sciendo

A Framework of Information Systems Development Concepts

Beatriz Meneses
University of Minho, MIEGSI, Portugal
João Varajão
University of Minho, Centro ALGORITMI, Portugal

Abstract

Background: Information Systems Development (ISD) is responsible for designing and implementing information systems that support organizational strategy, leveraging business models and processes. Several perspectives on this activity can be found in the literature, addressing – often in an undifferentiated manner – approaches, lifecycles, methodologies, and process models, among others. **Objectives:** The vast diversity of ideas and concepts surrounding ISD and the multiple underlying views on the subject make it harder for researchers and practitioners to understand the relevant aspects of this important activity. This article aims to systematize and organize ISD's main concepts to create a coherent perspective. **Methods/Approach:** We conducted a literature review and thematic analysis of ISD's main concepts. **Results:** To contribute to filling the research gap, this article proposes a new framework that addresses the key aspects related to ISD. **Conclusions:** The framework comprises ISD's core concepts, such as lifecycles, process models, deployment approaches, and methodologies.

Keywords: Information Systems Development; Digital Transformation; ISD; Information Systems; Lifecycle; Process Model; Approach; Deployment; Methodology; Method; Framework.

JEL classification: M15, O22
Paper type: Research article

Received: Aug 7, 2021
Accepted: May 8, 2022

Acknowledgements: This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

Citation: Meneses, B., Varajão, J. (2022), " A Framework of Information Systems Development Concepts", Business Systems Research, Vol. 13 No. 1, pp.84-103.
DOI: 10.2478/bsrj-2022-0006

Introduction

Information Technologies (IT) and Information Systems (IS) are fundamental for improving organizational performance (Bulchand-Gidumal et al., 2011, Pejić Bach et al., 2018). Organizations need to continuously evolve and adopt new, improved and modern ways of doing things (Ngereja et al., 2021) – IT has become essential to this end and inseparable from IS endeavours (Pearlson et al., 2016).

Several authors point out the impact resulting from the adoption of IT when organizations improve their IS. Such impact can be organized into four main categories (Alavi et al., 2015): efficiency improvement in business processes and transactions; communication improvement and centralized access to information, facilitating the decision-making process; modification of the basis of competition and the industry structure, leading to competitive advantages; and exploring new business models.

Given IT's wide diversity, it is not always easy for organizations to determine the most suitable technologies to adopt in a specific organizational context, nor how they can be operated (Dasgupta et al., 1999). This is usually done through implementing Information Systems Development (ISD) projects, often called digital transformation projects (Kääriäinen et al., 2020). The underlying objectives of these projects are, for example, to improve business models, products, services, processes, communication channels (Haffke et al., 2016), specific practices (e.g., fraud detection (Pejić Bach et al., 2020)), or the relationship with clients or suppliers (Bharadwaj et al., 2013).

Several perspectives and concepts can be found in the scientific literature and the practitioners' lexicon regarding ISD, such as lifecycles, process models, deployment approaches, methodologies, methods, etc. However, the inexistence of a shared understanding of the concepts results in messy vocabulary use and a sort of conceptual chaos. This article aims to contribute to solving this issue by proposing a framework to address and organize the main ISD concepts. The main contribution is both theoretical and practical. On the one hand, the framework provides an organized perspective on the relevant concepts of ISD; on the other hand, it can be used by practitioners to raise their awareness of the different alternatives to be followed in their projects – for instance, regarding deployment approaches.

The document is structured as follows: section 2 presents the background; section 3 describes the research framework; section 4 addresses the research method; section 5 presents the results, and section 6 discusses the results; finally, section 7 presents the conclusions, limitations, and proposals for further work.

Background

An IS is "a combination of intelligent agents (human and/or artificial), processes, and IT (hardware, software, and infrastructure) related to the dissemination and use of data, information, and knowledge in an organization" (Varajão et al., 2021). Accordingly, an IS project can be defined as "a temporary endeavour undertaken to improve organizational IS, and can take on many forms" (Varajão et al., 2020), from the development of a software artefact to the implantation of a commercial-off-the-shelf application (Varajão et al., 2018). As a result, the term ISD can also be defined from different points of view (Hirschheim et al., 1996).

For Laudon et al. (2007), ISD is characterized by the activities involved in creating an IS, and its origin can be traced back to organizational problems or opportunities. Carvalho (1996) stated that ISD processes are triggered when organizations become aware of the necessity to improve their IS, which results from the continuous monitoring of their performance. According to Varajão (2002), ISD interventions emerge from the necessity of achieving the change devised (or planned) at the time of IS planning.

Authors such as Hirschheim et al. (1996) mention that ISD results from combining a major influx of activities, specifically, IS analysis, design, construction, and deployment. On the other hand, Welke (1983) defines ISD as a change process regarding a system of objects whose purpose is to meet the proposed goals and improve IS performance.

Hirschheim et al. (1996) view ISD from a social action theory perspective and define ISD as "the purposeful crafting and construction of artefacts", including "hardware configurations, design and analysis documents, code, user documentation, organizational structures and procedures, etc.". The same authors mention that technology, organization, and language are the main points of change in ISD.

Different proposals for systems development (Laudon et al., 2007) vary according to the type and dimension of the system being developed. Carvalho (1996) proposes several scenarios based on the different IS interpretations, which illustrate the "path" of the development process based on two dimensions: (i) phases of the development process (perception, conception, and implementation); (ii) object of intervention (organization, information system, and computer system). He also states that even though any scenario can be associated with the term information systems development, only one of them can make better use of that designation. In such a scenario, ISD is conceived as an organizational intervention to improve IS (Carvalho, 1996).

ISD's inherent complexity can be easily overlooked (Varajão, 2002). Since organizations increasingly depend on IS to perform their activity and evolve, continuous efforts are required (Pereira et al., 2022). This, in turn, introduces more complexity into the process (Morcov et al., 2020), which is then reflected in projects (Xia et al., 2005), requiring a comprehensive understanding of all ISD-relevant aspects.

