# An Ontology-based Approach to Enable Data-Driven Decision-Making in Agile Software Organizations

**Paulo Sérgio dos Santos Júnior[1,2], Monalessa P. Barcellos[1], João Paulo A. Almeida[1]**

[1] Ontology and Conceptual Modelling Research Group (NEMO),
Computer Science Department – Federal University of Espírito Santo
Vitória, ES, Brazil

[2] LEDS Research Group,
Department of Informatics - Federal Institute of Education, Science and Technology of Espírito Santo,
Serra, ES, Brasil

paulo.junior@ifes.edu.br, monalessa@inf.ufes.br, jpalmeida@ieee.org

***Abstract.*** *Software organizations adopt agile practices and data-driven software development aiming at a competitive advantage. This requires changes in the organization's culture, structure, and practices, which may not be easy. In addition, data items often remain spread in heterogeneous applications, adopting different semantics, which poses a barrier to integrated data usage. Therefore, data-driven decisions in agile development have been uncommon, missing valuable opportunities for informed decision-making. We aim to provide an ontology-based approach that integrates data of software-development support applications using an ontology network to enable data-driven decision making in agile software organizations.*

## 1.Introduction

Software development organizations are increasingly challenged with fast-changing and unpredictable market needs, complex and changing customer requirements, as well as pressures of shorter time-to-market. To address these challenges, many organizations have adopted agile methods (e.g., Scrum)[Karvonen et al. 2015] and use different applications to support agile practices.

The intensive use of applications in software development creates opportunities involving the various kinds of data they store, enabling data-driven decision-making. Despite the vast amount of data stored in applications, decisions related to software development are commonly based on subjective aspects. Particularly in the agile development context, the challenge is to use data to support decision-making in such a way that does not represent a bottleneck to process agility [Berntsson Svensson et al. 2019]. This can be obtained, e.g., by extracting and integrating data automatically and presenting integrated data in meaningful dashboards, without requiring extra effort from the team [Santos Júnior et al. 2021a].

One of the reasons organizations fail to leverage data stored in applications is the difficulty to access, integrate, analyze, and view data handled by heterogeneous applications. A source of difficulty for data integration is semantic heterogeneity, which can result in conflicts whenever the same information item is given divergent interpretations, a situation that may not even be detected. For addressing these problems, ontologies can be used to establish a common conceptualization about the domain to support communication and applications integration. They can be used as an interlingua to map the concepts used by different applications, enabling data and services understanding [Calhau and Falbo 2010].

To overcome these challenges, this research project proposes an ontology-based approach, called *Immigrant*[1], to integrate data stored in applications and provide integrated and meaningful data aiming at enabling data-driven decision-making in agile software organizations. Due to the complex domain that software development involves, this project stands for the need to use an ontology network to aid in integrating data from different software subdomains (e.g., Software Requirements, Coding, Software Quality) and extracting insights that support data-driven decision-making in the agile software organizations.

This paper provides an overview of the research project. Section 2 presents a brief background. Section 3 discusses related work. Section 4 addresses the research method. Section 5 presents the current state of research and outlines future work.

## 2. Background

*Traditional x Agile Development.* Traditional software development is organized sequentially, handing over intermediate artifacts (e.g., requirements, designs, code) between different functional groups in the organization. This causes many handover points that lead to problems such as time delays between handovers of different groups and amounts of resources are applied to creating these intermediate artifacts that, to a large extent, are replacements of human-to-human communication. In agile software development, cross-functional multidisciplinary teams play a central role. These teams have the different roles necessary to take a customer's need all the way to a delivered solution. Moreover, the notions of small, empowered teams, backlog, daily meetings, and time-box guide software development through shorter cycles and help bring the software development closer to the client [Bosch 2014].

*Semantic Integration.* Integration can be defined as the act of incorporating components into a complete set, conferring some expected properties [Izza 2009]. Interoperability, in turn, is the ability of application components to exchange or share data and services [Wegner 1996]. It provides two or more business entities with the ability of exchanging or sharing information and using the functionality of one another in a distributed and heterogeneous environment. In this work, the term integration is adopted in a broader sense, covering both integration and interoperability. When integration deals with the intended meaning of the concepts, we have Semantic Integration. It requires contrasting and harmonizing the conceptualizations underlying the applications to be integrated. In this setting, ontologies have an important role to play.

