Business-IT alignment anti-patterns: a thought from an empirical point of view

Jean-Philippe Gouigoux
Group CTO SALVIA Développement, Aubervilliers
France *jp.gouigoux@salviadeveloppement.com*
Dalila Tamzalit
Université de Nantes, CNRS, LS2N, F-44000 Nantes
France *Dalila.Tamzalit@univ-nantes.fr*

Abstract

This preliminary work aims to formalizes observed recurring bad business-IT alignment scenarios. This observation has been conducted subsequently to a 6-years empirical experience of audits of about thirty companies. It considers two research questions: 1) are there recuring BITA problems independently of the business domains? 2) how to formalize them? 14 BITA anti-patterns have been identified. A visual representation and an identity card are proposed to formalize them and illustrated on the 4 most encountered BITA anti-patterns. A first milestone is thus proposed towards a common base of BITA anti-patterns and open the discussion with BITA experts among researchers and practitioners, to pooling our efforts and identify research tracks. In fact, BITA is steel a crucial challenge for companies to have a good alignment between business and software. Moreover, handling misalignments is becoming much more sensitive for companies to move towards adoption of new digital capabilities in Digital Transformation challenges.

Keywords: Antipatterns, Business-IT Alignment, Information systems Governance, Digital Transformation, Reuse.

1 Introduction

While several companies and industries conduct their Digital Transformation (DT), misalignments and bad dependencies between business processes and their supporting software systems are discovered and become much more sensitive and even increase in highly-dynamic contexts of digital strategy. Most companies for now have been considering IT as an aside specialty and almost all of them have a dedicated direction or service for IT, with an average 72% of the budget being burnt in maintenance only [36], whereas investment in evolutions towards Business-IT alignment should represent the higher part of budget [31]. Business-IT alignment (BITA) represents a crucial issue for information systems. It is among the top concerns for Information Technology (IT) leaders [2]. By Business-IT alignment, we mean the dynamic ability of a business organization to use, in an efficient and coherent way, Information Technology (IT) to achieve business objectives and improve its competitiveness [20,21], [16], [35]. BITA aims at maximizing the value created by a company by aligning its digital strategy with its business strategy. As Information Technology (IT) is skimming competition among companies, it is becoming the strongest asset of companies and their main differentiator, especially in the context of Digital Transformation (DT) that attracts lot of enthusiasm this last couple of years [24], [17], [13], [26], [31], [37,38]. By considering the new trend of digitalization and regarding its complexity, several authors argue that it’s time to shift the paradigm for a better understanding on how to meet the challenges of IT in its several layers (like hardware, virtualization, management tools, automation tools, operating systems or software applications) in the issue of BITA [14], [31], [38, 39, 40]. According to Kahre et al. [38], 48% of the CIOs in US firms spend most of their time attempting to align their IT strategies with the overall organizational objectives. The scope of BITA is large. We do not address strategic alignment (e.g., how to use IT do obtain competitive advantages) but we explicitly focus on the operational nature of Business-IT alignment.

We focus on the software part of IT, namely the business applications and the alignment between the business functions layer and the supporting software layer. We leant on the results, based on a six-years industrial experience, of previous audits and advices of about thirty of IT systems in France. We formalize the observations we made by identifying recurring bad business-IT alignment scenarios. Our article is addressed to BITA experts, both researchers and practitioners, as a material to work on, to improve, to complete. It represents the first milestone towards a common base of BITA anti-patterns. We identified 14 BITA anti-patterns, but for reasons of space, we detail one anti-pattern, the most frequent in our experience, and we summarize 3 other anti-patterns that are also strongly present in the studied information systems.

The paper is organized as follows: after this introduction, Section 2 presents the exploratory research methodology, the context of the work and the two addressed research questions. Section 3 addresses issues and challenges of Business/IT alignment between the four considered layers of an information system. Section 4 presents the visual recognition and documentation, via the identity card, of BITA anti-patterns. Section 5 illustrate in a detailed way our approach through the most important anti-pattern and presents a discussion about the main causes, impacts and possible solutions to address it. Section 6 present a summary of three other important anti-patterns. Section 7 addresses related work, while Section 8 discuss the limitations of our work and opens future research directions before concluding in Section 9.

2 Exploratory research methodology

The starting point of this work came from a brainstorming about how to bring together academia and business in closer collaboration. Among different domains, the Business-IT alignment concern was quickly identified. In fact, BITA problems have not yet been solved. They are always weighing heavily on Information Systems of companies and industries, exacerbated with the Digital Transformation trend. We took a step back and analyzed different Information Systems previously audited by the practitioner author and we leant on the software architecture experience of the academic author to identify, aggregate and formalize our observations.

