# A Process for Distributed Software Evolution

A proprietary software case study

Joao Ricardo M. Camilo
Federal University of Technology – Parana (UTFPR)
Cornelio Procopio, Brazil
jrmcamilo91@gmail.com

Alexandre L'Erario
Federal University of Technology – Parana (UTFPR)
Cornelio Procopio, Brazil
alerario@utfpr.edu.br

Tiago Pagotto
Federal University of Technology – Parana (UTFPR)
Cornelio Procopio, Brazil
pagotto@alunos.utfpr.edu.br

Jose Augusto Fabri
Federal University of Technology – Parana (UTFPR)
Cornelio Procopio, Brazil
fabri@utfpr.edu.br

## ABSTRACT

Proprietary enterprise software is commonly embedded in multinational organizations and therefore has multiple sources of global or local demand. This type of software is subject to constant evolutions motivated by improvements in the features or by changes in the legal and economic context of its environment. Problems arising from the demand for requirements are associated with the suitability of a particular local requirement to its global context, for example: tax rules are characteristic of a specific country while a business rule can achieve every global structure. In addition, software customization can be implemented by the software producer, external partners, programmers allocated within the customer company, and so on. The coordination of this scenario is considered critical for the productive sector company that uses proprietary enterprise software. If the software does not evolve, the company processes can be temporarily compromised, and the software becomes obsolete. We report a case study of a large proprietary ERP system in a multinational company located in Brazil that is among the three largest exporters in its market segment. As a contribution, we present the current structure of the distributed evolution process of the software in question and how the stakeholder coordination of this scenario occurs.

## KEYWORDS

Distributed Software Development, Software Evolution, Distributed Software Evolution

**ACM Reference Format:**
Joao Ricardo M. Camilo, Alexandre L'Erario, Tiago Pagotto, and Jose Augusto Fabri. 2018. A Process for Distributed Software Evolution: A proprietary software case study. In *ICGSE '18: 13th IEEE/ACM International Conference on Global Software Engineering , May 27–29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3196369.3196376


*ICGSE '18, May 27–29, 2018, Gothenburg, Sweden*

ACM ISBN 978-1-4503-5717-3/18/05…$15.00
https://doi.org/10.1145/3196369.3196376

## 1 INTRODUCTION

Since the emergence of software engineering, researchers and practitioners have struggled to deal with constant demands for changing requirements to evolve software systems. This evolution poses several research challenges, especially in large and complex systems [34].

This work deals with the study of a scenario that involves the context of Software Evolution (SE), where an ERP system, considered a large and complex enterprise system, undergoes constant evolutions motivated by changes from multiple distributed sources, such as company policies, technology, standards, legal and economic context. But also mixes with the Distributed Software Development (DDS) concepts, which is characterized as a scenario in which multiple distributed stakeholders are involved in a development process of the same project.

In practice, software systems evolve over time to remain satisfactory and respond to any changes in their environment [14] [25]. Changes are inevitable in any software system. To bring the global organization to a common platform, different countries demand different requirements [11]. It is a significant challenge to keep the operation of software with high quality. For example, over the time and with excessive changes, software becomes more complicated and tracking changes is not an easy task [21].

In the context of enterprise systems, Enterprise Resource Planning (ERP) systems are one of the most common information system solutions adopted by organizations around the world. ERP systems are large and complex, as they include all the processes, interactions, and financial transactions of a company on the same platform [16].

Therefore, in the context of this work, we define the term Distributed Software Evolution (DSE) as an adaptation of Distributed Software Development [24] [5] added to the characteristics of Software Evolution [10]. This concept involves demands from different sources, and different players that have to collaborate to enable the implementation of these demands.

This Distributed Software Evolution is supported by stakeholders who demand requirements and stakeholders that implement such requirements. Stakeholders who want requirements are the stakeholders associated with the organizational strategy, departments of the industry, international agencies and also governments with standards and law; while the stakeholders that implement these requirements are associated with the software

producer, external partners, ad-hoc consultants, programmers allocated within the industry, and so on.

There is a large literature, in fact, dealing with advanced methods and tools for modeling components of ERP systems [22] [32], align ERP system functionalities with business requirements [15] [33] and optimize implementation of ERP systems [29] [37]. However, the literature on ERP systems focuses primarily on the ERP implementation and early usage phases and critical success factors for post implementation. [9] [26] [31].

The goal of this research detailed herein is to identify the current structure of distributed evolution process in the software in question and the way the stakeholders' coordination of this scenario occurs. This approach is evaluated through a case study in a customer company of a proprietary ERP system. The question we want to answer is: How is the current structure of the Distributed Evolution Process of a proprietary enterprise system? The contribution of this work is in obtaining DSE evidence from an industry with 15 years of experience using an ERP system.

The rest of this article is organized as follows. In Section 2, a background is presented. In Section 3, we detail the research method, followed by a presentation and discussion of the case study in Section 4. In Section 5, an overview of related work is presented. Finally, Section 6 concludes our work and suggests guidelines for future work.

## 2 BACKGROUND

### 2.1 ERP systems

Enterprise Resource Planning is defined as a type of enterprise system that allows the complete integration of information from all functional areas into company through a database, and accessible through a unified interface [1].

An ERP system is a modularized or packaged software [17] that allows customization or addition of modules or packages by third parties, not being developed as other types of software [12].

ERP systems inevitably evolve to align their functionality with changing business requirements [19] [35]. Following the evolution of software developments, ERP systems post implementation customizations can be driven by internal reasons. Such as a new and more efficient logistics management policy suggested by the customer company that should be reflected in the ERP system; or by external factors, such as suppliers, technology providers or regulatory agencies [9]. For example, an ERP system may require updates to accommodate new tax regulations by financial authorities to address emerging money laundering practices [21]. Under an ERP system, Rothenberger and Srite [27] define customization as the modification of an ERP system package to match business processes in the organizational context.

A proprietary ERP system, or proprietary enterprise system, is a software product developed by a private company. It is important to note that proprietary ERP system has specific/limited access and implementation interfaces, and vendor does not always provide all source code resources for third parties to modify [8].

