, even in case of VM-based languages. The list of the depended library is essential to determine the portability (Sherehiy et al., 2007).

- Modularization: It is the capability to integrate new changes in a part of the software without high impact in the rest of the source code. It can be measured by multiple metrics, for example by computing the cohesion and coupling between software artefacts or computing the presence of god classes (Sherehiy et al., 2007).

Response Time Reduction

- Refactoring: It is the capability to make the code cleaner and less complex. It improves the extensibility, and makes it reusable by other parts of the source code. In software development, making a refactoring means that a part of source code will be changed without changing its behaviour, it improves the non-functional aspects of the source code. It is done using multiple refactoring technics (Fowler, 2002).

- Code optimization: It is the practice of reducing the occupied space by data or the program itself. The code space can be reduced using different design patterns, and detecting dead code. The practice of extreme Programming (XP) allows developers team to improve this criterion (Beck, 2003)

- Code profiling: the goal of code profiling is to measure, analyse and reduce the execution time of a function or a method. It is calculated based on source code execution time (Summers, 2002).

Continue to Achieve the Objectives

- Unit Test: It is the tests that validates each method or function. It can be measured by the ratio of passed tests on non-passed tests (Beck, 2003).

- Coverage Test: it is the ratio of source code covered by tests. It measures the quality of testing. It is measured by analysing the source code tested when running all tests (Beck, 2003).

- Scalability: it is the capability to handle a growing amount of work. It can be measured by testing the system in extreme situation. It defines the limit of a system. For example, the scalability can be defined for a number of users connected to the application.

Customer Involvement

- Meeting frequency: it is the frequency of meetings between the client and the project team, e.g. the software engineers. The goal of these meetings is the definition of new features or any changes of existing features of the software. It can be measured by the frequency of meetings dedicated to the definition of functionality between users and project team (Schwaber, 2004).

- Validation of functionalities: It is the number of functionalities that have been validated by the user. It mean that the user ask for the feature and a test has been written by the project team. It can be measured by two metrics: one that measures the number of functionalities validated by the users before the implementation. It measure the need of the functionality, one that measures the test of the functionality, that validates the use of the functionality (Schwaber, 2004).

- Usage of functionalities: A functionality can be requested by users and not be really used. There are multiple reasons: it can be because the need has changed, the HMI is not easy to use, the need was not expressed clearly, and so on. This indicator can be measured by tracking the number of call of a function during a defined period.

4.3.2 Service Level

Appropriation process

- Use Flexibility: it reflects the possibility of the use of the same services on different contexts. It can be measured by calculating the number of contexts of a service.

- Parametrisation flexibility: It expresses the variability of parameters of a service. For a same service, it measures the capability to achieve several scenarios for various uses. The measure can be a variability rate based on the ratio of configurable parameters on the fixed parameters.

Adapting to change

- Extensibility: It is the possibility of adding new parameters on a service. It can be computed by measuring the number of new settings that can be added.

- Loose coupling: The correlation between services should be minimal. If a service depends on another one, it implies that the service can become unstable if the other undergoes disturbances such as modification, unavailability. This indicator can be measured by computing the number of depended services (Channabasavaiah et al., 2003).
- Re-composition: It is the possibility of creating a new service composing multiple existing services. It can be computed by measuring the number of services using the focused service.

Response Time Reduction

- Service performance: It defines the access time to the service for users. The access time is evaluated by computing the execution of a request from the sending of the request to the receipt of the response (Menascé, 2004).
- Profiling Service: It is the capability of analysing the execution time of a service. It means that developers team is able to control and to decrease the response time to ensure to be in phase with the business. It is measured by knowing if there is a profiling system for the service. The answer is yes or no (Menascé, 2004).

Continue to Achieve the Objectives

- Functional testing: It is the number of functionalities that have been requested by the user and that has a test to ensure the validity of the behaviour regarding the business. It can be measured by the ratio of passed functional tests on non-passed tests.
- Scalability: it is the capability to handle a growing amount of work. As at software level, it can be measured by testing the system in extreme situation. It defines the limit of a service. For example, the scalability can be defined for a number of users connected to the service.