2.1 Context: the observed information systems

The practitioner author audited and advised 29 companies during 6 years, from 2014 to 2019 when he was the CTO of MGDIS[1], a French software development and vendor applications that targets public collectivities.

Table 1. Observed information systems.

20 Observed organizations (2014-2018)				
Index	Type	Year	# Users	Domain
1	Regional Council	2014	450	Complete map
2	Departmental Council	2014	200	Persons + Finance
3	Regional Council	2014	300	IAM + Persons
4	Regional Council	2014	500	Persons
5	Regional Council	2014	450	Persons + Finance
6	Regional Council	2014	300	Persons
7	Software Editor	2014	130	Business Intelligence
8	Equipment Industry	2014	8000	Complete map
9	Regional Council	2015	400	Persons
10	Regional Council	2015	650	Persons + EDM + IAM
11	Regional Council	2015	200	Complete map
12	Chamber of Commerce	2015	300	Professionnal training
13	Departmental Council	2016	80	Complete map
14	Regional Council	2016	800	IAM
15	Regional Council	2016	300	Complete map
16	Equipment Industry	2016	1200	IAM
17	Software Editor	2016	170	Planning + Persons
18	City	2017	120	Persons
19	Chemical Industry	2017	200	Finance + BI
20	Regional Council	2018	650	IAM + Persons

9 Observed organizations (2018-2019)				
Index	Type	Year	# Users	Domain
21	Agriculture	2018	1200	Complete map
22	Pollutants management	2018	900	Complete map
23	Lawyers	2018	200	IAM + Persons
24	News industry	2018	1500	Geographical
25	Agriculture	2019	4300	Persons
26	Food transformation	2019	1000	Data ontology
27	Software Editor	2019	800	Persons
28	Software Editor	2019	40	Persons + IAM
29	Government agency	2019	80	Data ontology

Though it is not possible to explicitly identify the advised companies for confidentiality reasons, the following list (**Table 1**) gives an overview of their characteristics. These organizations include public agencies, like ministries, regional and departmental councils

[1] http://www.mgdis.fr

mainly in France, as well as some large cities and some mid-sized private enterprises. From 2014 to 2018, organizations that have been advised for information system alignment projects were similar (**Table 1**– Left side). They were all public organizations with the almost same business domain and following same regulation. So, recurring alignment issues appeared to us to be just normal. In 2018, new different categories of companies (public as well private) started to be advised (**Table 1** – Right side). The same recurring alignments issues were noticed. The experience of the first phase (2014-2018) helped to clearly and quickly identify recurring misalignment scenarios in the second phase (2018-2019) and to answer them. By analyzing afterwards, in 2020, the results of the conducted audits of these about thirty different information systems, we identified a couple of same recurring problems, like bad smells: whatever was their number of users (from 40 to 8000), the technologies used and the business domains, all companies encountered a couple of similar misalignment scenarios.

2.2 Research questions

It was interesting to try to detect the cause/consequence connections for each recurring problem and potential similarities. The post-analysis conducted on all the information systems encountering a given problem established the presence of the same obvious, but wrong, solutions. We noticed that these solutions are commonly (re)used but are usually ineffective and risks being highly counterproductive [42]. We can formalize our objective through two main research questions:

1. RQ1: "When faced with recurring alignment problems in different information systems, is there a same common wrong solution? Are there independent of the business domain?"
2. RQ2: "What are their characteristics? How to formalize them, in order to classify them and populate a dedicated base?"

The use of the "*anti-pattern*" expression came from the GOF book [8]: an anti-pattern distinguishes itself "*from a bad habit, bad practice, or bad idea, by being a commonly-used process, structure of pattern of action that, despite initially appearing to be an appropriate and effective response to a problem, has more bad consequences than good ones*". This definition, coupled to our observations and our research questions lead us to use the expression "*Business-IT Alignment anti-patterns*" or, for short, "*BITA anti-pattern*". We identified 14 BITA anti-patterns in the considered information systems, and behind each of them, we identified the same recurring bad scenario. We propose to formalize the cause/consequence connections and suggest an identity card to document identified BITA anti-patterns. Before explaining our findings, we first present issues and challenges of BITA.

3 Issues and challenges of Business-IT alignment in information systems

In the context of BITA's challenges [1], [35], [21], the main encountered issue is inappropriate and bad coupling between business and IT, mainly:

1. When modifications are directed by IT changes, several side effects are introduced (e.g., prohibitive costs, high failure risks or strong coupling).
2. When the business strategy changes, the underlying changes at the IT layer are hindered because the business is often, especially in legacy systems, embodied in the IT.