ERP system projects differ from other types of software projects since they cover thousands of business activities, require various configurations and modification activities to reflect immediate business requirements [6] [16].

### 2.2 Software Evolution

The successful software requires constant changes that are triggered by evolving requirements, technologies, and knowledge. These constant changes consist of the Software Evolution. The software evolution has gained importance and has recently moved to the attention of software developers. As a consequence, most software development now takes place at the software evolution stage. This event has changed the face of software engineering [25].

There are several definitions of Software Evolution and Software Maintenance. Some authors refer to software evolution when new features or another aspect of the software is improved, and some authors relate to software maintenance mainly to the adaptations and fixes to keep the software running [14]. Since software maintenance objectives aim at reduced goals, maintenance processes are usually more straightforward and more predictable than the software evolution processes [25]. In this study, we adopted evolution and software maintenance through these concepts.

Practitioners involved in software evolution consciously or unconsciously confront some of the constraints imposed by the laws of software evolution that Meir M. Lehman introduced in the 1970s. These laws or empirical hypotheses, suggest that any actively used software system must continually change to satisfy its stakeholders [10] [13]. Therefore, to remain satisfactory, these programs must be continually changed and updated. Lehman's studies have examined mainly proprietary and monolithic systems developed by identifiable teams of developers within an industrial environment, while open source systems and their development seem to be subject to very different evolutionary pressures [8].

The acceptability of the ERP system evolution depends on the results delivered to users and other stakeholders. They must be continuously improved, adapted and fixed if they want to remain effective within a constantly evolving environment. Thus, the evolution of such systems is a complex phenomenon characterized by a multi-level, multi-loop and multi-agent feedback [14].

### 2.3 Proprietary vs Open Source

Most studies on software evolution address the evolution of open source software [38] [2]. It is known that the objectives, processes, economics and even the policy of open source software development are surprisingly different from those of most proprietary systems. Although there are a lot of open source systems that can be studied freely, the vast majority of commercial software is still created as a proprietary and closed source.

Proprietary software studies are often subject to restrictions that limit the details that can be published. This poses two fundamental problems: first, many empirical studies cannot easily be found, replicated, or extended; and second, many details are usually left out of the study [8].

The fact is that the solutions that revolve around the open source system come from a community of developers and can make it difficult to meet local requirements. Proprietary systems solutions come from the software producer itself and specialized external partners [11] [25].

In the proprietary ERP system, there is a zero or partial opening of source code that customers and external partners can modify, resulting in a great dependence on the software producer [11].

However, the proprietary ERP system software producer ensures a greater chance of implementing legal requirements on time. Also, it must be considered that each country has its own local communities to adapt global and local requirements to open source software. In the case of open source software, there is no guarantee of implementing all the necessary requirements from the developer community on time.

Considering this information, it can be concluded that the evolution process of an open source system may differ from the proprietary system in several characteristics [11]. Therefore, studying the evolution of open source software is a worthy subject on its. [8].

# 3 RESEARCH METHOD

## 3.1 Research Goal

We set out to answer the following research question by conducting a single case study [39]:

*RQ: How is the current structure of the Distributed Evolution Process of a proprietary enterprise system?*

## 3.2 Case Study

In this work, we present a qualitative single case study of the distributed evolution process of a proprietary enterprise system through the study of the distributed evolution of an ERP system in a customer company.

The case study is considered an exploratory research technique that investigates a contemporary phenomenon in the context of real life [39]. Case studies have been increasingly adopted in software engineering in general [36]. Based on Yin [39] approach, case studies can be classified into three different types: descriptive, exploratory, explanatory. For this research, a descriptive and exploratory type of study was considered more appropriate. The objective of this research was not to test or validate any theory, but to identify the phenomenon of distributed evolution of software in a proprietary enterprise system.

Trust in a single case study has been criticized for its limited generalization [36]. However, Yin [39] observes its adequacy according to five rationales, specifically when the case is: critical, unusual, common, revealing or longitudinal. The revelatory circumstance is employed in this research, which means that the use of a single case is acceptable. Although changes in ERP systems are not uncommon in private sector organizations, this phenomenon is generally not accessible to researchers as in open source systems [2]. This defines the case under consideration as revealing.

## 3.3 Data Collection

Data collection is an important part of a case study. In this sense, [39] identified seven sources of empirical evidence in case studies:

*3.3.1 Documents:* written documents (published and unpublished), company reports, memos, administrative and departmental reports, e-mail messages, newspaper articles or any document that presents some evidence/information.

*3.3.2 File records:* can be service records, organizational records, name lists, search data and other records.

*3.3.3 Interviews:* interviews can be used to identify variables and their relationships mainly to find data.

*3.3.4 Questionnaires:* structured questions written and provided to multiple interviewees.

*3.3.5 Direct observation:* obtaining data through industry visits.

*3.3.6 Participant-observation:* intervention in the case study.

*3.3.7 Physical Artifacts:* physical artifacts can be instruments, tools, or some physical evidence that can be collected as part of a field visit, primarily for data collection.

In this case study we collected documents, records, interviews and direct observation. The researcher visited an ERP system costumer company to find out how the structure of the distributed evolution process of a proprietary ERP system works from the customer company perspective. He collected documents that identify business processes and describe their current operations. This helped the researcher learn details about post implementation projects management, including projects paths and stakeholders' involvement .

Archive records are an integral part of the data that needs to be collected. The main records that have been used are the e-mail exchanges between stakeholders. Project folders were accessed, where external partner service contracts were allocated, spreadsheets with project schedule records, user test worksheets, problem records and solutions found during the project.

To know the day-to-day work of the team on different projects, and to understand how the current process works, direct observation was essential. Direct observation was made during 3 months in 4 large projects executed during the period.