Customer Involvement

- Meeting frequency: it is the frequency of meetings between the client and the project team, e.g. the software engineers. The goal of these meetings is the definition of new services or any changes of existing services of the application. It can be measured by the frequency of meetings dedicated to the definition of services (Schwaber, 2004).
- Transparency: The team must build a relationship of trust between the customer and the project team. The project team should show the project progress and provide frequently the finished services. The customer has to fix and respect the specifications already provided. It can be measured by the number of notifications sent by the client or the project team to the other part (Schwaber, 2004).

4.3.3 Business process level

Appropriation Process

- Re-setup: Re-setup is the action of re-tuning some parameters of a business process. The goal is to make a Business Process suitable for various applications in order to achieve several scenarios. The more a business process has the ability to be re-setup, the more it can be qualified as agile. We can evaluate this indicator by the ratio of the number of configurable parameters to the total number of parameters (Ambler, 2002; Raschke, 2010).

- Process traceability: Traceability is the ability to trace the activities of the life of a business process. The goal is to utilize the knowledge captured and stored within traces. From these traces, one may be able to generate workflows and to improve the process of chaining. Traced Experience Based Reasoning (Mille et al., 2013) is a possible way for gauging the level of traceability (Beck, 2000; Arteta and Giachetti, 2004).

Adapting to Change

- Gain evaluation: Regarding changes that may occur, one may be able to evaluate the gain and to keep it at the highest level. To do so, the assessment of the business drivers must be performed at the beginning of the project. The evaluation of the gain can be computed in two ways. The first one is the computation of the cost difference between incurred and planned resources. The second one is the quotation of the suppliers in terms of cost, delay and quality (Ambler, 2002).
- Maintainability: Maintainability is the ability for components or applications to be maintained consistently and cost-effectively. This indicator is computed by various ratios, ratios between maintenance costs and the general performances of the process, ratios for maintenance policy and assets management, and ratios that express the availability of the process during maintenance operations (Beck, 2000).
- Reduce time to market: The Reduce time to market indicator is related to a product creation approach that proposes to reduce the risks and the impacts of failures by rapid iterations. It can be evaluated by various aspects of the DevObs paradigm like automation of the provision of the resource, continuous deployment (Erickson et al., 2005).

Response Time Reduction

- System performance: The performance of a system is linked to the workflow execution time. By calculating the time that consumes a workflow, one can get a measure of the system performance (Cohen et al., 1996).
- Frequency: Frequency is related to the processing of the clients requests. It expresses the dynamics of the system. The evaluation can be performed by the calculation of the number of output files (Couto et al., 2015).

Continue to Achieve the Objectives

- Evaluation of Goals (TO-BE): The evaluation of the goals is related to an actual representation of the existing business processes (AS-IS), and the review of those existing processes regarding the planed objectives (TO-BE). In order to express the differences between AS-IS and TO-BE, one can evaluate the gap between them.
- Positive Operating: In order to keep or to reach positive operating, operating assessments must be organized within a company. A differential of the performance can be made to evaluate this indicator.

- Quality control: Quality control is the process allowing to review the quality of business processes. The results of the control enable a company to determine the level of a process efficiency, and to adapt its strategy. Measures of quality are the evolution of default rates and of associated costs, prevention costs, time for the treatment of the problems.

Customer Involvement

- Continuous Delivery: Continuous delivery relates to the ability during production or exploitation, to build, test, and release software faster and frequently. This indicator can be evaluated by the rate of iterations, the number of versions.
- Customer Support: Customer support consists in providing assistance to customers in case of trouble after product delivery. It can be evaluated by claim rate and reactivity rate (Tseng and Lin, 2011; Hepp et al., 2005).

4.3.4 Information system level

When it comes to the information system level, a strong interaction between the user perception of the intrinsic capabilities, and his/her real ability to interact with the system exists. Moreover, this level is directly linked to the global business model of the organisation. As a consequence, some metrics are more conceptual, less easily implementable, and require measures outside of the system with consistent complementary tools (Galliers, 2006; Tseng and Lin, 2011; Couto et al., 2015; Heikkilä et al., 2016).