Research Framework

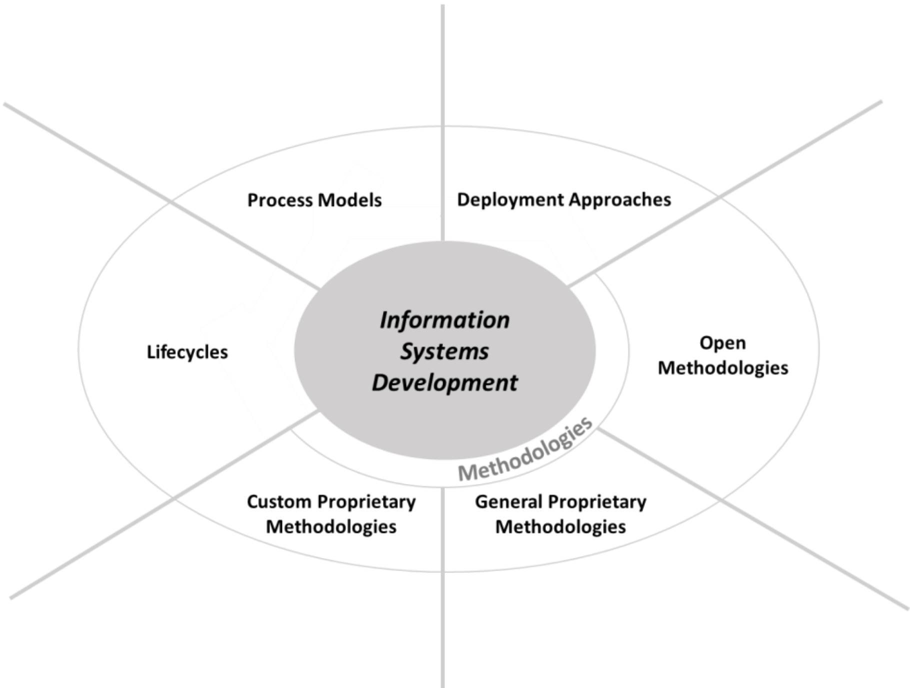
As a complex activity, ISD can be approached from multiple perspectives. On the one hand, it is possible to recognize several lifecycles in an ISD intervention, which are related to the project as a whole (Wong et al., 2018), its execution (PMI, 2017), and also to the products or services resulting from it (Varajão, 2018b, Varajão et al., 2022b).

On the other hand, given the distinct nature of each intervention, it is necessary to adopt different process models closely related to the development lifecycles (Avison et al., 2006b, Ozturk, 2013). Since there are several process models for ISD (Singh et al., 2019), to foster the project's success, it is necessary to analyze which is the most appropriate for a given context (Boehm, 1988), taking into account not only the nature of the project, its application, the methods and tools to be used, but also the specific conjuncture, type of control and expected results (Pressman, 1997). IT adoption and implementation is one of the most important activities of the ISD process. In this case, ISD is perceived as improving an organization by adopting IT (Spohrer, 2016, Varajão et al., 2022a). As with process models, it is important to assess and select an adequate deployment approach by analyzing each option's associated costs, risks, and benefits (O'Leary, 2000).

Another important aspect of an ISD intervention is the methodology used to support such activity, whose primary purpose is not limited to providing a solution to some of the difficulties inherent to the development process (by systematizing and organizing it) (De Leoz, 2017), but also to deal with the complexity of ISD projects (Avison et al., 1999). Methodologies can be grouped into three categories: open methodologies (e.g., SSADM (Ashworth, 1988)); general proprietary methodologies, which are developed by major organizations in the IS arena and made available to their partners (e.g., SAP Activate (SAP, 2017)); and custom implementation methodologies created only for internal use of organizations (e.g., companies such as Accenture have their proprietary methodologies).

All of the abovementioned aspects comprise the research framework depicted in Figure 1, which will be further detailed in the ensuing sections.

Figure 1
Research Framework



Source: Author's illustration

Research Method

This section describes the literature review that was carried out to support the concepts identified in the research framework.

Regarding data sources, the research focused on two of the most recognizable platforms in the academic and scientific context: Scopus and Web of Science. It is important to note that the results were often redirected to other data sources, although these two were the most used sources. Therefore, the following platforms were also used: AIS eLibrary, ScienceDirect, Research Gate, SpringerLink, IEEE Electronic Library, and Google Scholar.

Before conducting the research, it was necessary to define the key concepts, and so the following terms were used, considering the research framework:

- "information system* development", "ISD";
- "life cycle*", "lifecycle*";
- "process model*";
- "implementation strateg*", "deployment strateg*", "implementation of enterprise system*";
- "method*".

The search queries were formulated through logic statements defined based on the previously mentioned terms. Some restrictions were imposed: (1) regarding source type – only journals and conference proceedings were considered; (2) regarding the area of study – the selected areas were computer science, engineering, social sciences, business management and accountancy, and decision science (as well as similar areas, depending on the search engine that was used). It should be noted that no restrictions were defined regarding the period.

In the first step, only one research expression was used, including all of the terms mentioned. However, a preliminary analysis of the results obtained first verified that they were ambiguous and did not explore the key concepts in detail. Therefore a phased search was subsequently conducted. Given the large number of articles obtained from applying some research terms, it was decided that, in such cases, only the titles would be analyzed, rather than the combination of title, abstract, and keywords (often, the title is enough to assess whether the article fits the purpose of the analysis or not (Kraus et al., 2020)). The obtained results are presented in Table 1.

Table 1
Results from search

	Scopus	Web of Science	Selected articles
	Results	Results	
(TITLE (“information system* development” OR “ISD”))	1,407	945	22
(TITLE ("information system*" OR "information technolog*") AND TITLE-ABS-KEY ("implementation strateg*" OR "deployment strateg*" OR "implementation of enterprise system*"))	114	39	20
(TITLE (“information system*” OR “information technolog*”) AND TITLE (“process model*”))	66	32	21
(TITLE (“information system*” OR “information technolog*”) AND TITLE (“life cycle” OR “lifecycle”))	90	52	28
(TITLE (“information system*” OR “information system* development”) AND TITLE (“method*”))	1,771	685	35

Source: Author’s work

The resulting literature list was compiled in an Excel file. Preliminary filtering was carried out to eliminate repeated articles. For selection purposes, the articles were evaluated and filtered in multiple stages to assess their relevance. The first stage was conducted based on the articles’ titles (in such a way that any title failing to match the scope of the research would be automatically excluded, and the more doubtful cases would move on to the next stage). The second evaluation consisted of reading the articles’ abstracts, and if any of them failed to mention the keywords related to the study, they would be similarly excluded. Nevertheless, whenever the information contained in the abstract was considered insufficient to analyze the article's relevance, a full reading was required, with particular emphasis on the introduction and conclusion. At the third and final evaluation stage, the articles were fully read, resulting in the total number of selected articles identified in the last column of Table 1, which includes all the articles that address the topic in a more detailed analysis.
It is important to note that, during the detailed analysis of the articles, it was found that many of them included references to other articles. So the cases of repeatedly quoted references, or references considered relevant, were added to the list of articles for further reading.

Results

ISD Lifecycles

Lifecycle models supply an orientation basis when developing and evaluating complex systems (McConnell, 1996, Hoffer et al., 2007). Therefore, it is relevant to approach three different but intimately connected lifecycles (Varajão, 2018b, Varajão et al., 2022b): the system development lifecycle (SLDC), the project's lifecycle, and the product's lifecycle.

System Development Lifecycle

The System Development Lifecycle (SDLC) plays a crucial role In the IS area (Avison et al., 2006a) because, as the name suggests, it is a powerful basis for IS development (Hedman et al., 2009, Oz, 2009), and it provides a set of necessary guidelines for IS implementation.