Semi-structured interviews were also conducted. For a current context overview, we first interviewed the IT team coordinator who could give us a good picture of the history, structure and usual practices. The rest of the interviewees involved the team of 6 internal consultants of the customer company. Each consultant is responsible for an ERP system module. Therefore, each participant has different roles and different experiences, each of them between five and nineteen years of experience with the ERP system. The interviews were relatively unstructured and conversational to maintain adaptability to the roles and individual experiences of employees in different roles. Respondents were invited to describe their own experiences in the context of ERP system evolution, how the distributed development was, the usual practices and the successes and challenges in the project related to the evolution of the ERP in the context in which they are. About 0.5 hours of interviews were spent with each internal consultant, totaling 3 hours of interviews.

These records helped the researcher to draft an initial structure of the distributed evolution process of the proprietary ERP system. We performed archival analyses to further investigate and validate findings from the interviews.

## 3.4 Data Analysis

The first step in data analysis focused on ensuring that research instruments were valid and reliable. Therefore, in the first moment, folders and e-mail exchanges were analyzed in some closed projects.

This aimed to collect the practices used by the company, evidence related to the distributed evolution and the relationship between stakeholders.

This rich mine of data brings historical data from multiple projects, considering its opening to closure. This makes it possible to consult the successes, delays, difficulties, errors, lessons learned, among other characteristics. Given the potential importance of the historical data of the projects, we took advantage to extend the period of analysis of the documents. Most of the documents analyzed were dated between January and December 2017.

Main topics discussed in the interviews were transcribed by the author. These topics were coded for the same concept of usability. The transcribed interviews were analyzed by qualitative coding.

Based on all these considerations, we concluded that the measures used to analyze the data collection had adequate properties.

# 4 CASE PROJECT

## 4.1 Case Organization

The company is a multinational company present in 2 countries (Brazil and United Kingdom) and is a subsidiary of one of the 5 largest Japanese trading companies. The unit analyzed, located in Brazil, is among the three largest exporters of its market segment in the country. Currently the company has about 800 employees. There are 650 in the factory, 100 in another branch (same state), 40 in branches (other states) and 10 abroad.

ERP system implementation was in 1999. All employees of the customer company use the ERP system in some way, either in the new product register, in the records of production, or in the strategic analysis issued by the system. Therefore, all users impact and are impacted by the ERP system use. Because it is an industry, it usually counts on the functional areas of Production, Purchasing, Sales, Quality Control, Information Technology, Accounting, Finance, Logistics, among others. Most accounting processes, payments, receipts, electronic invoice for Brazil, quality control, label control, batch control and production, permeates through the ERP system, which integrates most of the existing transactions in the company.

The ERP team works as ERP system support. Each responsible for an ERP system module, and 2 internal programmers. We collected ERP team data as shown in Table 1. We considered studying this team because of their years of experiences. The team has about 15 years of experience in the ERP system. This gives us a large historical data on ERP system projects, with multiple distributed stakeholders relationship during these years. The internal programmers ate responsible for implementing the improvements that are possible to be implemented by the customer company. The team has already received the services of more than 10 external partners during 19 years of ERP use. These external partners were from Brazil and United Kingdom. Customer company also received support from the software producer, from Germany, and its local subsidiary, located in Brazil, during this period. This scenario makes the case study company a rich source of historical data experience, considering that the ERP system implementation was in 1999.

The ERP system is made up of two environments, production and test. Data from the test environment database is replicated from the production base annually, and stakeholders can access real data from previous real scenarios. ERP system also has versioning control. However, when major changes are performed, that are considered evolution processes, the restoration of previous versions becomes impractical, as there are a large number of artifacts and data involved. The solution to this case is the complete restoration of previous ERP system backups.

**Table 1: Customer Company ERP team**

| Employee | Module | ERP experience | Age |
|---|---|---|---|
| C1 | Financial | 7 years | 28 |
| C2 | Accounting | 19 years | 52 |
| C3 | Sales | 5 years | 27 |
| C4 | Logistics | 19 years | 38 |
| C5 | Purchase | 19 years | 43 |
| C6 | Production | 19 years | 52 |
| P1 | Programming | 19 years | 58 |
| P2 | Programming | 8 years | 34 |

It is important to emphasize that the ERP system remains active 24/7 along with the organization's activities. Short and critical stops should be negotiated previously with all areas of the organization. For example, if there is a need to stop a module upgrade of the electronic invoice for Brazil, internal consultants negotiate with all functional areas involved. Then document emissions are advanced or delayed.

The relations between the stakeholders take place during project development. These are constituted through exchanges of the artifacts generated in the process model stages that manage this development. The relationship may also consist of casual communications that occur informally but impacts on projects.

Verification of the frequency of major changes made in the year 2017 was carried out. We considered 8 changes in corporate policies, 5 changes in laws, 2 changes in technology and 2 changes motivated international standards.

Based on the data collection, it was possible to structure a first version of the distributed software evolution process and how the stakeholders' coordination of this scenario occurs. This process was described below:

## 4.2 Identify the changes

The demands for changes in ERP system can arise from multiple internal and external sources. In the context of this study, the main findings were highlighted.

*4.2.1 Changes from Law.* The government (national, state, municipal) may issue, through its various agencies and regulatory agencies, various types of changes, which may include changes in law, and may have a direct impact on the organization. One of the critical factors in the changes in the law is in the deadlines proposed by the government.

*4.2.2 Changes from Technology.* Technology changes are a consequence of changes in the industry context, often triggered by technological innovations such as innovations in product traceability, manufacturing automating or even server technology,

programming languages and database technology. These reflect directly the adaptation of the ERP system to suit the new technological needs.

*4.2.3 Changes from International Standards.* As a result of global market demands and certifications with issues such as quality, safety and the environment. International agencies develop global recognized norms and standards. The ERP system customer company is also subject to the certifications required by current and potential customers.

*4.2.4 Changes from Company Policies.* As a result of the changes in the corporate organization's policy and working mode, the ERP system often needs adapt to new processes of production, logistics, financial, accounting, and so on.

In each of these situations, both the changes and the solutions can come from multiple and distributed sources. A change in law can trigger a need that can demand both the internal consultants of the customer company, who identify the changes, or external partners and software producer in the development and implementation of the evolution.

We identified that the process path to be followed for the distributed evolution of the ERP system depends initially on the type of change (municipal, state, national or global level) and what stakeholders will be involved.

