

# Analysis and Implementation Phases in the Two-Segmental Model of Information Systems Lifecycle

Jędrzej Wiczorkowski  
Warsaw School of Economics  
Al. Niepodległości 162  
02-554 Warszawa, Poland  
Email:  
jedrzej.wiczorkowski@sgh.waw.pl

Przemysław Polak  
Warsaw School of Economics  
Al. Niepodległości 162  
02-554 Warszawa, Poland  
Email: ppolak@sgh.waw.pl

1

**Abstract**—Analysis and implementation phases in the lifecycle of ERP software packages involve many resources and are most relevant to the buyers of such software. Therefore, it is important for them to understand the objectives of those phases and activities involved with them. The two-segmental model proposed by the authors is aimed at better representation of the life cycle of information systems. This article aims to demonstrate that the actual course of the two phases is better mapped in the proposed model. The classical waterfall model of the life cycle of information systems was used as a reference point for the investigation.

## I. INTRODUCTION

**B**UILDING and implementing information systems is a difficult and complicated process, burdened with a high risk of failure. Models of the lifecycle of information systems allow to understand better that process and thus make possible the reduction of risks associated with the implementation. They are widely used in the content of textbooks and courses on software engineering and information systems development. However, in order to achieve that aim, those models must appropriately reflect the actual activities carried out during the process.

## II. CLASSIC MODELS OF THE INFORMATION SYSTEMS LIFECYCLE

The waterfall model of the software lifecycle is one of the earliest attempts to describe the full life cycle of information systems, in particular the production process. Originally it was published by Royce in 1970 [1]. In its initial form, each phase included feedback to previous phases. Later, in practice, the model was often confined to a strictly linear form, treating it as synonymous to the sequential approach with identified specific phases (see Fig. 1) [2], [3].

The practical usefulness of the waterfall model receives sometimes radically different assessment. On the one hand, it is claimed that no real big project was carried out strictly in accordance with that model. On the other hand, when treated more flexibly, particularly allowing reasonable iterations, most real-life information systems projects match up that model [4]. Generally, the waterfall model is useful to describe projects, in which it is unlikely to return to work

done in previous phases and the final products of those phases remain unchanged.

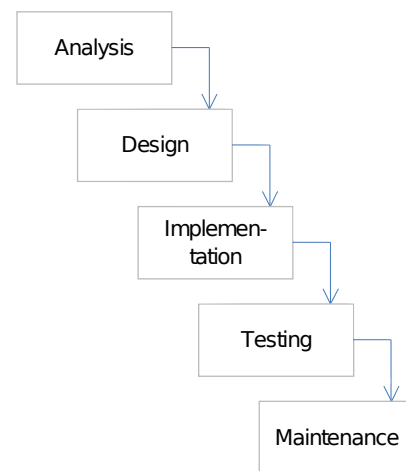


Fig. 1 The basic form of the waterfall model

In order to eliminate the fundamental disadvantages the waterfall model, such as the imposition of strict work sequence, the high cost of errors made in the initial stages or long breaks in direct relations between customers and producers, many other models were proposed, primarily using iterative connections (incremental and spiral models) and prototyping. Also, more complex models were developed including [2], [3], [5], [6], [7], [8]: „b” and „V” models, a parallel model, a database structure design model, a formal methods model, an extreme programming model, a formal transformation model, models for structured methods and object-oriented models. However, the waterfall model was the starting point for most of those concepts, and it became the basis for the development of structured analysis and design methods (for example SSAMD - Structured Systems Analysis and Design Method) [2], [3].

The majority of life cycle models in software engineering literature reflects the process of producing completely new software. However, the tendency to apply the extended usage of existing elements is observed in the practice of software development. That approach is a basis for the object-oriented methods of building systems, a reusable

software and the component-based development model, as well as the usage of standard software packages.