Appropriation process

- IS traceability: IS traceability is the capability of the system to track the origin of each piece of information and its evolution in the process. The goal is to ensure information consistency to all the customers. It can be evaluated, for example, by measuring the number of empty metadata, the number of information owner, by considering the various components of an Enterprise Resource Planning (ERP) system.
- Knowledge capitalization: Knowledge capitalization is the capability of an information system to easily store and retrieve knowledge and best practices. An associated measure can be the number or the rate of added, read, used or changed of such identified objects inside the information system.
- Anticipation: Anticipation describes the level of autonomy of the system and its ability to be reactive and to make decision. The system must be able to leverage tools related to data-mining and big data such as Extract-transform-load (ETL), OnLine Analytical Processing (OLAP). The aim is to improve the decision making process.
- Risk management: Risk management is the capability of the controlling system to identify potential dysfunctions, prevent them or define correction processes with consistent identification controls.

Adapting to change

- Capacity of Evolution: Capacity of Evolution is the capability of the system to successfully be modified according specifications requested by users. An example of measure is the number of successful change order on the total number of change requests.

- Added value prioritization: The prioritization of value-added solutions is the capability of the system to choose the modifications really needed by the user. It also reflects the rigidity of the system by counting the number of its non-usable tools or functionalities.

Response time reduction

- IS response time : The response time of an information system corresponds to the time that is necessary to satisfy a customer need. Let's mention the request/answer time. The speed of improvement metrics (Cohen et al., 1996) or the Time-to-recovery from the SCOR model (Jung et al., 2015) are good examples.

- Services availability: The availability of already existing services or new offered services is related to the delay for accessing them. It is a measure of the rate of people affected by the services proposed by an information system.

Continue to Achieve the Objectives

- Bug rates: Bugs have an impact on the ability of the system to continue to achieve its objectives. Measuring the number of bugs on the total number of users provides information about this ability.

- Customer satisfaction: Customers can express their feeling about the ability of the system to achieve its objectives. Customer feedback are directly retrieved for example by surveys or satisfaction questionnaires.

Customer Involvement

- IS training: System training sessions are of prime importance for the adoption of an information system by organisations. The number of proposed sessions, the number of followed sessions are possible measures to evaluate the ability of the system to be adopted.

- Customer feedback: Customer communication with the support team reflects its involvement. The quality and efficiency of this communication can be measured by the number of exchanged emails, the issue tracking system log.

- Acceptance of change: Both the customer and the provider are implied in the process of change acceptance. Acceptance of change depends on multiple factors: the type of change, the initiator of the change, the support for change and the delay of change.

4.4 Use of the framework

In this framework, we define the two dimensions that structure all metrics that can be applied to evaluate the agility of an EIS.

Nevertheless, according to the specific context of improvement project, none all of them are required in order to both define the initial situation and the vision. For instance, a SME, specialised in mechanical part production and which wants to change or update its ERP to improve its production efficiency, will focus on the business and information systems levels, since it may not have internal competencies in order to modify the software.

On the contrary, a software developer will need to implement on each level, each dimension one or multiple metrics. Based on the defined metrics, one can identify the lacks of his/her system and so improve it.

We provide the Table 2 as an example of the use of the framework. It is important to consider that each system and each context is different. Each situation has its own goal. These goals have to be transform in metrics and in threshold values. These metrics and threshold are used in the framework.

5 Conclusion and Perspectives

Enterprise Information Systems have a strong impact on the global performance of an organisation and the difficulties to make them evolve at the same pace than the organisation strategy are important sources of wastes. To tackle this research gap, we propose to integrate agility as a key characteristic of an EIS, in order to measure, control and improve this performance indicator.

In our approach, we do not adopt a bottom-up approach, which is classical in computer science (i.e. from the source code to the process), but a top-down approach. In fact, the requirements are derived from the strategy and the business processes. The EIS has to evolve to be aligned with their evolution in an uncertain context.

For this end, we first propose an holistic definition of the agility concept applied to EIS, combining both intrinsic and extrinsic capabilities of the system and its dynamics. This definition mixes the two main domains that interact in an EIS, i.e. computer science and agile manufacturing.

On the basis of this definition, we propose a complete framework that structures all the possible metrics dedicated to the agility measurement, according to the four levels of an EIS (from software to information systems) and the five facets of the agility definition. By construction, according to the organisation maturity and the context of evolution, each level can be tackled independently and the different metrics can or not be combined between levels.