## 4.3 Identify the stakeholders

To identify the distributed evolution process of a proprietary corporate system it is necessary to initially identify the process stakeholders. Through literature, interviews and documentation, the stakeholders of the Table 2 were identified.

After identifying the stakeholders, we identified the types of changes that influence the evolution of the ERP system.

## 4.4 Identify the process

It is important to highlight that the distributed evolution process can be carried out in different ways, because it depends precisely on the type of evolution that the ERP system will suffer. Evolution at the International Standards level may directly involve the Software Producer and External Partners. And, while evolution at the Government municipal level may involve only External Partners, as the type of solution may not require direct interference from the Software Producer.

The description of the identified process was based on the perspective of the customer company of the ERP system, based on direct observation, interviews, project documents and e-mails between stakeholders. The stages were divided according to the events occurred and monitored at each moment of the process. The nomenclature of the steps was based on different events raised during data analysis as presented below:

*4.4.1 Step 1. Change.* Triggered by the need for evolution of the system, motivated by any of the changes previously identified in the subsection 4.2. At this stage there may be participation and interaction of one or more stakeholders. Each type of change will lead to a different path of process. For example, if an International Agency, hired by Customer Company, requests a change in the International Standards, the path of the evolution process may be carried out involving the Software Producer and may also involve the External Partners in the specification, development and implementation of change at Customer Company. In another case, if the Government at municipal level issues new law, it may be that only External Partners and Customer Company are actively involved in the specification, development and deployment of the

**Table 2: Identified stakeholders**

| Stakeholder | Description |
|---|---|
| Technology Providers | global marketplace that acts in technological evolution, providing new software and hardware solutions to improve the current context of technology |
| International Agencies | works in the elaboration of global norms and methodologies of structured operations that help organizations in the areas of quality, safety and environment, like the International Organization for Standardization (ISO). Due to the global evolution of security, control, and quality standards, for example, software should also evolve |
| Government | composed of national agencies that work in the elaboration of norms, laws and national taxes. Organizations are subject to the evolution of their software in response to government demands at national, state and municipal level |
| Software Producer | acts in the development and sale of the ERP, providing improvements pressured by the technology or customizations demanded by the other stakeholders |
| Local Subsidiary | Local software producer subsidiary that acts in the local representation of the ERP system in the country of ERP system customer company |
| External Partners | work mainly in the development of solutions and development of ERP projects, being responsible for assisting the ERP system customer company |
| Customer Company | ERP system customer, composed of departments, customers, suppliers. It also consists of employees that can be: end users, internal consultants, key users, support analysts or internal programmers who work directly with the ERP system |

solution. As result, process will go a different path. A third case could result in a different path, and so on.

*4.4.2 Step 2. Diagnostic.* Triggered by the need of stakeholders who demand requirements to adapt to the changes proposed by the context in which they are inserted. At this stage there may be participation and interaction of one or more stakeholders. At this stage, stakeholders who implement requirements keep abreast of the changes that can be triggered by the context in which stakeholders who demand requirements are inserted. The path in this step can also be executed in different ways. For example, a change in technology triggered by the Technology Providers may directly involve Software Producer, who identifies the change and acts on the evolution of the corporate system. In another case, the path of a change in Company Policies is far less complex than the path of a change in technology and may only involve External Partners and Customer Company in the diagnosis, specification, development, and enhancement of change. It is very important to highlight that paths of different projects can be executed simultaneously in the same environment, and the stakeholders must manage the interference between paths.

*4.4.3 Step 3: Specification.* From there, the Specification of the Change is carried out. At this stage there may be participation and interaction of one or more stakeholders. For example, once the Local Subsidiary identifies a change in the context in the Diagnostics step, it performs the change Specification. Then the Software Producer performs the development change and makes the packages available to the External Partners make the necessary changes to the solution. Finally, they provide the solution to the Customer Company. In another case, Customer Company issues a need to change Company Policies. After diagnosis, it is realized that the change can be implemented by the Customer Company itself. Therefore, the specification, development and deployment are the responsibility of Customer Company.

*4.4.4 Step 4: Development.* After making the Specification available, it is up to the stakeholders involved in the Development step the development of changes. At this stage there may be participation and interaction of one or more stakeholders. For example, a specified change may involve development by External Partners, but Customer Company actively participates in testing for validation in a testing environment. In another case, it may involve the development of primary packages by the Software Producer, and the development of secondary packages by External Partners.

*4.4.5 Step 5: Third Party.* In parts of the cases identified, in the Development step the Software Producer develops and makes available packages which directly result in the creation of new solutions developed by External Partners, and they must adapt these solutions to these new packages. Therefore, we also identify this intermediary step.

*4.4.6 Step 6: Deployment.* It consists of the Deployment of the product of the distributed evolution of software, as result of the artifacts produced by the stakeholders involved in the Diagnostic, Specification, Development and, where it occurs, Third Party steps. At this stage there may be participation and interaction of one or more stakeholders. For example, in a path managed by Customer Company itself, there may be no External Partner involvement at the Deployment step. In another case, other multiple paths may involve directly External Partners in the Deployment of changes in Customer Company.

Figure 1 shows the process design described in the subsection 4.4, confirming all identified aspects: the distributed software evolution process can occur through multiple different paths depending on the type of demand and the type of solution. Through Figure 1 it is possible to generically visualize the possible paths of the relationship between stakeholders in the process of distributed evolution. Stakeholders may be physically or temporarily distant and may or may not participate in the same step. The arrows represent the stakeholders who will participate in each step of the process. Therefore, it is possible that different stakeholders may or may not participate in the same Diagnosis, Specification, Development or Deployment. Figure 2 presents one example of the multiple possible paths that can be executed through a change in the national law.

## 4.5 Discussion

Through the structured distributed evolution process and data analysis, it was possible to observe some characteristics, difficulties and challenges. We discuss evidence obtained from historical data, considering ERP team interviews, projects registries and e-mail exchanges between stakeholders.

During the execution of the paths, many problems occurred in execution time, despite all diagnosis and specification. It is perceived that at all steps of the path, involving multiple stakeholders also impact the process. If different changes are being implemented in the same period by different stakeholders, it is necessary to evaluate the impact and interferences of one change over another.