Systems supporting the operational activities of companies and broadly defined management support (Enterprise Information Systems), including, in particular ERP systems, are generally based on the concept of standard software packages. The same software products after adaptation to the needs of specific companies (in the process of customizing) are used in organizations from various industries characterized by the different activities. The typical software life cycle models proposed by software engineering are designed for systems dedicated to a particular customer (custom software). In terms of ERP systems, this approach is used very rarely. In the case of standard software packages, the classical approach towards the life cycle describes only the steps in the production of a system, aside from the issues of implementation. In the literature on software engineering, the acquisition of standard systems is only mentioned (for example the concept of packet-based solutions [3], but it can not be recognized as a comprehensive model.

Thus, the literature describing the ERP systems contains attempts to describe and model the life cycle of software packages. For example:

- O'Leary [9] distinguished the following processes, which must be completed by a company implementing ERP class system: deciding to go system, choosing system, designing, implementing, operation after going live, training.
- Similarly, Ross and Vitale [10] differentiated sequenced steps: design, implementation, stabilization, continuous improvement, transformation.
- Flasiński [11] distinguished three main sequential phases: preimplementation analysis, system selection with signing a contract and proper implementation.
- Evolutionary approach proposed by Harwood [12] includes 5 spirally linked phases: needs, vendor selection, implementation, go live and review, improvement.

The issues of work performed by the customer or implementation partner during system acquisition was also discussed by Lenart [13], Ray[14], and Leon[15].

Worthy of note is the paradox – almost all of the above approaches are based on the classical linear approach when describing customer-side actions. The description of the software packages life cycle in software engineering theory is significantly different from the practice of systems implementation, which reflects the implementation methods offered by the leading vendors of ERP systems. Therefore in this article, the classical waterfall model is adopted as a reference point to discuss the characteristics of software packages implementation.

### III. IMPLEMENTATION METHODOLOGIES

An integrated management information system implementation methodology can be defined as structured, formal and complete description of activities conducted by

an implementation team during the implementation of the system. Those activities are organized into stages and phases [16]. The implementation methodology is a plan and a guide, which makes the implementation process easier and more efficient. However, using it never fully guarantees the success of a project.

The usefulness of implementation methodologies comes from the fact that those projects are often extremely complex. In contrast to the software design projects, which are carried out in IT companies, implementation projects take place in various firms of different industries, and project teams contain not only professionals but also persons not connected on a daily basis with IT [11]. This gives rise to communication problems, arising inter alia from poor preparation of professionals to understand the business processes of enterprises and the poor knowledge of modern technologies among the team members from organizations, in which systems are to be implemented. As a result, producers and vendors of standard information systems increasingly recommend the formalization of implementation approaches and the use of methods that include solutions, which have been proven to be effective. The ASAP (Accelerated SAP) methodology and other approaches developed by SAP – a leading provider of integrated management information systems – are the perfect example of an implementation methodology and its ongoing evolution. In recent years, the introduction to the market new products (R/3, mySAP.com, mySAP Business Suite) was also accompanied by the change of offered and used in practice implementation methodologies and tools supporting them [17], [18].

There are significant differences between the implementation methodologies and the methodologies aimed at improving the software development process, which can be described as design and production methodologies. The second group of methodologies is much more general. Those methodologies are usually based on structural or object-oriented approach (sometimes the social approach is also distinguished) and they typically use formal or agile methods. However, that group is not practically applicable in the implementation of standard software packages.

### IV. THE TWO-SEGMENTAL MODEL OF INFORMATION SYSTEMS LIFECYCLE

The Two-Segmental Model of Information Systems Lifecycle, developed by the authors [19], represents a typical lifecycle of highly parameterized software packages. The main feature of the model is its division into two segments (see Fig. 2):

- Segment I – activities occurring on the side of a system vendor,
- Segment II – activities occurring on the side of a customer.

The first segment, in its principal run, is completed only once. Re-runs are possible in case of the development of new versions. The second segment is run multiple times. It is repeated independently for each customer.

The article focuses only on two phases of the customer segment: the implementation analysis and the proper implementation. The combination of those phases is justified by the characteristics of the ERP systems implementation methodologies, which often consider those phases together as one phase.