Although there are other distinct classifications and possibilities when it comes to structuring the phases, simply put, the lifecycle of the traditional systems development implies the existence of five phases – planning and problem identification, analysis, design, development, and, lastly, operation and maintenance (Hedman et al., 2009). Laudon et al. (2007) use a metaphor to highlight that, as with any human or living organism, a system's lifecycle can be broken down into three distinct moments: a beginning, a middle, and an end. Using a different approach than previously mentioned, these authors organized the system development lifecycle into six phases: project definition, system study, project, programming, installation, and post-implementation. Another example of the system development lifecycle comes from Avison and Fitzgerald (2006a), who divided it into the feasibility study, system investigation, system analysis, system conception, implementation, revision, and maintenance.

Avison et al. (2006a) pointed out some inherent benefits of using SDLC. They highlight its simplicity and ease of understanding and the existence of a methodological basis with specific documentation, deliveries, tools, and guidelines that support each phase. The typical progression in a lifecycle model is a linear sequence that follows a particular order in which every phase is related to the other. The outputs from one phase are used as inputs for the following phase (Van de Ven et al., 1995). Another lifecycle characteristic was noted by Avison et al. (2006a), which focused on the formal task division between the different specialists on a business and technical level. The traditional approach to the system development lifecycle has been gradually replaced by alternatives that also boost IS development, aiming at dealing with the limitations of the classic lifecycle by organizing activities in a waterfall format. For Laudon et al. (2007), the waterfall lifecycle is rigid and inflexible when reviewing requirements and specifications.

Similarly, Griffin et al. (2010) mentioned that a change made in one of the phases might result in modifications in the other phases as well, given their sequential nature, which assumes that one phase must be finished so that the next one can proceed. Also, Avison et al. (2006a) pointed out some flaws in responding to management needs and the excessive emphasis on the technical component, which tends to cause client dissatisfaction. Other aspects, such as time or financial constraints, can occasionally impact determine a different approach for systems development (Oz, 2009).

Project Lifecycle

The system development lifecycle (without the post-implementation operation/maintenance phases) is typically integrated into another cycle, the project's lifecycle. The project's lifecycle comprises the phases that describe a project's lifespan, from start to finish (PMI, 2017, 2021).

Authors such as Pinto et al. (1988) and Thamhain et al. (1975) stated that a project's lifecycle is crucial in determining its successful implementation. Phases can be sequential, iterative, or overlapped, and the designation, number, and duration of each phase are directly related to the organization's need for management and control, as well as the project's nature and application field (PMI, 2017). Although differences may occur according to the business sector and the project itself, particularly in terms of dimension and complexity, in PMI (2021)'s point of view, four phases describe a project's life cycle: project start, organization and preparation, work execution and, finally, project closure. Monitoring and control are required along these four phases.

It should be noted that it is generally during the first phase that the system development lifecycle is defined. Consequently, SLDC can be considered an integral part of a project's lifecycle, as it fits in the work execution phase.

Product Lifecycle

Since the expected result of an ISD project is the introduction of one or more IT artefacts in the organization that lead to modifications (outcomes), it is necessary to consider another lifecycle, the product's lifecycle (PLC). The product lifecycle perspective is commonly related to its market introduction and evolution in the literature. According to Buzzell (1966), the PLC represents the unit sales line of a product, depicting the evolution of the market's attributes and characteristics (Polli, 1968) from the moment it is introduced in the market until its removal. Authors such as Levitt (1965) considered that a product should go through certain phases to be successful. Even though the literature presents different considerations regarding these phases, a well-accepted example is suggested by Levitt (1965), who referred to them as development, growth, maturity, and market downturn, thus stressing the importance for organizations to outline strategies compatible with each phase (Dean, 1950, Clifford, 1965). The product's lifecycle can support production planning and control (Forrester, 1958, Cox Jr, 1967, Cao et al., 2011).

Within the scope of this work, it is pertinent to analyze the perspective of adopting and introducing a product into an organization, which follows the same reasoning as a product introduced in the market (Varajão, 2018b). In this case, an IT product is developed and adopted by an organization to respond to previously identified business problems or opportunities. Maintenance activities should be untaken to ensure permanent alignment with business needs and product suitability for as long as possible. However, given that the changes in the internal and external business environment happen all the time, the organization may have to adopt a different solution in the future so it can evolve, which could mean replacing the product, thus resulting in its decline and removal. Consequently, a new IT product will have to be created, and the ISD process will be repeated.

ISD Process Models

Related to the system's lifecycles, there are process models. To better understand the term "process model", it is important to clarify the different interpretations of this concept.

According to Van de Ven (1992), a process can be seen from three perspectives: (i) a series of events that describe evolution through time; (ii) a category of concepts or variables related to actions undertaken by individuals or organizations; (iii) a logic that explains a causal link between variables, whether these are dependent or

independent. In this way, process models are projected to create sequences of events or stages to obtain a given result (Mohr, 1982), making clear how and why a process evolves in a specific way to achieve certain results (Mohr, 1982, Newman et al., 1992, Van de Ven et al., 1995, Langley, 1999, Cule et al., 2004). Process models are commonly associated with a particular type of ISD, which involves software development (creation). A software process model consists of a series of activities needed to develop a software product. Pressman (1997) defended that the process model selection should be based on the project's nature, the type of methods and tools to be used, and the need to make frequent deliveries and controls. Process models are closely connected to lifecycles and can also be used to ease and/or restrain the deployment approach.
The main models are described hereafter: Waterfall Model, Prototyping Model, Spiral Model, RAD Model, V-Shaped Model, Incremental Model, and Agile Models.

Waterfall Model (1970)
The waterfall model, also known as the classic lifecycle model (Pressman, 1997), is one of the most widespread models. As this model is sequential, it is impossible to proceed to the following phase if the previous one is not finished. According to Royce (1970), this model comprises the following phases: requirements definition, system design, unit implementation and testing, and operation. At the end of each phase, the project will be reviewed to ensure it is evolving as intended.

Prototyping Model (1970)
Clients frequently define a set of overall goals for a software project while not fully specifying the set of requirements to be checked, thus hindering the work of the development team (Pressman, 1997). The prototyping model is suitable for dealing with this kind of situation, as it begins with the preliminary gathering of requirements together with the client. Based on these requirements, an initial draft of the solution is then created, featuring only the representation of the visible aspects of the software for the client (Pressman, 1997), which will subsequently lead to the prototype construction. The prototype is cyclically used and evaluated by the client, so the requirements can be built and perfected until the final product is achieved.

Spiral Model (1988)
The main feature of the spiral model, originally proposed by Boehm, sets it apart from other models. It includes the notion of risk, which solves many existing difficulties (Boehm, 1988). The spiral shape, so typical of this model, represents the phases that comprise it, and risk assessment should be made in each one. Every "lap" of the spiral is divided into four sections: goals definition; risks identification, evaluation, and respective mitigation; development and validation; and planning of the upcoming iteration. A software project will go through each phase sequentially and repeatedly, and each resulting spiral is based on the baseline spiral. The spiral model is divided into activities, including analysis, design, implementation, testing, and deployment.