> *Deploying projects executed by different external partners during the same period has its critical aspects. First, they work in the same testing environment. Second, external partners always blame each other for certain changes, which usually delays both projects in progress Both projects have their priorities and deadlines.*
>
> ERP team member

Implementing customizations from multiple software sources in a corporate organization is an activity that can move the entire company, because it can affect directly or indirectly the processes currently running. For example, making a new version available in the electronic invoice for Brazil may involve several areas of ERP performance, such as Production, Financial, Accounting, Purchasing, Sales. The end users of Customer Company are also key stakeholders impacted by the evolution of the software, since the new features affect their work routines.

> *The projects move the entire organization. A change in the production area can impact the sales, financial, accounting and logistics processes, for example. These users are key to the scenarios validation proposed by the change.*
>
> ERP team member

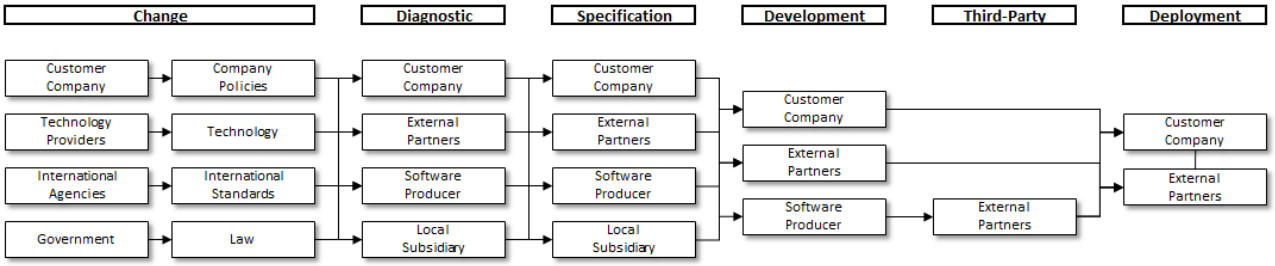


**Figure 1: Generic process of distributed software evolution in a proprietary enterprise system.**

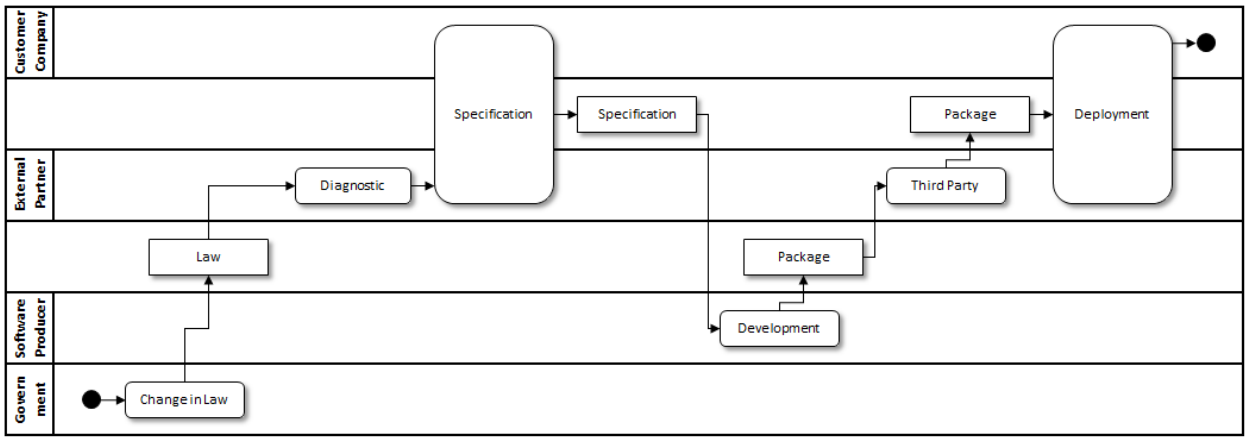


**Figure 2: Process of distributed software evolution identified in a change from national law.**

Through the project contracts analysis, it was possible to identify that each stakeholder carries out the impact analysis of evolution from its perspective. The analyzes are not as transparent and accessible to other stakeholders in the hierarchy. The lower the stakeholder hierarchy, the less transparent and accessible is the impact analysis. For example, although Software Producer and External Partners affirm that there is an impact analysis on the projects specified and developed by them, these stakeholders only inform Customer Company that they "guarantee" the operation of the system transactions and standard functions impacted by the project, disregarding previously customized transactions and functions.

> *The ERP system is so large and complex that any change can directly impact a process from another module. One of the critical situations is considered in projects involving external partners and software producer. They do not consider an impact analysis on functionalities not developed by the software producer.*
>
> ERP team member

During the case study two important events occurred, considered the main problems identified until then in the process of distributed software evolution. The first occurred when External Partners were unable to identify all requirements of the stakeholder that demanded requirements, even making a diagnosis and specification of the change. This situation delayed the project execution and the entire remaining project schedule, is necessary to return from the deployment stage to the stage of specification They realized that they needed Software Producer involvement in the process.

> *The more globally the stakeholder demands requirements, the more difficult it is to validate all requirements and business rules. This happens because the impact of change involves multiple customers and the stakeholder that demands requirements does not seem to feel compelled to detail all the scenarios of multiple customers.*
>
> ERP team member

In another case, an International Agency hired by the Customer Company did not warn them of a change. In this way, it was discovered that Customer Company was not meeting the new standards imposed by the Government. The situation was discovered when the Government initiated a process of validation of the solution offered by the International Agency. As result, the sales stopped for one day.

> *We had a recent situation with an International Agency that changed the standard for a new validation feature of the barcode in partnership with the Government at national level. The result was that the company's sales stopped for a day until we solve the problem.*
>
> ERP team member

It is important to note that the separation of environments (test and production) does not solve the problems that exist today. The stakeholders involved in the changes do not always validate the numerous scenarios, or do not know how far a change will reach a certain scenario/user or not. Nothing prevents Customer Company from detecting through end-users that the deployment still needs some adjustments and fixes in the Deployment step. However, from this point, it can be considered that the problem can be solved through the system maintenance process for adjustments and correction of errors.