*Ontologies and Ontology Networks.* An ontology is a formal, explicit specification of a shared conceptualization [Studer et al. 1998]. For large and complex domains, representing the domain conceptualization as a single ontology result in a large and

[1] The name *Immigrant* was inspired by the *Immigrant Song* by the Led Zeppelin band because the approach allows organizations to migrate from a decision-making process not based on data-driven (*land of ice and snow*) to a data-driven decision-making process (*western shore*).

monolithic ontology, that is hard to manipulate, use, and maintain. On the other hand, representing each subdomain in isolation is a costly task that leads to a very fragmented solution that is again hard to handle [Ruy et al. 2016]. Hence, ontologies can be organized in an ontology network, which consists of a set of ontologies connected to through relationships (e.g., dependency and alignment) in such a way to provide a comprehensive and consistent conceptualization [Suárez-Figueroa et al. 2012].

## 3. Related Work

In the literature, there are several works reporting the experience of organizations when implementing agile practices. There are also works proposing data-driven approaches in software development in general.

In the context of providing data to support decision-making in software development, some works propose mining software repository solutions. As examples, we can cite [Mattila et al. 2017] and [Destefanis et al. 2016]. Different from our work, these works are not concerned with semantic aspects. Neglecting semantic aspects can lead to conflicts whenever the same information item is given divergent interpretations. Our work proposes the use of an ontology network to assign semantics to information items in applications and structure the repository of the integration solution. Since the repository is created based on the ontologies, it can be used to integrate different applications addressing similar scope, while mining software repository solutions are often created for extracting data from specific applications.

Some works addressing software process improvement have also proposed the use of integrated data to support decision-making. [Kleebaum et al. 2018] and [Johanssen et al. 2019] are some of them. Like our work, these works extract and integrate data from applications to support decision-making. However, similarly to the previously cited works, they do not address semantic aspects. Moreover, none of the cited works address data integration in the agile development context.

## 4. Research Method

The research method adopted in this work follows the Design Science Research (DSR) paradigm, which concerns extending "*human and organizational capabilities by creating new and innovative artifacts*" [Hevner 2007]. It comprises the following steps [Peffers et al. 2007]: (i) *Problem identification and Motivation*, (ii) *Definition of the objectives for a solution*, (iii) *Design and Development*, (iv) *Demonstration*, (v) *Evaluation*, and (vi) *Communication*. These steps are organized in an iterative process, with three cycles: *Relevance Cycle*, *Design Cycle,* and *Rigor Cycle.*

A DSR project begins with the *Relevance Cycle*, which involves defining the problem to be addressed, the requirements, and the criteria for evaluating the results [Hevner 2007], including the steps *(i)* and *(ii)*. The *Design Cycle* involves developing and evaluating artifacts or theories to solve the identified problem [Hevner 2007], comprising steps *(iii)*, *(iv)*, and *(v)*. Finally, the *Rigor Cycle* refers to using and generating knowledge [Hevner 2007], comprising the step *(vi)* and the use of knowledge and foundations along with the work.

## 5. Current State of the Research and Future Work

In this section, we describe the current state of the work by following the steps of the adopted research method that have been already executed. At the end, we identify future work that must be pursued to conclude the research project.

In the *Problem identification and motivation* step, the problem was identified from the literature (e.g. [Karvonen et al. 2015], [Berntsson Svensson et al. 2019], and [Kasauli et al. 2020]) and it was also observed by the first author when working as Scrum master and consultant in some organizations. The problem refers to the need for integrating applications considering semantic issues to provide useful information that enables data-driven decision-making in agile software organizations. Thus, in the *Definition of the objectives for a solution* step, we decided to develop *Immigrant*, an ontology-based approach to enable data-driven decision-making in agile software organizations. As requirements of the proposed approach, we defined that it must: (*R1*) allow identifying organization´s information needs that are important to support decision-making in agile software development context and (*R2*) provide integrated and meaningful data, considering the organization information needs and available data, aiming at enabling data-driven decision-making process.