Rapid Application Development (1991)
Rapid Application Development (RAD) is a model proposed by James Martin based on rapid prototyping approaches. This incremental model prioritizes short, rapid, iterative, and low-cost development cycles and quality enhancement and the enrollment of the development team and the clients throughout the entire process. It should be noted that a prototype that is being created can undergo changes, and therefore any modifications regarding the requirements can be easily incorporated into the final solution. The RAD model also covers the following phases: business, data

and process modelling, application generation, and testing and re-use (Pressman, 1997).

V-Shaped Model (1991)
The V-Shaped model comprises two major moments: the decomposition and definition moment and the integration and verification moment. The model starts by answering the user's requirements and finishes with a system that the user properly validates. As with the waterfall model, the V-Shaped model also aims to implement each phase sequentially so that the previous phase must be completed for the next. More specifically, one side of the V-model, which comprises the development life cycle phases (requirements definition, analysis, design, and coding), goes down as the waterfall model. In contrast, the other side, where the testing phases are performed (unit test, integration test, system test, acceptance test) flows upwards, as a successive progression takes place regarding assemblies, units, and subsystems, with the respective checking, ending at the system level (Forsberg et al., 1992). These same authors claimed that the respective testing stage could be conducted in parallel and in a corresponding way for every phase of the development cycle.

Incremental Model (n.d.)
The incremental model combines elements from the linear sequential and iterative prototyping models (Pressman, 1997). Unlike the waterfall model, where the development takes place all at once, in this model, software increments are produced at each linear sequence. As a first increment, some emphasis is given to the main product (Pressman, 1997) since the goal is to attend to the necessities and requirements to ensure operations continuity. Each increment goes through the requirements, project, implementation, and testing phases. Before moving on to the following increment, a plan is developed to deal with the multiple modifications to the main product (Pressman, 1997). As each increment is finished, an operational product is delivered to the client. This process is repeated until a new product is completely produced.

Agile Models (2001)
Because previously detailed models are usually considered rigid, agile models emerged to make software development more efficient and effective. Generally speaking, the agile models are characterized by the following attributes (Abrahamsson et al., 2002): incremental (by creating "small" versions of the target product(s) with fast development cycles), cooperative (with constant communication between the client and the development team), simple (the created models are easy to learn and modify), and adaptable (there is the capacity to adapt to unpredictability and requirements modification). Agile models have become accepted as a way for organizations to create new products (Durbin et al., 2021).

ISD Deployment Approaches
The ISD activity, as a "project", typically ends with implementing all the modifications designed for the organization, including IT implantation. As this is a critical activity for achieving success and deeply impacts the organization, the organization must choose the right deployment approach, considering the new IS solution coverage and suitability. In addition, according to O'Leary (2000), defining a deployment approach should not only be based on cost and risk analysis but also the benefits stemming from each option. Regarding the organization of ISD deployment activities, the big-bang and the phased approaches are the most commonly used (Robinson,

2010). Besides the mentioned approaches, others are also used, such as the parallel and pilot approaches.

Big-Bang Approach

As the name implies, during the big-bang approach, the deployment is straightforward. It happens all at once, so every application and project modification is implemented simultaneously in the entire organization and during the same period. This approach implies higher risk because the legacy system is immediately disabled and makes room for a new system. The failure of only one component may jeopardize the entire system. Deployment time can be shorter when choosing the big bang approach, so the time required for employees to become acquainted with the new system also tends to be shorter. From the employees' perspective can be beneficial and advantageous, as it is not necessary to use two different systems simultaneously for the work to be implemented (O'Leary, 2000).

Phased Approach

Implementing the phased approach, also known as the incremental approach, implies the existence of several stages for a specific period. It can be operationalized in three ways (O'Leary, 2000): (1) IT modules or module grouping, in which greater emphasis is placed on the IT main modules to be implemented first, and in a limited number of organizational units, following the implementation of the remaining modules; (2) business unit, where the implementation is made in one or more business units at a time; (3) geographic area, which occurs in organizations distributed by several locations. Compared to the big-bang approach, the phased approach presents a lower risk in case of implementation failures, as the organization can resort to the legacy system until the problems are solved (O'Leary, 2000), even though this tends to incur more costs. According to Robinson (2010), this approach is more complex due to the common existence of dependencies between the different modules and business units. The phased approach, characterized as being slower, allows the introduction of significant and continuous improvements between phases and better management. Because this approach takes longer, an organisation's employees can acquire the necessary skills and expertise to use the new system. However, this same reason can result in a disadvantage, as the top management's and team's engagement may diminish over time, even before the project is fully completed (O'Leary, 2000).

Parallel Approach

The parallel approach implies the simultaneous execution of the legacy and new systems, resulting in a safer implementation. This way, users can learn to operate the new system while performing their daily tasks on the legacy system. When the new system is properly implemented and at the most appropriate time, the integral transition to the new system takes place (Leon, 2009), and the previous one is abandoned. Since this is an intermediary approach to the deployment mentioned above, the total failure risk is reduced. However, the parallel execution of two systems can be expensive, require more resources, and may reduce the new system's potential due to scattered attention (O'Leary, 2000).

Pilot Approach

In the pilot approach, firstly, only part of the organization uses the new system; later, to implement the other parts, it is necessary to use either the big-bang approach or the parallel approach (O'Brien et al., 2010). The major advantage associated with this type of implementation is to find and solve potential problems at the pilot site without having a widespread impact on the organization (Xu, 2019). This way, the use of a new system will only be extended to other sites when the pilot test is properly evaluated and successfully concluded. One disadvantage of this approach is that each site's singularities may differ from the pilot site's (O'Brien et al., 2010).

Hybrid Approach

Additionally, several authors, such as Leon (2009), consider another deployment approach: the hybrid approach. This approach combines different types of implementations, which, for instance, allows organizations to use a big-bang approach for smaller business units, and the phased approach for the others. Therefore, as two cycles are required to conduct the deployment, fewer resources are needed compared to the pure big-bang approach, and the deployment time is also reduced. However, the possibility of returning to a previous phase may affect the associated deployment costs (Elizabeth et al., 2015).

Agile Approach

Another approach for ISD deployment that has not been explored in-depth in the research literature outside the software development field is the agile approach. According to Moran (2015), agile allows obtaining independent, simple, and partially functional increments, forming the full product together. The project's product is gradually enhanced by implementing successive iterations of a typically fixed duration. With each iteration, the development team not only benefits from the lessons learned with the previous iteration and reflects on the modifications required by the client but also integrates new functionalities. Therefore, a new usable product increment is created and delivered for the client to use it and provide subsequent feedback (Abbas et al., 2008). These same authors also highlight a major advantage of adopting this approach – the ability to adapt to the uncertainty of the development environment.

ISD Methodologies

According to Avison et al. (1998), ISD methodologies consist of crucial structures that can be used during the definition and development process. Conventional approaches tend to overly emphasize the technical part, which is insufficient to support the entire ISD process. Therefore, it is important to encourage the balance between technical rationality and the social aspects that are part of the ISD process. Methodologies typically have particular life cycles (Mittermeir, 1992), process models (Verhage, 2009), and deployment approaches.