Today, the framework is not exhaustive in the sense that all existing or necessary metrics have not been included inside. A first ongoing research work is to face up this challenge. In parallel, an implementation of the framework has begun and is currently partially tested. Nevertheless, a complete implementation and validation will be realised in future works.

References

Ahuja, S. (2012). Strategic alignment maturity model (SAMM) in a cascading balanced scorecard (BSC) environment: Utilization and challenges. In *International Conference on Advanced Information Systems Engineering*, pages 567–579. Springer.

Alberts, D. S. and Hayes, R. E. (2003). Power to the edge: Command... control... in the information age. Technical report, DTIC Document.

Ambler, S. (2002). *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley & Sons.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4):50–58.

Arteta, B. and Giachetti, R. (2004). A measure of agility as the complexity of the enterprise system. *Robotics and Computer- Integrated Manufacturing*, 20(6):495–503.

Baskerville, R., Pries-Heje, J., and Madsen, S. (2011). Post-agility: What follows a decade of agility? *Information and Software Technology*, 53(5):543–555.

Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley professional.

Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.

Benaben, F., Mu, W., Boissel-Dallier, N., Barthe-Delanoe, A.-M., Zribi, S., and Pingaud, H. (2014). Supporting interoperability of collaborative networks through engineering of a service-based Mediation Information System (MISE 2.0). *Enterprise Information Systems*, pages 1–27.

Böhme, R. and Freiling, F. C. (2008). On metrics and measurements. In *Dependability metrics*, pages 7–13. Springer.

Chan, A. T., Ngai, E. W., and Moon, K. K. (2017). The effects of strategic and manufacturing flexibilities and supply chain agility on firm performance in the fashion industry. *European Journal of Operational Research*, 259(2):486–499.

Chan, Y. and Reich, B. (2007). IT alignment: what have we learned. *Journal of Information Technology*, 22:297–315.

Channabasavaiah, K., Holley, K., and Tuggle, E. (2003). Migrating to a service-oriented architecture. *IBM DeveloperWorks*, 16.

Charles, A., Lauras, M., and Van Wassenhove, L. (2010). A Model to Define and Assess the Agility of Supply Chains: Building on Humanitarian Experience. *International Journal of Physical Distribution & Logistics Management*, 40(8-9):722–741.

Cho, J.-H., Hurley, P. M., and Xu, S. (2016). Metrics and measurement of trustworthy systems. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 1237–1242.

Cohen, M. A., Eliasberg, J., and Ho, T.-H. (1996). New product development: The performance and time-to-market tradeoff. *Management Science*, 42(2):173–186.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3):329–354.

Conboy, K. and Fitzgerald, B. (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. In *2004 ACM workshop on Interdisciplinary software engineering research, Newport Beach, CA, USA*, pages 37–44.

Couto, E. S., Lopes, M. F. C., and Sousa, R. D. (2015). Can IS/IT governance contribute for business agility? *Procedia Computer Science*, 64:1099–1106.

DeLone, W. H. and McLean, E. R. (2016). Information Systems Success Measurement. *Foundations and Trends® in Information Systems*, 2(1):1–116.

Dove, R. (2005). Fundamental principles for agile systems engineering. In *Conference on Systems Engineering Research (CSER), Stevens Institute of Technology, Hoboken, NJ.*

Erickson, J., Lyytinen, K., and Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4):88–100.

Evelson, B. (2011). Trends 2011 and beyond: Business intelligence. agility will shape business intelligence for the next decade.

Fayezi, S., Zutshi, A., and O'Loughlin, A. (2017). Understanding and Development of Supply Chain Agility and Flexibility: A Structured Literature Review. *International Journal of Management Reviews*, 19(4):379–407.

Fowler, M. (2002). *Refactoring: Improving the Design of Existing Code*, pages 256–256. Springer Berlin Heidelberg, Berlin, Heidelberg.

Fritscher, B. and Pigneur, Y. (2015). A visual approach to business it alignment between business model and enterprise architecture. *International Journal of Information System Modeling and Design (IJISMD)*, 6(1):1–23.

Galliers, R. D. (2006). Strategizing for agility: Confronting information systems inflexibility in dynamic environments. *Agile Information Systems*, 1.