It is of importance that the business strategy should not be guided by IT changes: *it is of importance to have loose* coupling *between business and its underlying IT solutions* to cope more easily with changes. To position our claims, we consider the following layers of an IS:

3.1 Business & IT: the 4 layers of an information system and their relationships

We lean on a four-layer diagram, as often seen in Enterprise Architecture [34], [29], [5], [18], to articulate issues and pitfalls of Business/IT alignment (BITA). The considered framework is the four-layer representation of the CIGREF, a network of major digital French companies and public administrations. This representation (**Fig. 1**) is widely accepted in by all French CIOs. This diagram is used to represent information systems by separating their content in four layers, each relying on the following one:

- **Layer 1– Business processes:** this layer presents the value definition of the enterprise, through chained activities to perform the business, like the Business Process Models. It is the layer where the digital transformation happens.

- **Layer 2– Business functions**: is the catalogue of all business functions, generally classified in five domains: Business, Master Data Management, Collaborative, Common tools and Business Intelligence.

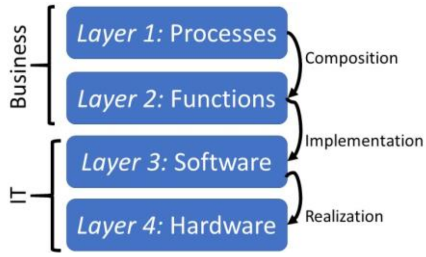


Fig. 1. Four-layers diagram of an information system.

- **Layer 3– Software**: represents the applications owned or used by the company. Not all business functions have corresponding software, as some operations may be entirely manual.
- **Layer 4– Hardware**: namely hardware and infrastructure supporting the software.

The relationships between the layers have different semantics: Layer 2 exposes business functions used for the *composition* of business process of Layer 1. Layer 3 is an *implementation* of function of Layer 2 and Layer 4 is the *realization* of the Layer 3.

3.2 BITA issues on the four layers

Lack of alignment between the two business layers (layers 1 and 2) or between the two technical layers (layers 3 and 4) is not a critical issue. On layers 1 and 2, only the business is concerned. Adjustment can be made on processes or functions without generally involving the IT department. Conversely, adjusting the software (Layer 3) and hardware (Layer 4) to each other is a pure IT problem and the business is not supposed to be concerned about how many machines are used or which version of the OS is exploited, as long as business functions are correctly exposed. The Achilles heel of Business-IT alignment is at the interface between Business and IT, so between Layers 2 and 3: daily-concerns of functional and IT people are often non-convergent and their background and skills avec different while both have to work together towards the same global business objective of the company.

3.3 BITA challenges

A utopic IT system would have a perfect alignment with the business it supports, thus, between layers 2 and 3. Each business function (element of layer 2) is realized by one, and only one, software component (element of layer 3), as illustrated in Fig. 2. Of course, this one-to-one relation is not about number of runtime application processes: there may be as many as needed provided since they are all based on the same component/version. A more realistic example of a good alignment is illustrated in Fig**.** 3. It is a picture of a real company's organization (details are omitted for confidentiality reasons). The links between layers are of interest since they show the underlying degree of coupling and dependencies.

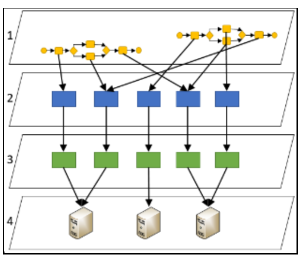


Fig. 2. Ideal alignment for simplistic and elementary information systems.

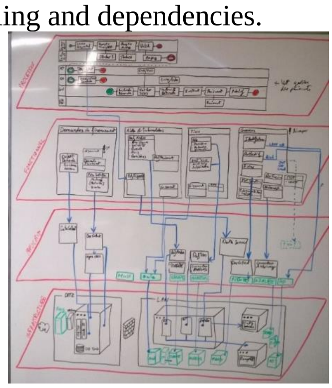


Fig. 3. Correct alignment for a simple real-world information system.

The main point to notice in Fig**.** 3 is that a few business functions (layer 2) point towards the same instance of software (layer 3). In addition, there is no sharing of software across business domains (the highest-level grouping inside layer 2): the only sharing is between functions that are operated by the same organizational direction, or for software that implements the processes themselves (ERP, middleware or orchestration software for instance), which is acceptable as it does not convey unwanted coupling. As we can see in Fig. 2 and Fig**.** 3, graphical representation helps to propose and understand a global overview of a complex situation by focusing on points of interest. Graphical summaries have often been used to concisely explain a pattern (like for Object-Oriented Programming [8] or for Enterprise Integration Patterns [15]).