Open Methodologies

Overall, methodologies are designed to improve existing approaches or propose new forms of ISD (Avison et al., 2006b). A set of public domain methodologies can be found in the scientific literature, so there is no restriction on their access and use. For this reason, they are referred to here as open methodologies. Some of the main ISD open methodologies are presented next: Information Engineering, Soft Systems Methodology, MERISE, SSADM, Multiview, and Yourdon Systems Method.

Information Engineering (IE) (1981)

The Information Engineering methodology identifies a set of activities and techniques concerning ISD. It comprises several phases (Richmond, 1991): information strategy planning, business area analysis, system analysis and representation, construction, transition, and production.

Soft Systems Methodology (SSM) (1981)
The Soft Systems Methodology is a systemic methodology that optimizes continuous learning about a complex real-world situation, providing a structure that overcomes the lack of a consistent definition of the problem (Checkland, 1989).

MERISE (1983)
The MERISE relies on three cycles: the decision cycle, which is related to the multiple decision mechanisms; the lifecycle, which reflects the chronological IS progress; and the abstraction cycle, which is the focal point of this methodology that portrays the different processes and data models through several stages (conceptual, logical, and physical) (Avison, 1991), contrary to what happens in other methodologies, which were not designed to equally emphasize each phase.

SSADM (1988)
Considered the "modern" version of the traditional approach to the ISD lifecycle, SSADM divides a project into structured tasks that fit into a sequence of seven phases. This methodology combines three principles bearing the same degree of relevance: method structure, structured techniques and their respective correlation, and documents and forms created (Ashworth, 1988). Each of these principles provides a different vision of the same system, contributing to the final projection of the system's model.

Multiview (1990)
Multiview is seen as a hybrid process involving the development team and the final users of the system (Avison et al., 1998), thus ensuring that a special focus is given to the social and technical side throughout the whole ISD process. It should be noted that this method includes five development phases.

Yourdon Systems Method (YSM) (1993)
This method consists in breaking down a problem into functional units. This method comprises three main phases (Avison et al., 2006a): feasibility study, construction of the main model, and deployment model.

Additionally, we can mention the Unified Process (UP) (Jacobson et al., 1999), which is focused on software development. Moreover, three additional proposals were found – engIS (Carvalho, 2017), agilIS (Varajão, 2018a), and AgileMIP (Soares, 2020) – but they are currently restricted to teaching activities. Overall, based on the literature review, an apparent stagnation of the proposal and evolution of ISD open methodologies stand out.

8.2. ISD General Proprietary Methodologies
Some of the major organizations in the IS arena develop their methodologies to be used by themselves and their business partners. For this reason, they are called here *general proprietary methodologies*. These methodologies supply the tools and guidance for ISD project management, including the phases of the IT business solutions implantation process. Just three examples are mentioned next to avoid presenting an extensive list: Oracle Unified Method, Microsoft Dynamics Sure Step Methodology, and SAP Activate.

OUM (2006)

The Oracle Unified Method is a methodology proposed by Oracle based on the Unified Process. According to Oracle (2016), this methodology provides organizations with a quick, flexible, business-oriented approach that is also easily adaptable to a wide range of specific organizational situations. OUM also includes an operational structure that allows anticipating a project's needs and critical points. This methodology covers the entire deployment process by organizing it into six phases – initiation, conception, construction, transition, and production – and includes the optimization and updating phases.

Microsoft Dynamics Sure Step Methodology (2007)

Microsoft Dynamics Sure Step Methodology integrates project management tools and recommended practices to assist the implementation of Microsoft business solutions. It comprises six main phases (diagnosis, analysis, design, development, deployment, and operation) and two additional phases (optimization and updating). Sure Step comprises eight processes (business processes analysis, configuration, data migration, infrastructure, installation, integration, testing, and training) carried out during the different phases (Microsoft, 2014, 2021).

SAP Activate (2015)

SAP Activate is a methodology featuring a modular and agile structure developed by SAP, designed to implement SAP S/4HANA and other SAP solutions. SAP Activate provides a wide set of specific and structured tools, practices, models, and processes tailored to the specific business needs of the organizations. This methodology comprises six phases: discovering, preparing, exploring, performing, implementing, and executing. Systematically, there are quality checkpoints throughout the different phases, which are decisive for the project's success (SAP, 2017).

ISD Custom Proprietary Methodologies

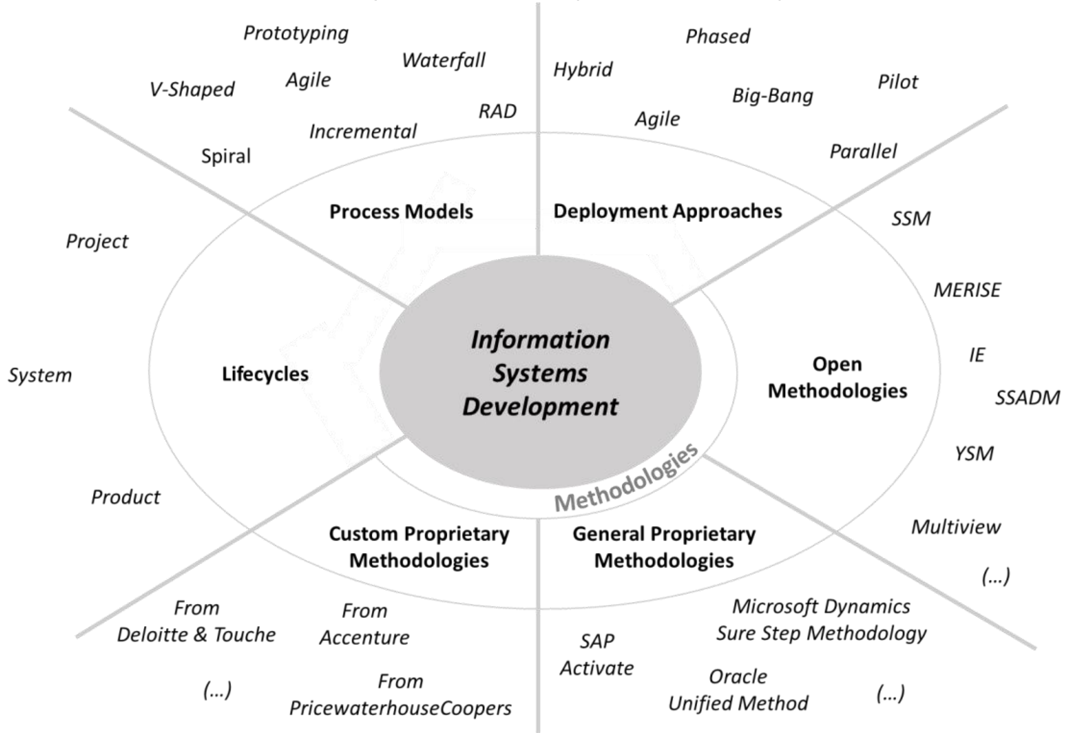
Some of the major organizations with a strong component in the IS field developed their implementation methodologies. Contrary to the general proprietary methodologies addressed in the previous section, these methodologies are restricted to the owner organizations. Only those organizations can use them internally, so they are designated as *custom proprietary methodologies*. As with the general proprietary methodologies, custom proprietary methodologies can also provide the necessary tools and guidelines for project management and, more precisely, for all the phases of the business solutions development process. Deloitte & Touche, Accenture, and PricewaterhouseCoopers are examples of organizations that have developed their methodologies, which are not in the public domain.