V. IMPLEMENTATION ANALYSIS

The implementation analysis is the first phase after selecting a system. Earlier analytical work, carried out in an organization in which the software is implemented, is aimed on choosing the system appropriate for the needs of the organization.

The implementation analysis phase and implementation phase can be performed only when the result of the system selection and the concept of its acquisition are known, as a way to carry it out can vary, for example, depending on the implementation methodology. The analysis should lead to the development of a detailed specification defining the functions of the system, including the characteristics of its components, the principles of communication within and outside the system, as well as the conditions of use.

The analysis may be done in the traditional manner, i.e. through the specification of an IT strategic plan and the preimplementation analysis. In this case, the implementation analysis is produced by studying the existing state of information systems and management procedures in an organization. Some methodologies (e.g. ASAP), focused on the speed of action, limit the importance of the implementation analysis and instead propose predefined

solutions addressing specific business sectors and regions. This approach can be described as an accelerated analysis methodology. Such solutions are supported by the reference models of business processes, which are one of the customization methods. They are based on the recommendations of organizational settings, designed for the most efficient use of the system in the specific type of business. In this case, the company should implement the reference model using, for example, the ready-made maps of business processes. Some proprietary methodologies refer to such activities as implementation modeling, stressing that the system model is formed primarily by the location and parameterization of system solutions, and not through the identification and detailed analysis of customer processes. The use of business process reference models in many cases seems to be a reasonable compromise between ensuring the flexibility of solutions and maintaining the internal integrity of the software package.

The use of the accelerated method of analysis, particularly the use of the reference models of key business processes, can incur a risk of the partial loss of competitive advantage. Such approach enables to use best practices in an industry, but in case of implementations in market leaders, this argument is questionable at best. In such cases, customers should take into account the possibility of the strong customization of such activities, often in spite of recommendations from implementation partners, which, by simplifying the implementation, work in their own interest. On the other hand, excessive process customization can lead to the disproportionate individualization of the system, what