In the *Design and development* step, we aim at developing the *Immigrant* approach. We started by performing three exploratory studies to give us useful knowledge to define the approach. The first exploratory study aimed to answer the following research question: *Is it useful and feasible to use ontologies to integrate existing data stored in applications to meet organization's data needs to decision-making in the agile context?* The first author worked as a Scrum master in a software development project in a public software organization that adopted agile practices. He observed that the development team had difficulty extracting and using decision-making data stored in the applications used to support the development process. Aiming to solve this problem, we developed a software architecture that allows integrating, sharing, and exchanging data from different applications and provides a dashboard (with integrated application data) to support decision-making. The architecture is based on the *Scrum Reference Ontology*, which provides a common conceptualization of Scrum (e.g., Scrum Project, Scrum Process, Product Backlog, Ceremonies, and Sprint) and is integrated into the *Software Engineering Ontology Network* (SEON) [Ruy et al. 2016]. With this study, we learned how networked ontologies can be used to integrate, share, and exchange data from different applications to support decision-making in the agile context. Details about the study can be found in [Santos Jr. et al. 2021a].

Considering that data integration must be guided by the organization's needs, we carried out other two exploratory studies aiming to answer the following research question: *How to identify the organization's data needs that are important to support decision making in agile software development context?* The study reported in [Santos Jr. et al. 2020] resulted in a system theory-based process that helps organizations to identify strategies to implement agile practices. The study reported in [Santos Jr et al. 2021b], in turn, resulted in *Zeppelin*, a diagnosis instrument that aids in understanding the current state of an organization by identifying the agile practices it performs. Together, the studies' results showed that by understanding the current state of an organization it is possible to identify aspects that need to be improved and, thus, create strategies to implement improvements aiming at agile development. The organization information needs can be derived from the strategies.

Considering the knowledge obtained from the exploratory studies, currently, we are working on developing the integrated solution of *Immigrant*. Figure 1 shows an overview of it. The approach considers, in a top-down perspective, information needs to be obtained from the current state of the organization and the actions to implement agile practices. The organization diagnosis and the definition of the actions to be performed to move from traditional to agile development are supported by the system theory-based process presented in [Santos Jr. et al. 2020] and the *Zeppelin* diagnosis instrument proposed in [Santos Jr. et al. 2021b]. These artifacts are meant to meet the requirements R1 cited above. From a bottom-up perspective, the approach considers data available in the applications used by the organization to support its development process. Considering the organization information needs and the available data, the approach uses networked ontologies (e.g., Scrum Reference Ontology [Santos Jr. a et al. 2021a] and Continuous Integration Reference Ontology) of SEON [Ruy et al. 2016] as an interlingua to integrate data from different applications and present integrated and meaningful data in dashboards to enable monitoring and insights aiming at data-driven decision-making. In this way, the approach meets the requirement R3.