ISD Framework and Discussion

Framework

Figure 2 summarizes the main results by presenting a framework of ISD concepts.

Figure 2
The framework of Information Systems Development Concepts



Source: Author's illustration

Considering all the concepts, ISD endeavours can be identified as three lifecycles: project, system development, and product. The project lifecycle comprises the initiation, planning, execution, and closing phases. In the initiation and planning phases, it is important to consider the process model (e.g., Waterfall) and the deployment approach (e.g., Big-Bang) to be adopted since they influence how project activities are organized. The system development lifecycle globally regards the execution activities of the project lifecycle, which can be organized according to different process models (e.g., Waterfall). One important set of process model activities is the deployment of project outputs, which can also be conducted following different approaches (e.g., Big-Bang). When the outputs are made available to the organization, then the product lifecycle starts. The methodologies may comprise all the lifecycles and typically have an implicit process model and preferred deployment approach, whether open, general proprietary, or custom proprietary.

Contributions to Theory

As described in the previous sections, there are different aspects and related concepts that should be considered regarding ISD, which are so deeply intertwined that they are frequently addressed in an undifferentiated manner and create confusion and give rise to misconceptions. The framework presented here contributes to the ISD body of knowledge, aiming to provide an integrated perspective on ISD concepts by identifying, describing, and relating four major conceptual dimensions: lifecycles, process models, deployment approaches, and methodologies.

Implications for Practice

Many aspects must be considered in an ISD project. For instance, it is important to recognize the different lifecycles to define success evaluation milestones, addressing the project management success and the business impact (Varajão, 2018b, Varajão et al., 2022b). The process model and the deployment approach should also be defined by considering the project's characteristics and context (e.g., the need for frequent deliveries). Furthermore, it is important to adopt suitable methodologies because they provide detailed guidelines on the activities to be carried out and the artefacts to be produced. Our framework provides practitioners with guidance on these aspects and promotes awareness of the different alternatives to be followed in projects.

Also, of note is the relevance of the proposed framework for teaching since it provides a structured perspective on the relevant concepts and is described progressively (from lifecycles to methodologies).

Conclusion

On the one hand, the wide range of ideas and concepts surrounding ISD and the multiple underlying views on the subject make it harder for researchers and practitioners to understand the relevant aspects of this important activity. On the other hand, extant research is typically focused on different aspects of ISD (e.g., a particular process model). The same expressions are often used to refer to different ISD aspects, or different expressions refer to the same ISD aspects. The ISD framework presented here systematizes and organizes the main concepts of ISD to create a coherent perspective regarding lifecycles, process models, deployment approaches, and methodologies.

Limitations

As a major limitation of this study, we identify the superficial exploration of ISD's proprietary methodologies. This is partially due to the article's length restrictions and the restricted access to those methodologies since they are not in the public domain. Another limitation is related to the methodology's coverage since the main goal here was to classify them and not to provide a complete list of extant methodologies. These limitations open avenues for further research.

Future Research Directions

Stagnation is noted regarding ISD open methodologies, at least apparently, since we could not find recent proposals in the research literature on this matter. On the one hand, software development has been fertile in recent years. It has shown high dynamism by proposing several frameworks, process models, and methodologies, mainly related to the agile approach (e.g., Scrum, XP, Crystal, etc.). On the other hand, in the case of ISD, it seems that this kind of effort has been limited to private companies. The agile approach emerged to fill some of the existing gaps in classical models and has been widely adopted in software development. However, to the best of our knowledge, there is no research literature focused on agile ISD processes or methodologies, and this can be an opportunity for future work. It would be an interesting avenue to develop a comparative study on proprietary methodologies, aiming at identifying similarities, differences, improvement opportunities, and good practices.

A final message is addressed to companies to disclose their proprietary methodologies to the public domain. Making the methodologies accessible to the public domain would contribute to research and education in this field, and therefore their disclosure (even if partial) is strongly encouraged.

References

1. Abbas, N., Gravell, A. M., Wills, G. B. (2008), „Historical Roots of Agile Methods: Where Did "Agile Thinking" Come From?", in Abrahamsson, P. et al. (Eds.), Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing. Berlin, Heidelberg, Springer, pp. 94-103.
2. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. (2002), „Agile Software Development Methods: Review and Analysis", Espoo, Finland, VTT Technical Research Centre of Finland, VTT Publications.
3. Alavi, M., Yoo, Y. (2015), „Use Information Technology for Organizational Change", in Locke, E. A. (Ed.), Handbook of Principles of Organizational Behavior, Chichester, Wiley, pp. 595-614.
4. Ashworth, C. M. (1988), „Structured systems analysis and design method (SSADM)", Information and Software Technology, Vol. 30 No. 3, pp. 153-163.
5. Avison, D. E. (1991), „MERISE: A European methodology for developing information systems", European Journal of Information Systems, Vol. 1 No. 3, pp. 183-191.
6. Avison, D. E., Fitzgerald, G. (1999), „Information Systems Development", in Currie, W., Galliers, B. (Eds.), Rethinking Management Information Systems: An Interdisciplinary Perspective, Oxford, Oxford University Press, pp. 136-155.
7. Avison, D. E., Fitzgerald, G. (2006a), Information Systems Development: Methodologies,Techniques, and Tools, 4th ed., New York, McGraw-Hill Education.
8. Avison, D. E., Fitzgerald, G. (2006b), „Methodologies for Developing Information Systems: A Historical Perspective", in Avison, D., Elliot, S., Krogstie, J., Pries-Heje, J. (Eds.), The Past and Future of Information Systems: 1976–2006 and Beyond, IFIP WCC TC8 2006, IFIP International Federation for Information Processing, Boston, MA, Springer.
9. Avison, D. E., Wood-Harper, A. T., Vidgen, R. T., Wood, J. R. G. (1998), „A further exploration into information systems development: the evolution of Multiview2", Information Technology & People, Vol. 11 No. 2, pp. 124-139.
10. Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., Venkatraman, N. V. (2013), „Digital business strategy: Toward a next generation of insights", MIS Quarterly, Vol. 37 No. 2, pp. 471-482.
11. Boehm, B. W. (1988), „A spiral model of software development and enhancement", Computer, Vol. 21 No. 5, pp. 61-72.
12. Bulchand-Gidumal, J. Melián-González, S. (2011), „Maximizing the positive influence of IT for improving organizational performance", The Journal of Strategic Information Systems, Vol. 20 No. 4, pp. 461-478.
13. Buzzell, R. D. (1966), „Competitive behavior and product life cycles", in Wright, J., Goldstucker, J. (Eds.), New Ideas for Successful Marketing, Chicago, American Marketing Association, pp. 46-67.
14. Cao, H., Folan, P. (2011), „Product life cycle: the evolution of a paradigm and literature review from 1950–2009", Production Planning & Control, Vol. 23 No. 8, pp. 641-662.
15. Carvalho, J. Á. (1996), „Information Systems Development: From Computer Systems Construction to Organizational Reengineering", Guimarães, Portugal, Department of Information Systems, University of Minho.
16. Carvalho, J. Á. (2017), „engIS (v3)", Guimarães, Portugal, Department of Information Systems, University of Minho.
17. Checkland, P. B. (1989), „Soft Systems Methodology", Human Systems Management, Vol. 8 No. 4, pp. 273-289.
18. Clifford, D. K. (1965), „Leverage in the Product Life Cycle", Duns Review of Modern Industry, Vol. 85 No. 5, pp. 62-70.
19. Cox Jr, W. E. (1967), „Product life cycles as marketing models", Journal of Business, Vol. 40 No. 4, pp. 375-384.
20. Cule, P.E., Robey, D. (2004), „A dual-motor, constructive process model of organizational transition", Organization Studies, Vol. 25 No. 2, pp. 229-260.
21. Dasgupta, S., Agarwal, D., Ioannidis, A., Gopalakrishnan, S. (1999), „Determinants of information technology adoption: an extension of existing models to firms in a developing country", Journal of Global Information Management, Vol. 7 No. 3, pp. 30-40.