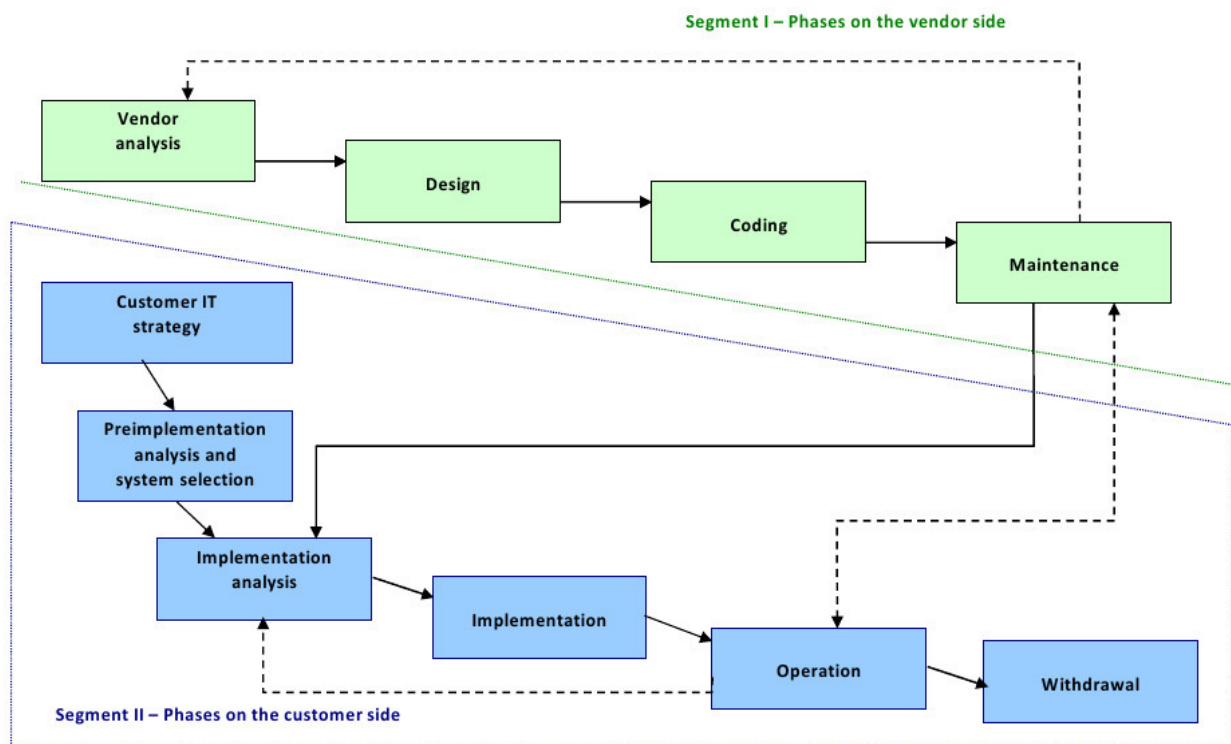


Fig. 2 The two-segmental model of software packages lifecycle

is associated with an increase in the complexity of the project and, in the long run, the difficulty of software maintenance. These problems usually do not occur in case of supportive or typical processes, which are implemented throughout the industry in a very similar way.

In real life, the details of the procedure used during the implementation analysis can vary greatly depending on the chosen solution, which may be based, for example, on the customization of the basic version of a system or on obtaining a predefined industry solution. Regardless of whether the two phases, the implementation analysis and the proper implementation, are executed separately or are merged in one phase, they are in practice a single project. The analysis is the basis for keeping the costs of the project on the intended level. Thanks to it, the difficulties in the functioning of organizations linked to the acquisition system are minimized. For the project success, the fundamental aspects of the organizational set up of each project are particularly important. They include: the establishment of the organizational structure of the project, defining the objectives of the project (if possible, measurable and verifiable), defining the scope of the project, developing the project schedule with the definition and valuation of resources and a timetable for the deployment of funds, the application of the chosen methodology, along with the rules of the supervision and acceptance of the project and with risk, quality and change management.

In order to make the comparison of the implementation analysis phase from the two-segmental model of information systems lifecycle with the waterfall model of software lifecycle, it is necessary to refer to a phase usually named as the requirements analysis, and partly to the design phase. There are many common elements in the corresponding phases, particularly in case of the traditional approach to the implementation analysis. But even in such cases, the implementation analysis does not distinguish between structured and object-oriented approaches. Moreover, the implementation analysis is dominated by the process-oriented approach, particularly in case of the accelerated analysis.

## VI. SYSTEM IMPLEMENTATION

The implementation phase of software packages can be divided generally into two main concurrently performed set of activities:

- the adaptation of a system to the specific needs of an organization (technology customization),
- the adaptation of business procedures and processes to opportunities arising from system capabilities.

Both these processes appear to be equally important. On the one hand, management solutions are primary to IT solutions [20]. On the other hand the implementation of a computer system may become a stimulus to carry out the optimization of management solutions. The customization of a system can be carried out using various methods, depending on adjustment mechanisms available in a system. Every system contains a limited set of available programming and non-programming methods [21].

The extended usage of programming methods can justify the formal separation of a new subphase - the design on the customer side. That subphase is very similar to the design phase in the classical model. However, using programming methods is an option, therefore, in the two-segmental model, that subphase is not specified and any design work on the customer side belongs to the implementation phase.

The second group of activities, the adaptation of organizational procedures and processes may be associated with business processes reference models, with the implementations of changes resulting from the thorough reorganization of business processes in accordance with BPR concept, or with continuous process reorganization following the implementation of BPM (Business Process Management) strategy. Regardless the notion of changes, the process approach dominates the implementation phase, in the same way as in the implementation analysis phase. The process approach is also typical for contemporary implementations of ERP software packages.

In the case of the implementation phase formed on the basis of the waterfall model, the group of activities related to organizational changes is also included. However, the formal software production models do not devote enough attention to those changes.