Garcia-Alcaraz, J., Maldonado-Macias, A., Alor-Hernandez, G., and Sanchez-Ramirez, C. (2017). The impact of information and communication technologies (ICT) on agility, operating, and economical performance of supply chain. *Advances in Production Engineering & Management*, 12(1):29–40.

Goepp, V. and Millet, P. (2011). Special issue: Enterprise engineering and alignment. *International Journal of Computer Integrated Manufacturing*, 24(11):971–973.

Gunasekaran, A. (1998). Agile manufacturing: Enablers and an implementation framework. *International Journal of Production Research*, 36(5):1223–1247.

Heikkilä, M., Bouwman, H., Heikkilä, J., Solaimani, S., and Janssen, W. (2016). Business model metrics: an open repository. *Information Systems and e-Business Management*, 14(2):337–366.

Hepp, M., Leymann, F., Domingue, J., Wahler, A., and Fensel, D. (2005). Semantic business process management:a vision towards using semantic web services for business process management. *e-Business Engineering*, pages 535–540.

Imache, R., Izza, S., and Ahmed-Nacer, M. (2012). An enterprise information system agility assessment model. *Computer Science and Information Systems*, 9(1):107–133.

ISO 15745 (2003). ISO/IEC 15745: Industrial automation systems and integration - open systems application integration framework. Standard, International Organization for Standardization, Geneva, CH.

ISO 25010 (2011). ISO/IEC 25010: Systems and software engineering - systems and software quality requirements and evaluation (SQuaRE) - system and software quality models. Standard, International Organization for Standardization, Geneva, CH.

Jung, K., Morris, K., Lyons, K. W., Leong, S., and Cho, H. (2015). Mapping Strategic Goals and Operational Performance Metrics for Smart Manufacturing Systems. *Procedia Computer Science*, 44:184–193.

Kidd, P. (1994). *Agile Manufacturing. Forging New Frontiers*. Addison- Wesley series in Manufacturing Systems.

Knabke, T. and Olbrich, S. (2013). Understanding Information System Agility - The Example of Business Intelligence. In *46th Hawaii International Conference on System Sciences (HICSS)*, pages 3817–3826.

Kupiainen, E., Mäntylä, M. V., and Itkonen, J. (2015). Using metrics in Agile and Lean Software Development - A systematic literature review of industrial studies. *Information and Software Technology*, 62:143–163.

Le Moigne, J. (1990). *La modélisation des systèmes complexes*. Dunod Paris.

Lee, Y. T., Kumaraguru, S., Jain, S., Robinson, S., Helu, M., Hatim, Q. Y., Rachuri, S., Dornfeld, D., Saldana, C. J., and Kumara, S. (2017). A Classification Scheme for Smart Manufacturing Systems' Performance Metrics. *Smart and sustainable manufacturing systems*, 1(1):52–74.

Lenhard, J. (2016). *Portability of Process-Aware and Service-Oriented Software: Evidence and Metrics*, volume 23. University of Bamberg Press.

Liu, H., Ke, W., Wei, K. K., and Hua, Z. (2013). The impact of IT capabilities on firm performance: The mediating roles of absorptive capacity and supply chain agility. *Decision Support Systems*, 54(3):1452–1462.

Lowry, P. B. and Wilson, D. (2016). Creating agile organizations through IT: The influence of internal IT service perceptions on IT service quality and IT agility. *The Journal of Strategic Information Systems*, 25(3):211–226.

Maurer, C. (2010). Measuring information systems agility: Construct definition and scale development. In *Proceedings of the Southern Association for Information Systems Conference*, pages 155–160.

Menascé, D. A. (2004). Response-time analysis of composite web services. *IEEE Internet computing*, 8(1):90–92.

Mille, A., Champin, P.-A., Cordier, A., Georgeon, O., and Lefevre, M. (2013). Trace-based reasoning modeling interaction traces for reasoning on experience. *Proceedings of The 26th International FLAIRS Conference*, pages 1–15.

Mordal-Manet, K., Anquetil, N., Laval, J., Serebrenik, A., Vasilescu, B., and Ducasse, S. (2013). Software quality metrics aggregation in industry. *Journal of Software: Evolution and Process, Special Issue: Software Maintenance and Reengineering (CSMR 2011)*, 25(10):1117–1135.