22. De Leoz, G. M. (2017), Exploring the impacts of development methodologies on the collective action of information systems development project teams, Doctoral dissertation, Baylor University.
23. Dean, J. (1950), „Pricing Policies for New Products“, Harvard Business Review, Vol. 28, pp. 45-53.
24. Durbin, M., Niederman, F. (2021), „Bringing templates to life: overcoming obstacles to the organizational implementation of Agile methods“, International Journal of Information Systems and Project Management, Vol. 9 No. 3, pp. 5-18.
25. Elizabeth, M., Sanjana, M. R., Nayantara N. V., Sheetal V. A. (2015), „A Survey on ERP Implementation Techniques“, International Journal of Emerging Technology and Advanced Engineering, Vol. 5 No. 4, pp. 66-69.
26. Forrester, J. W. (1958), „Industrial dynamics“, Harvard Business Review, Vol. 36 No. 4, pp. 37-66.
27. Forsberg, K., Mooz, H. (1992), „The Relationship of System Engineering to the Project Cycle“, Engineering Management Journal, Vol. 4 No. 3, pp. 36-43.
28. Griffin, A. Brandyberry, A. (2010), „Systems Development Methodology Usage in Industry: A Review and Analysis“, Journal of Information Systems Applied Research, Vol. 3 No. 19, pp. 1-17.
29. Haffke, I., Kalgovas, B. Benlian, A. (2016), „The Role of the CIO and the CDO in an Organization"s Digital Transformation“, in Thirty Seventh International Conference on Information Systems, Dublin, Ireland, pp. 1-20.
30. Hedman, J., Lind, M. (2009), „Is There Only One Systems Development Life Cycle?“, in Wojtkowski W., W. G., Lang M., Conboy K., Barry C. (Eds.), Information Systems Development: Challenges in Practice, Theory, and Education, Boston, Springer.
31. Hirschheim, R., Klein, H. K., Lyytinen, K. (1996), „Exploring the Intellectual Structures of Information Systems Development: Action Theoretic Analysis“, Accounting, Management and Information Technologies, Vol. 6 No. 1-2, pp. 1-64.
32. Hoffer, J. A., George, J. F., Valacich, J. S. (2007), Modern Systems Analysis and Design, 5th ed., Upper Saddle River, NJ, Pearson Prentice Hall.
33. Jacobson, I., Booch, G., Rumbaugh, J. (1999), The Unified Software Development Process, Boston, MA, Addison Wesley.
34. Kääriäinen, J., Pussinen, P., Saari, L., Kuusisto, O., Saarela, M., Hänninen, K. (2020), „Applying the positioning phase of the digital transformation model in practice for SMEs: toward systematic development of digitalization“, International Journal of Information Systems and Project Management, Vol. 8 No. 4, pp. 24-43.
35. Kraus, S., Breier, M., Dasí-Rodríguez, S. (2020), „The art of crafting a systematic literature review in entrepreneurship research“, International Entrepreneurship and Management Journal, Vol. 16 No. 3, pp. 1023-1042.
36. Langley, A. (1999), „Strategies for theorizing from process data“, Academy of Management Review, Vol. 24 No. 4, pp. 691-710.
37. Laudon, K. C., Laudon, J. P. (2007), Management Information Systems: Managing the Digital Firm, 10th ed., Upper Saddle River, NJ, Pearson Prentice-Hall.
38. Leon, A. (2009), Enterprise Resource Planning, 2nd ed., New York, McGraw-Hill Education.
39. Levitt, T. (1965), „Exploit the product life cycle“, Harvard Business Review, Vol. 43, pp. 81-94.
40. McConnell, S. (1996), Rapid Development: Taming Wild Software Schedules, Redmond, Washington, Microsoft Press.
41. Microsoft (2014), "Implementation methodology", available at www.docs.microsoft.com (15 August 2022)
42. Microsoft (2021), "Implementation methodology", available at https://docs.microsoft.com/en-us/dynamicsax-2012/appuser-itpro/implementation-methodology (15 August 2022)
43. Mittermeir, R. (1992), „Shifting Paradigms in Software Engineering“, in Proceedings of the 7th Joint Conference of the Austrian Computer Society (OCG) and the John von Neumann Society for Computing Sciences (NJSZT), Klagenfurt, Austria.
44. Mohr, L.B. (1982), Explaining organizational behavior, San Francisco, Jossey-Bass.