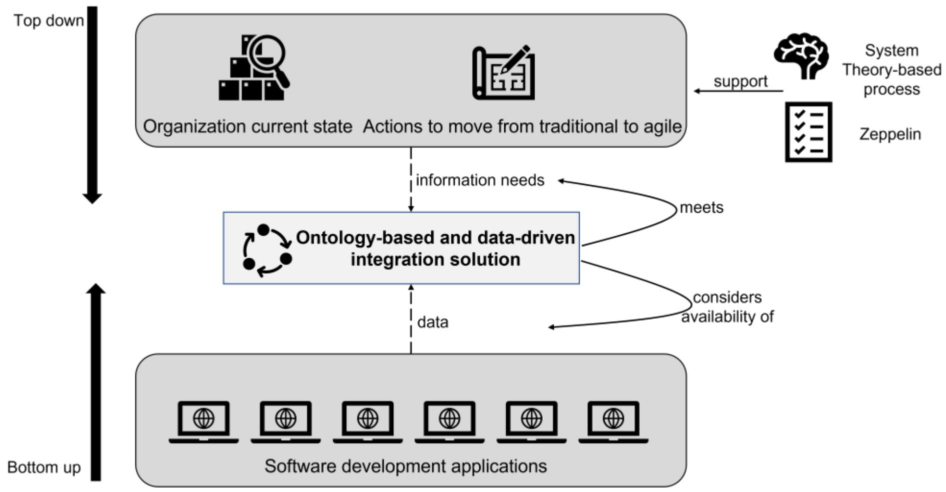


**Figure 1 – Overview of the proposed approach**

After developing *Immigrant,* in the *Demonstration* step, we will perform a proof of concept to show the feasibility of the proposed approach and, thus, in the *Validation* step, the approach will be applied in a case study in a software organization to evaluation and refinement. Finally, in the *Communication* step, we have presented the research results by publishing papers. New results will be published in future papers and the complete research project will be documented in the doctorate thesis.

## References

Berntsson Svensson, R., Feldt, R. and Torkar, R. (2019). The Unfulfilled Potential of Data-Driven Decision Making in Agile Software Development. In XP 2019. Springer.

Bosch, J. (2014). Continuous Software Engineering: An Introduction. Continuous Software Engineering. Cham: Springer International Publishing. p. 3–13.

Calhau, R. and Falbo, R. (2010). An Ontology-Based Approach for Semantic Integration. In 14th IEEE EDOC 2010.

Destefanis, G., Ortu, M., Counsell, S., et al. (2016). Software development: Do good manners matter? PeerJ Computer Science, v. 2016, n. 7.

Hevner, A. (2007). A Three Cycle View of Design Science Research. In SJIS, v. 19.

Izza, S. (2009). Integration of industrial information systems: From syntactic to semantic integration approaches. Enterprise Information Systems, v. 3, n. 1, p. 1–57.

Johanssen, J. O., Kleebaum, A., Paech, B. and Bruegge, B. (2019). Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners. In Journal of Software: Evolution and Process.

Karvonen, T., Lwakatare, L. E., Sauvola, T., et al. (2015). Hitting the Target: Practices for Moving Toward Innovation Experiment Systems. In ICSOB 2015.

Kasauli, R., Knauss, E., Nakatumba-Nabende, J. and Kanagwa, B. (2020). Agile Islands in a Waterfall Environment: Challenges and Strategies in Automotive. ACM ICPS.

Kleebaum, A., Johanssen, J. O., Paech, B., Alkadhi, R. and Bruegge, B. (2018). Decision knowledge triggers in continuous software engineering. In ICSE 2018.

Mattila, A.-L. L., Systä, K., Sievi-Korte, O., Leppänen, M. and Mikkonen, T. (2017). Discovering Software Process Deviations Using Visualizations. In LNBI.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007). A design science research methodology for information systems research. JMIS, v. 24.

Ruy, F. B., Falbo, R. de A., Barcellos, M. P., Costa, S. D. and Guizzardi, G. (2016). SEON: A software engineering ontology network. In LNCS.

Santos Jr., P. S., Barcellos, M. P. and Calhau, R. F. (2020). Am I going to Heaven? In Proceedings of the 34th Brazilian Symposium on Software Engineering. ACM.

Santos Jr., P. S., Barcellos, M. P., Falbo, R. de A. and Almeida, J. P. A. (2021a). From a Scrum Reference Ontology to the Integration of Applications for Data-Driven Software Development. Information and Software Technology, v. 136, p. 106570.

Santos Jr., P. S., Barcellos, M. P. and Ruy, F. B. (2021b). Tell me: Am I going to Heaven? A Diagnosis Instrument ofContinuous Software Engineering Practices Adoption. In EASE 2021.

Studer, R., Benjamins, V. R. and Fensel, D. (1998). Knowledge engineering: principles and methods. Data & Knowledge Engineering, v. 25, n. 1–2, p. 161–197.

Suárez-Figueroa, M. C., Gomez-Perez, A., Motta, E. and Gangemi, A. (2012). Introduction: Ontology Engineering in a Networked World. Ontology engineering in a networked world. Berlin, Heidelberg: Springer. p. 1–6.

Wegner, P. (1996). Interoperability. ACM (CSUR), v. 28, n. 1, p. 285–287.