> *Although we have the test environment to validate the project before deploying to the production environment, it is practically impossible to validate all the scenarios. For example, the validation of the electronic invoice for Brazil has numerous possible refusal codes and descriptions, one for each type of situation and each type of business.*
>
> ERP team member

Stakeholders should keep abreast of new stuff in the global economic, political and technological market. Customer Company should be aware of the budget and resources needed to keep pace with the evolution of the software, and External Partners must be aware of the changes proposed so that they have time to make adaptations of their solutions or develop new solutions based on these evolutions.

> *It is very important that all stakeholders are in sync with the market news. It is also important for stakeholders to be more transparent with each other, so that customers are also aware of the changes impact made by stakeholders that demand requirements.*
>
> ERP team member

The case study also allows us to conclude that, although it seems to be a cascade development model, the same path can return from the Development stage to the Specification stage, for example. For example, the Government sets a deadline for publication of a new Law in the Change stage. In the Diagnosis and Specification stage, one or more stakeholders involved in the process of system evolution design the specification so that other stakeholders can implement the new requirements in the Development stage. However, in the middle of the Development phase, before the deadline for publication of the new Law, the Diagnosis step is triggered again, motivated by a new Government requirement, and the path should return in the Diagnostic and Specification stages. So, that changes are also taken into account in the evolution of the system. Therefore, it is verified that the process can be considered iterative and incremental, because it refers to a possibility of repetitive execution of the same until the purpose is reached.

Analyzing the case study, there's a need to model distributed evolution in a way that helps the stakeholders in the process of distributed evolution of proprietary corporate systems, since multiple stakeholders may be involved with different projects in the same environment. This type of case study opens a discussion for the enhancement of the proposed generic model.

### 4.6 Threats to validity

As in any study, our study has limitations. First, during the case study, we had access only to the ERP system's customer company because we could not get directly to external partners and software producer.

Although the customer company's internal consultants have, on average, more than 15 years of experience, this was the only business segment they worked on until then. However, this limitation is reduced because the historical information exchange with other companies is large. According to the members of the team, the customer company has the custome to always exchange information with other companies, aiming to solve common problems.

### 4.7 Limitations

This study focused on just one organization for sample selection. Future studies may want to use samples from many organizations to generalize the results.

Our study also used an unbalanced sample of stakeholders. The future study may want to use a balanced sample to eliminate the undue influence of a given group of respondents on the results.

The case study is aimed at identifying the process of distributed evolution, not involving a deep analysis of the source code and artifacts involved.

## 5 RELATED WORK

To the best of our knowledge, there is no work that addresses distributed software evolution cases. The more closely related works are discussed in this section.

Studies evaluate that development and evolution can be spread across multiple teams to leverage different knowledge, experience, or capabilities. In this way, breaking a task into smaller and more manageable pieces is often an effective way to deal with the kind of complexity. The artifacts developed or modified separately need to be assembled as efficiently as possible into a consistent whole in which the artefact still function as specified [3].

Previous studies on software customization processes have analyzed the phenomenon from different perspectives. For example, some authors have analyzed the implementation processes of an ERP system from the customer companies perspective [4]. However, this study examines aspects of the company's context change about the users' readiness for ERP system implementation.

Others compared maintenance process with existing standards (IEEE / 12207 maintenance and process standard) [17]. Despite bringing relevant information about the application of existing standards in maintenance, the study deals with a direct relationship between a customer company maintenance need and software

producer packages, without involving multiple external partners. Therefore, the study does not represent the context of this work.

Caporarello and Viachka [4] evaluate that ERP implementation is an important organizational event because it greatly affects the way people work and can make previous knowledge obsolete. In their study, they assess the end-user's readiness for change. Although the article focuses on the implementation phase, the addition of new features can have a similar effect for users during evolution.

Seifermann et al. [30] describe the main challenges in maintaining security properties during software evolution. These main challenges highlight an important challenge, which refers to the fact that there are multiple stakeholders interacting with the system. However, the challenges are only related to architecture security where the ERP system is allocated.

Oseni et al. [18], through an empirical evaluation, present a literature review of ERP systems post implementation modifications to identify the current state of the art, the main issues related to post ERP implementation changes, inherent gaps in the literature on post implementation changes and specific areas that require further research. The directions of research point to aspects that involve organizational learning post implementation.

Pagano and Brügge [20] present an empirical case study that explores the current user involvement practice during software evolution. We found that user feedback also contains: important information for developers, helps improve software quality, and identifies missing resources. This study highlights the end users' involvement in all stages of the project.

Salameh et al. [28] present a systematic review on visualization technique, which of stakeholders better understand on how software evolves and which parts of the software are most affected by a change. However, the proposed approach involves only the relationship between the members of a team of developers and source code. This technique can be very interesting for the access of the programmers of the Company Company to the changes made by third parties in the ERP system environment.

Parthasarathy and Sharma [23] analyze the impact of the ERP system customization on ERP system quality. This study establishes a relationship between the customization carried out by software producers and the resulting quality of ERP. The lack of prior empirical findings in this area lends additional importance to such a research effort.

Parhizkar and Comuzzi [21] state that post implementation customizations are likely to decrease the quality of the ERP system and the data they use, which affects negatively organizational performance. The study proposes a structure for analyzing the impact of post implementation changes of the ERP system, through a structure of dependency mapping between the system components and, based on these dependencies, automatically assesses the impact of a change through a new set of impact metrics. This study is the closest contribution that we find in the type of context that this work finds.

The integration of people is an inevitable result of process integration. The more processes are integrated, the more operations are carried out in a cross-cutting way and the greater the need for interactions and cooperation across the value chain [7] [4].

In the literature there are no directional studies focused on the distributed software evolution of proprietary enterprise systems. Therefore, we adopted the study of the distributed evolution of software from the perspective of a customer company to contribute to a better understanding of the process and stakeholders' involvement in this environment.