45. Moran, A. (2015), Managing Agile - Strategy, Implementation, Organisation and People, Switzerland, Springer International Publishing.
46. Morcov, S., Pintelon, L., Kusters, R. (2020), „Definitions, characteristics and measures of IT project complexity - a systematic literature review“, International Journal of Information Systems and Project Management, Vol. 8 No. 2, pp. 5-21.
47. Newman, M., Robey, D. (1992), „A social process model of user-analyst relationships“, MIS Quarterly, Vol. 16 No. 2, pp. 249-266.
48. Ngereja, B. J., Hussein, B. (2021), „An examination of the preconditions of learning to facilitate innovation in digitalization projects: a project team members perspective“, International Journal of Information Systems and Project Management, Vol. 9 No. 2, pp. 23-41.
49. O'Brien, J. A., Marakas, G. M. (2010), Management Information Systems: Managing Information Technology In The Bussiness Enterprise, 10th Edition ed., New York, McGraw-Hill Education.
50. O'Leary, E. D. (2000), Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce and Risk, Cambridge, Cambridge University Press.
51. Oracle (2016), „Oracle Unified Method (OUM) Oracle's Full Lifecycle Method for Deploying Oracle-Based Business Solutions“, White paper. Oracle.
52. Oz, E. (2009), Management Information Systems, 6th ed., Great Valley, The Pennsylvania State University, Thomson Course Technology.
53. Ozturk, V. (2013), „Selection of appropriate software development life cycle using fuzzy logic“, Journal of Intelligent and Fuzzy Systems, Vol. 25 No. 3, pp. 797-810.
54. Pearlson, K. E., Saunders, C. S., Galletta, D. F. (2016), Managing and Using Information Systems: A Strategic Approach, 6th ed. Hoboken, NJ, Wiley.
55. Pejić Bach, M., Dumičić, K., Žmuk, B., Ćurlin, T., (2020), „Data mining approach to internal fraud in a project-based organization“, International Journal of Information Systems and Project Management, Vol. 8 No. 2, pp. 81-101.
56. Pejić Bach, M., Jaklič, J., Vugec, D.S. (2018), „Understanding impact of business intelligence to organizational performance using cluster analysis: does culture matter?“, International Journal of Information Systems and Project Management, Vol. 6 No. 3, pp. 63-86.
57. Pereira, J., Varajão, J., Takagi, N. (2022), „Evaluation of Information Systems Project Success – Insights from Practitioners“, Information Systems Management, Vol. 39 No. 2, pp. 138-155.
58. Pinto, J. K., Prescott, J. E. (1988), „Variations in critical success factors over the stages in the project life cycle“, Journal of Management Information Systems, Vol. 14 No. 1, pp. 5-18.
59. PMI (2017), A Guide to the Project Management Body of Knowledge (PMBOK Guide), 6th ed., Newton Square, Pennsylvania, Project Management Institute.
60. PMI (2021), A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th ed., Newton Square, Pennsylvania, Project Management Institute.
61. Polli, R. (1968), „A Test of the Classical Product Life Cycle by Means of Actual Sales Histories“, Doctoral dissertation, University of Pennsylvania.
62. Pressman, R. (1997), Software Engineering – a practitioner's approach, 5th ed., New York, McGraw-Hill Higher Education.
63. Richmond, K. (1991), „Information engineering methodology: A tool for competitive advantage“, Telematics and Informatics, Vol. 8 No. 1-2, pp. 41-57.
64. Robinson, P. (2010), "Should you implement ERP with the BIG BANG or phased approach?", available at http://www.bpic.co.uk/faq/bigbang.html (15 August 2022)
65. Royce, W. W. (1970), „Managing the Development of Large Software Systems“, Proceedings of IEEE WESCON, Vol. 26, pp. 328-338.
66. SAP (2017), "Implementing SAP S4HANA, on-premise with SAP Activate", available at https://wiki.scn.sap.com/wiki/display/ATopics/Implementing+SAP+S4HANA,+on-premise+with+SAP+Activate (15 August 2022)
67. Singh, A., Kaur, P. J. (2019), „Analysis of Software Development Life Cycle Models“, in Nath,V., Mandal, J. K. (Eds.), Development Life Cycle Models. Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems, Lecture Notes in Electrical Engineering. Singapore, Springer, pp. 689-699.
68. Soares, J. (2020), „AgileMIP“, Master Thesis, University of Minho.

69. Spohrer, K. (2016), Collaborative Quality Assurance in Information Systems Development: The Interaction of Software Development Techniques and Team Cognition, Switzerland, Springer International Publishing.
70. Thamhain, H. J., Wilemon, D. L. (1975), „Conflict management in project life cycles“, Sloan Management Review, Vol. 16 No. 3, pp. 31-50.
71. Van de Ven, A. H. (1992), „Suggestions for studying process: A research note“, Strategic Management Journal, Vol. 13 No. 1, pp. 169-188.
72. Van de Ven, A. H., Poole, M. S. (1995), „Explaining development and change in organizations“, Academy of Management Review, Vol. 20 No. 3, pp. 510-540.
73. Varajão, J. (2002), „Information Systems Function: Contributions to the successful adoption of information technologies and development of information systems in organizations“, Doctoral dissertation, University of Minho.
74. Varajão, J. (2018a), „agilIS – agile Information systems“, Guimarães, Portugal, Department of Information Systems, University of Minho.
75. Varajão, J. (2018b), „The many facets of information systems (+projects) success“, International Journal of Information Systems and Project Management, Vol. 6 No. 4, pp. 5-13.
76. Varajão, J., Carvalho, J. (2018), „Evaluating the Success of IS/IT Projects: How Are Companies Doing It?“, in International Research Workshop on IT Project Management, a13.
77. Varajão, J., Carvalho, J.Á., Silva, T., Pereira, J. (2022a), „ Lack of Awareness of IT Adoption and Use Theories by IT/IS Project Managers: Poor Relevance, Unfocused Research or Deficient Education?“, Information, Vol. 13 No. 2, pp. a48.
78. Varajão, J., Fernandes, G., Silva, H. (2020), „Most used project management tools and techniques in information systems projects“, Journal of Systems and Information Technology, Vol. 22 No. 3, pp. 225-242.
79. Varajão, J., Magalhães, L., Freitas, L., Rocha, P. (2022b), „Success Management – From theory to practice“. International Journal of Project Management, Vol. 4 No. 5, pp. 481-498.
80. Varajão, J., Trigo, A., Pereira, J. L., Moura, I. (2021), „Information systems project management success“, International Journal of Information Systems and Project Management, Vol. 9 No. 4, pp. 62-74.
81. Verhage, L. (2009), Management Methodology for Enterprise Systems Implementation, Delft, Eburon Academic Publishers.
82. Welke, R. J. (1983), „IS/DSS: DBMS Support for Information Systems Development“, in Holsapple, C. W., Whinston, A. B. (Eds.) Data Base Management: Theory and Applications, Dordrecht, Springer, pp. 195-250.
83. Wong, W. Y., Yu, S. W., Too, C. W. (2018), „A Systematic Approach to Software Quality Assurance: The Relationship of Project Activities within Project Life Cycle and System Development Life Cycle“, in 2018 IEEE Conference on Systems, Process and Control, Melaka, Malaysia, IEEE, pp.123-128.
84. Xia, W., Lee, G. (2005), „Complexity of Information Systems Development Projects: Conceptualization and Measurement Development“, Journal of Management Information Systems, Vol. 22 No. 1, pp. 45-84.
85. Xu, J. (2019), Essential Topics Of Managing Information Systems, Singapore, World Scientific Publishing Company.

About the authors

Beatriz Meneses is an Information Technology/Information Systems consultant at Accenture. She has an Integrated Master in Engineering and Management of Information Systems (MIEGSI) from the University of Minho. The author can be contacted at **a78260@alunos.uminho.pt**

João Varajão is currently a professor of Information Systems (IS) and Project Management (PM) at the University of Minho. He is also a researcher at the ALGORITMI centre. His current research interests are IS PM and IS Development (addressing IS and PM success). Before joining academia, he worked as an IS consultant, project manager, IS analyst, and software developer, for private companies and public institutions. He has published numerous refereed publications, authored and edited books, as well as book chapters and communications at conferences. He serves as editor-in-chief, associate editor, and member of committees for conferences and international journals. The author can be contacted at **varajao@dsi.uminho.pt**