In addition to the two main group of activities performed during the implementation phase, this phase includes, inter alia, such important activities as:

- the adaptation of system technical infrastructure, the management of operating environments and software installation,
- system integration with other applications,
- defining and implementing the principles of system administration and management,
- system testing,
- training for future users and the preparation of user instructions,
- the migration of data from existing systems.

The adaptation of system technical infrastructure includes, apart from the hardware setup, also the configuration of software such as operating systems, database management systems, etc. The management of operating environments involves building technical infrastructure and software installation. The separation of environments is associated with ensuring the safety and efficiency of the system by separating the functions of individual servers, software or data areas. Systems operate in several different roles, such as ongoing operation, development and training. Hence, most software vendors calls for the separation, in the form of individual installations or at least the areas of data, of operational system environments such as: production, training, development, testing and quality assurance, and also a spare security environment.

Activities related to the integration (merging) with other systems in the organization can be very labor consuming. They usually involve programming and include, in particular, the construction of interfaces for data and messages exchanged between applications. They are

particular important in the case of solutions based on components and service-oriented architecture (SOA). The process of implementing the integration platform, in most cases, should be treated as a separate IT project, which does not fall into the software package life cycle.

Implementation of software, the same as in the case of software development, requires defining and implementing the principles of system administration and management. These principles have huge impact on the system security. Within this group of tasks, it is particularly important to develop and implement the concept of access rights.

The system tests performed during the implementation phase may also include the tests of new programs and the test of software package modules: function tests, integration tests, stress tests and acceptance tests.

Training related to the project usually begins before the implementation phase. Such training involves management staff (in order to demonstrate the general capabilities of the system) as well as the staff assigned to the project. It can continue during the implementation phase. However, the essential training at this stage is addressed to future end users and system administrators. In the case of the large integrated systems, it can be very labor intensive because of the number of participants. The training involves also preparing user manuals.

The migration of data often requires the development of tools dedicated for this purpose. As with any software project, small systems lifecycles can be identified. They include their own analysis, design and coding phases.

The above-mentioned additional groups of activities are very similar to the actions associated with the implementation phase in the traditional waterfall model of software life cycle. Hence, both the two segmental model and the waterfall model similarly describe technical activities of the implementation phase. Whereas, differences are observed in the approaches towards business related activities.

The final outcome of the implementation phase is the conversion of information systems and the start of the production system. In the implementation of integrated systems, different approaches to the sequence of implementing modules can be used: sequential (step by step), overall (big bang), mixed – initially highest priority modules (middle size big bang). The overall approach is supposedly the most effective. It gives the possibility of parallelization of some work, but may be problematic and risky in case of large projects. The practice of implementations proves popularity of mixed methods in which the selection of the first group of modules results from the efforts to maximize synergy benefits from the first step of the conversion. Usually the first step includes highly integrated modules, crucial for implementing main business processes, for example: financial accounting, inventory management, procurement and sales. The following steps involve more modules, e.g. human resource management, controlling, etc. The sequential method can be ineffective due to the long duration of the project and, as a consequence, long waiting for its benefits.

## VII. CONCLUSION

The main feature that characterizes the two-segmental model of the information systems life cycle is the separation of vendor and customer segments, what reflects the main characteristic of software packages. However, also particular phases distinguished in the two-segmental model, and their sequence, differ from classical life cycle models. Traditional sequence (analysis, design, coding, implementation) was broken and the phases were divided between both segments. In the customer segment, two phases: the implementation analysis and implementation were distinguished. They correspond to the implementation methods developed by the vendors of software packages and, as those methods are widely used, to the practice of the IT market.

Software packages implementation project are characterized by the strong impact of the process approach on the phases of implementation analysis and implementation. But the process approach is also used in the classical software engineering – e.g. data flow diagrams, which have long been used in the structural analysis, are in fact process models. However, modern process approach, is not limited to the process transformations and the flows of data, but attempts to map business processes in organizations. Therefore, business processes play an important role in the implementation analysis, and the reference models are widely used in the customization of software packages.