## 6 CONCLUSIONS

The case study also allows us to conclude that our study aimed to contribute to the identification of the process for distributed software evolution of a proprietary corporate system that enables stakeholders to adapt a particular local requirement to their global context through demands that come from diverse origins. However, the study was exploratory in nature and was not intended to provide a complete definition of distributed software evolution, but rather to lay the groundwork for future research on the concept.

First, our results suggest that the path directly depends on the origin of evolution and that stakeholders are affected by the global context changes. However, the literature does not present solutions that involve this distributed evolution. The most studies present only the implementation or ERP system maintenances not directly linked to the context changes.

ERP systems have been of interest over the last two decades and the number of publications has increased dramatically in the last 10-17 years. However, the theme of post-implementation changes is still under development. It is our recommendation that scholars pay attention to as the era of new ERP implementations is gone and changes are essential for software evolution.

All these considerations lead to the conclusion that stakeholders are essential elements in the distributed evolution strategies of an ERP system. Consequently, the process must in a way that all stakeholders can be involved in different types of change over the same period. And this process can not compromise the business of the customer company.

### 6.1 Future Research

Based on the perspective that the distributed evolution process is not definitively conceptualized in the literature found, although models have been found that contribute to the evolution of the software, most cases approach the software maintenance process, not involving the set of perspectives, as:

- the constant changes that occur in the scenario in which the system is inserted;
- the large number of stakeholders that may be involved in the process;
- the distributed way of development that this scenario presents;
- the parallelism of different projects to meet different demands;
- controlling and monitoring of constant changes on the system;
- data security involved in a multiple stakeholders' context;

As part of an effort to improve the software process, of which this study is part, it is planned to enhance the proposed model of distributed software evolution process. It is important that more case studies are conducted to improve the initial path and identify new stakeholders.

## ACKNOWLEDGMENTS

The authors thank the Araucaria Foundation for the support of scientific and technological development given to carry out this study.

## REFERENCES

[1] Richard Addo-Tenkorang and Petri Helo. 2011. Enterprise Resource Planning (ERP): A Review Literature Report. (01 2011). https://doi.org/10.13140/2.1.3254.7844

[2] H. P. Breivold, M. A. Chauhan, and M. A. Babar. 2010. A Systematic Review of Studies of Open Source Software Evolution. In *2010 Asia Pacific Software Engineering Conference*. 356–365. https://doi.org/10.1109/APSEC.2010.48

[3] Petra Brosch, Gerti Kappel, Philip Langer, Martina Seidl, Konrad Wieland, and Manuel Wimmer. 2012. *An Introduction to Model Versioning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 336–398. https://doi.org/10.1007/978-3-642-30982-3_10

[4] Leonardo Caporarello and Assia Viachka. 2010. *Individual Readiness for Change in the Context of Enterprise Resource Planning System Implementation*. Physica-Verlag HD, Heidelberg, 89–96. https://doi.org/10.1007/978-3-7908-2404-9_11

[5] Fabio Q. B. da Silva, Catarina Costa, A. Cesar C. Franca, and Rafael Prikladinicki. 2010. Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review. In *Proceedings of the 2010 5th IEEE International Conference on Global Software Engineering (ICGSE '10)*. IEEE Computer Society, Washingtoon, DC, USA, 87–96. https://doi.org/10.1109/ICGSE.2010.18

[6] Maya Daneva and Roel Wieringa. 2008. Cost Estimation for Cross-organizational ERP Projects: Research Perspectives. *Software Quality Journal* 16, 3 (Sept. 2008), 459–481. https://doi.org/10.1007/s11219-008-9045-8

[7] Redouane El Amrani, Frantz Rowe, and Bénédicte Geffroy-Maronnat. 2006. The effects of enterprise resource planning implementation strategy on cross-functionality. *Information Systems Journal* 16, 1 (2006), 79–104. https://doi.org/10.1111/j.1365-2575.2006.00206.x

[8] M. W. Godfrey and D. M. German. 2008. The past, present, and future of software evolution. In *2008 Frontiers of Software Maintenance*. 129–138. https://doi.org/10.1109/FOSM.2008.4659256

[9] Young Mok Ha and Hyung Jun Ahn. 2014. Factors Affecting the Performance of Enterprise Resource Planning (ERP) Systems in the Post-implementation Stage. *Behav. Inf. Technol.* 33, 10 (Oct. 2014), 1065–1081. https://doi.org/10.1080/0144929X.2013.799229

[10] Israel Herraiz, Daniel Rodriguez, Gregorio Robles, and Jesus M. Gonzalez-Barahona. 2013. The Evolution of the Laws of Software Evolution: A Discussion Based on a Systematic Literature Review. *ACM Comput. Surv.* 46, 2, Article 28 (Dec. 2013), 28 pages. https://doi.org/10.1145/2543581.2543595

[11] Björn Johansson and Frantisek Sudzina. 2008. ERP systems and open source: An initial review and some implications for SMEs. 21 (10 2008), 649–658.

[12] C. Lopez and J. L. Salmeron. 2011. A framework for classifying risks in ERP maintenance projects. In *Proceedings of the International Conference on e-Business*. 1–4.

[13] Tom Mens. 2008. *Introduction and Roadmap: History and Challenges of Software Evolution*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–11. https://doi.org/10.1007/978-3-540-76440-3_1

[14] T. Mens, Y. G. Guehéneuc, J. Fernández-Ramil, and M. D'Hondt. 2010. Guest Editors' Introduction: Software Evolution. *IEEE Software* 27, 4 (July 2010), 22–25. https://doi.org/10.1109/MS.2010.100

[15] Pierre-Alain Millet. 2013. Toward a Model-driven, Alignment-oriented ERP Methodology. *Comput. Ind.* 64, 4 (May 2013), 402–411. https://doi.org/10.1016/j.compind.2013.01.004

[16] N. K. Ömüral and O. Demirörs. 2017. Effort Estimation for ERP Projects: A Systematic Review. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 96–103. https://doi.org/10.1109/SEAA.2017.68

[17] Celeste See-Pui Ng and Guy G Gable. 2010. Maintaining ERP packaged software – A revelatory case study. *Journal of Information Technology* 25, 1 (01 Mar 2010), 65–90. https://doi.org/10.1057/jit.2009.8

[18] Taiwo Oseni, Susan Foster, Rahim Mahbubur, and Stephen Smith. 2017. A Framework for ERP Post-Implementation Amendments: A Literature Analysis. *Australasian Journal of Information Systems* 21, 0 (2017). https://doi.org/10.3127/ajis.v21i0.1268

[19] Taiwo Oseni, Mahbubur Md Rahim, Stephen P Smith, and Susan Foster. 2014. An initial empirical evaluation of the influence of erp post-implementation modifications on business process optimisation. (2014).