Papp, R. (1999). Business-IT alignment: productivity paradox payoff? *Industrial Management & Data Systems*, 99(7-8):367–373.

Poonacha, K. and Bhattacharya, S. (2012). Towards a framework for assessing agility. In *45th Hawaii International Conference on System Science (HICSS)*.

Raschke, R. L. (2010). Process-based view of agility: The value contribution of IT and the effects on process outcomes. *International Journal of Accounting Information Systems*, 11(4):297 – 313.

Rasouli, M., Ghazanfari, H., and Eshuis, R. (2017). A process aware information system to support agility in relief operations. In *13th International Conference on Industrial Engineering (IIEC 2017)*.

Rdiouat, Y., Bahsani, S., Lakhdissi, M., and Semma, A. (2015). Measuring and improving information systems agility through the balanced scorecard approach. *International Journal of Computer Science Issues*, 12(5):58–71.

Romero, D. and Vernadat, F. (2016). Enterprise information systems state of the art: past, present and future trends. *Computers in Industry*, 79:3–13.

Rouse, W. B. (2007). Agile information systems for agile decision making. *Agile Information Systems*, pages 16–30.

Routroy, S., Potdar, P. K., and Shankar, A. (2015). Measurement of manufacturing agility: a case study. *Measuring Business Excellence*, 19(2):1–22.

Sangari, M. S., Razmi, J., and Zolfaghari, S. (2015). Developing a practical evaluation framework for identifying critical factors to achieve supply chain agility. *Measurement*, 62:205–214.

Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.

Shaarabh, M., Rishi, G., and Sharma, S. (2014). A review on measurement of agility. *Industrial Engineering & Management*, 3(1).

Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., and Lunt, B. (2006). Computing curricula 2005: The overview report. In *ACM SIGCSE Bulletin*, volume 38, pages 456–457. ACM.

Shaw, N., Burgessa, T., de Mattosb, C., and Stecc, L. (2005). Supply chain agility: the influence of industry culture on asset capabilities within capital intensive industries. *International Journal of Production Research*, 43(16):3497–3516.

Sherehiy, B., Karwowski, W., and Layer, J. K. (2007). A review of enterprise agility: Concepts, frameworks, and attributes. *International Journal of Industrial Ergonomics*, 37:445–460.

Slayton, R. (2015). Measuring risk: Computer security metrics, automation, and learning. *IEEE Annals of the History of Computing*, 37(2):32–45.

Solaimani, S. and Bouwman, H. (2012). A framework for the alignment of business model and business processes: A generic model for trans-sector innovation. *Business Process Management Journal*, 18(4):655–679.

Summers, C. (2002). System and method for profiling code on symmetric multiprocessor architectures. US Patent 6,401,240.

Tarafdar, M. and Qrunfleh, S. (2009). IT-business alignment: A two-level analysis. *Information Systems Management*, 26(4):338–349.

Tarafdar, M. and Qrunfleh, S. (2017). Agile supply chain strategy and supply chain performance: complementary roles of supply chain practices and information systems capability for agility. *International Journal of Production Research*, 55(4):925–938.

Trinh-Phuong, T., Molla, A., and Peszynski, K. (2012). Enterprise systems and organizational agility: A review of the literature and conceptual framework. *Communications of the Association for Information Systems*, 31.

Tseng, Y.-H. and Lin, C.-T. (2011). Enhancing enterprise agility by deploying agile drivers, capabilities and providers. *Information Sciences*, 181(17):3693–3708.

VersionOne (2015). *9th Annual State of Agile Survey*. versionone.com.

Vessey, I. and Ward, K. (2013). The dynamics of sustainable IS alignment: The case for IS adaptivity. *Journal of the Association for Information Systems*, 14(6):283–311.

Wiengarten, F., Humphreys, P., Cao, G., and McHugh, M. (2013). Exploring the important role of organisational factors in IT business value: Taking a contingency perspective on the resource-based view. *International Journal of Management Reviews*, 15(1):30–46.

Yang, H., Antunes, P., and Tate, M. (2016). Towards a unified conceptualisation of IS agility. In *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 269–275.