## REFERENCES

- [1] W. Royce, "Managing the Development of Large Software Systems", in *Proceedings of IEEE WESCON*, vol. 26, August 1970.
- [2] R. S. Pressman, *Software Engineering: A Practitioners Approach*, New York: McGraw-Hill, 2001, ch. 2.
- [3] J. Cadle, D. Yeates, *Project Management for Information Systems*, Harlow UK: Pearson, 2008.
- [4] A. Jaszkiwicz, *Inżynieria oprogramowania*, Gliwice: Helion, 1997, pp. 15-28.
- [5] A. Kobyliński, „Inżynieria oprogramowania. Modele cyklu życia oprogramowania” in *Wybrane zagadnienia informatyki gospodarczej*, J. Płodzień Ed., Warszawa: Szkoła Główna Handlowa w Warszawie, 2004, pp. 13-38.
- [6] I. Sommerville, *Software Engineering*, New York: Addison-Wesley, 2011, ch. 2.
- [7] W. Chmielarz, *Zagadnienia analizy i projektowania informatycznych systemów wspomagających zarządzanie*, Warszawa: Wydawnictwa Naukowe Wydziału Zarządzania Uniwersytetu Warszawskiego, 2000.
- [8] L. Maciaszek, B. L. Liong, *Practical Software Engineering: A Case-Study Approach*, Harlow: Pearson / Addison Wesley, 2005.
- [9] D. O’Leary, *Enterprise Resource Planning Systems – Systems, Life Cycle, Electronic Commerce and Risk*, Cambridge UK: Cambridge University Press, 2005.
- [10] J. Ross, M. Vitale, “The ERP Revolution: Surviving vs. Thriving”, *Information Systems Frontiers*, vol. 2 issue 2, August 2000, pp. 233-241.
- [11] M. Flasiński, *Zarządzanie projektami informatycznymi*, Warszawa: Wydawnictwo Naukowe PWN, 2006, ch. 3-6.
- [12] S. Harwood, *ERP – The Implementation Cycle*, Oxford: Butterworth-Heinemann, 2003.
- [13] Lenart A., „Wdrażanie i użytkowanie systemów ERP”, in *Informatyka Ekonomiczna. Podręcznik akademicki*, S. Wrycza Ed., Warszawa: Polskie Wydawnictwo Ekonomiczne, 2010, pp. 343-372.
- [14] R. Ray, *Enterprise Resource Planning. Text & Cases*, New Delhi: Tata McGraw-Hill, 2011, pp. 27-55.
- [15] A. Leon, *Enterprise Resource Planning*, New Delhi: Tata McGraw-Hill, 2008, pp. 124-138.

- [16] G. Bartoszewicz, *Projektowanie wdrożenia modułów logistycznych zintegrowanych systemów klasy ERP. Podejście procesowe*, Poznań: Wydawnictwo Akademii Ekonomicznej w Poznaniu, 2007.
- [17] V. Kale, *Implementing SAP R/3: The Guide for Business and Technology Managers*, Indianapolis: Sams Publishing, 2000, ch. 6
- [18] SAP, <http://help.sap.com>
- [19] J. Wieczorkowski, P. Polak, "The Two-Segmental Model of Software Packages Lifecycle" in *Proceedings of the IADIS International Conference Information Systems 2011*, P. Powell, M. B. Nunes, P. Avila: International Association for Development of the Information Society Press, 2011, pp. 291-295.
- [20] M. Skwarnik, „Elastyczne informacyjne systemy zarządzania. Mechanizmy realizacji”, in *Prace Naukowe Akademii Ekonomicznej we Wrocławiu* no 955, E. Niedzielska, H. Dudycz, M. Dyczkowski Eds., Wrocław 2002, pp. 444-454.
- [21] J. Wieczorkowski, P. Polak, „Customization of Software Packages – Technology and Business Process Perspectives”, in *Proceedings of the IADIS International Conference Information Systems 2010*, M. B. Nunes, P. Isaiás, P. Powell, Eds., Porto: International Association for Development of the Information Society Press, 2010, pp. 549-552.