[20] Dennis Pagano and Bernd Brügge. 2013. User Involvement in Software Evolution Practice: A Case Study. In *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13)*. IEEE Press, Piscataway, NJ, USA, 953–962. http://dl.acm.org/citation.cfm?id=2486788.2486920

[21] Minou Parhizkar and Marco Comuzzi. 2017. Impact analysis of ERP post-implementation modifications: Design, tool support and evaluation. *Computers in Industry* 84 (2017), 25 – 38. https://doi.org/10.1016/j.compind.2016.11.003

[22] K. Park and A. Kusiak *. 2005. Enterprise resource planning (ERP) operations support system for maintaining process integration. *International Journal of Production Research* 43, 19 (2005), 3959–3982. https://doi.org/10.1080/00207540500140799 arXiv:https://doi.org/10.1080/00207540500140799

[23] Sudhaman Parthasarathy and Srinarayan Sharma. 2017. Impact of Customization over Software Quality in ERP Projects: An Empirical Study. *Software Quality Journal* 25, 2 (June 2017), 581–598. https://doi.org/10.1007/s11219-016-9314-x

[24] Rafael Prikladnicki and Jorge Luis Nicolas Audy. 2010. Process Models in the Practice of Distributed Software Development: A Systematic Review of the Literature. *Inf. Softw. Technol.* 52, 8 (Aug. 2010), 779–791. https://doi.org/10.1016/j.infsof.2010.03.009

[25] Václav Rajlich. 2014. Software Evolution and Maintenance. In *Proceedings of the on Future of Software Engineering (FOSE 2014)*. ACM, New York, NY, USA, 133–144. https://doi.org/10.1145/2593882.2593893

[26] Jiwat Ram, David Corkindale, and Ming-Lu Wu. 2013. Implementation critical success factors (CSFs) for ERP: Do they contribute to implementation success and post-implementation performance? *International Journal of Production Economics* 144, 1 (2013), 157 – 174. https://doi.org/10.1016/j.ijpe.2013.01.032

[27] M. A. Rothenberger and M. Srite. 2009. An Investigation of Customization in ERP System Implementations. *IEEE Transactions on Engineering Management* 56, 4 (Nov 2009), 663–676. https://doi.org/10.1109/TEM.2009.2028319

[28] H. B. Salameh, A. Ahmad, and A. Aljammal. 2016. Software evolution visualization techniques and methods - a systematic review. In *2016 7th International Conference on Computer Science and Information Technology (CSIT)*. 1–6. https://doi.org/10.1109/CSIT.2016.7549475

[29] Premaratne Samaranayake. 2009. Business process integration, automation, and optimization in ERP: Integrated approach using enhanced process models. *Business Process Management Journal* 15, 4 (2009), 504–526. https://doi.org/10.1108/14637150910975516 arXiv:https://doi.org/10.1108/14637150910975516

[30] Stephan Seifermann, Emre Taspolatoglu, Ralf H. Reussner, and Robert Heinrich. 2016. Challenges in Secure Software Evolution - The Role of Software Architecture. *Softwaretechnik-Trends* 36 (2016).

[31] Levi Shaul and Doron Tauber. 2013. Critical Success Factors in Enterprise Resource Planning Systems: Review of the Last Decade. *ACM Comput. Surv.* 45, 4, Article 55 (Aug. 2013), 39 pages. https://doi.org/10.1145/2501654.2501669

[32] Pnina Soffer, Boaz Golany, and Dov Dori. 2003. ERP modeling: a comprehensive approach. *Information Systems* 28, 6 (2003), 673 – 690. https://doi.org/10.1016/S0306-4379(02)00078-9

[33] Pnina Soffer, Boaz Golany, and Dov Dori. 2005. Aligning an ERP System with Enterprise Requirements: An Object-process Based Approach. *Comput. Ind.* 56, 6 (Aug. 2005), 639–662. https://doi.org/10.1016/j.compind.2005.03.002

[34] A. Talai and Z. E. Bouras. 2017. Software evolution based activity diagrams. In *2017 8th International Conference on Information Technology (ICIT)*. 82–88. https://doi.org/10.1109/ICITECH.2017.8079949

[35] M. Themistocleous, Z. Irani, R. M. O'Keefe, and R. Paul. 2001. ERP problems and application integration issues: an empirical survey. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. 10 pp.–. https://doi.org/10.1109/HICSS.2001.927240

[36] J. M. Verner, J. Sampson, V. Tosic, N. A. A. Bakar, and B. A. Kitchenham. 2009. Guidelines for industrially-based multiple case studies in software engineering. In *2009 Third International Conference on Research Challenges in Information Science*. 313–324. https://doi.org/10.1109/RCIS.2009.5089295

[37] J. Hermosillo Worley, K. A. Chatha, R. H. Weston, O. Aguirre, and B. Grabot. 2005. Implementation and Optimisation of ERP Systems: A Better Integration of Processes, Roles, Knowledge and User Competencies. *Comput. Ind.* 56, 6 (Aug. 2005), 620–638. https://doi.org/10.1016/j.compind.2005.03.006

[38] G. Xie, J. Chen, and I. Neamtiu. 2009. Towards a better understanding of software evolution: An empirical study on open source software. In *2009 IEEE International Conference on Software Maintenance*. 51–60. https://doi.org/10.1109/ICSM.2009.5306356

[39] Robert K Yin. 2013. *Case study research: Design and methods*. Sage publications.