4 BITA anti-patterns between layers 2 and 3: recognition and documentation

We remind that we focus on BITA issues between layers 2 and 3 of an information system. Before presenting identified anti-patterns between these two layers, it is of interest to first highlight the alignment complexity between business functions (layer 2) and software (layer 3), especially during the integration:

4.1 Alignment complexity between layers 2 and 3

When a function (layer 2) is realized by a software artefact (layer 3), there is more than just align[2]ing the identifiers. A service means a name, a way to be called (including address, protocol and formats) and a contract. This contract itself being often much more than a technical one but including, for example, Service Level Agreement details or conditions of use. The contract, according to us, is the main mean of alignment between business and IT. Links between layer 2 and layer 3 can be bidirectional: items from layer 2 can be realized by item(s) of layer 3 and item(s) from layer 3 can depend on item(s) from layer 2. Let's illustrate, in **Fig. 4**., how these links had impacts on the information systems of the advised companies.

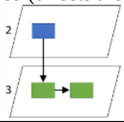
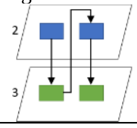
Technical coupling	Correct indirection
IT overused (directs the business)	Good balance through the use of contract interface.
	

Fig. 4. The two integration scenarios between layer 2 and layer 3

- *Bad situation: Technical coupling (left part of* Fig. 4*):* cooperation is done at the IT layer, between two pieces of software. This technical coupling is difficult and risky to evolve. An evolution in the software layer may break the equivalent functional dependency in Layer 2.
- *Good situation: Correct indirection (right part of* Fig. 4*):* integration of two functions is indirect: a function is realized by a software and this last implements the integration by pointing to the needed other function, which is in its turn realized by another piece of software. This indirection generates low coupling, and the caller can be changed without impact on the called piece of software and the implementation can be changed without any impact for the applications that call the function itself, since the contract protects them.

4.2 Identity card for the antipatterns

To characterize the bad BITA situations, we propose the *identity card* to document the BITA antipatterns. This identity card (**Fig. 5**) has different properties, spread out in three main parts:

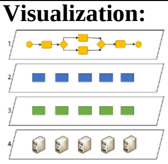
<table>
<tr><td rowspan="5">Visualization:
</td><td colspan="2">Short name</td></tr>
<tr><td rowspan="2">Description:</td><td>Effect on evolution:</td></tr>
<tr><td>Effect on usability:</td></tr>
<tr><td rowspan="2">Typical cause:</td><td>Ease of recovery:</td></tr>
<tr><td>Time to recover:</td></tr>
<tr><td colspan="3">Characterization:</td></tr>
<tr><td colspan="3">Consequences:</td></tr>
<tr><td colspan="3">Additional hints:</td></tr>
</table>

Fig. 5. Business-IT alignment antipatterns identity card.

1. A general presentation (*Visualization, Short name, Description and Typical cause*),
2. Four indicators for effects and recovery efforts (*Effect on evolution, Effect on usability, Ease of recovery, Time to recover*), with different possible values.
3. Additional information (*Characterization, Consequences, Additional hints*)

These properties (explained in **Table 2**) that have been identified from the industry standard on backup recovery, using Recovery Point Objective (how much is lost/has coupling problem) and Recovery Time Objective (how long does it take to go back to normal/to align the system).

Table 2: Explanations of the identity card of the BITA anti-patterns.

<table>
<tr><td rowspan="5">Visualization: quick visual identity to the antipattern</td><td colspan="2">Short name: the name should be consensual for aims of documentation and classification.</td></tr>
<tr><td rowspan="2">Description: explains the visual appearance (visualization) of the pattern, mainly the dependencies between the layers.</td><td>Effect on evolution:
- None: the antipattern does not hinder the system’s ability to evolve. Its impact is on other aspects (e.g., performance, ease of use.)
- Weak: business evolutions can be implemented, but not as easy as it could ideally be.
- Medium: evolution of the system is possible but requires some planning and analysis.
- Strong: the system is blocked or severely constrained by the antipattern.</td></tr>
<tr><td>Effect on usability:
- Non-noticeable: the antipattern has no impact on the business users’ day-to-day work.
- Low: the users perceive the problem in some situations, with limited impact.
- Medium: the situation has a significant impact on the business.
- High: the IT system is difficult to use; workarounds are common and/or errors arise in it that can have a measurable impact on the business.</td></tr>
<tr><td rowspan="2">Typical cause: a short description of the main reason why the antipattern may appear in an IT system</td><td>Ease of recovery:
- Easy: multi-step, well-documented methods exist to reduce and suppress the antipattern, with limited or virtually inexistent impact on the users.
- Average: the antipattern can be reduced or removed with low impact providing a careful analysis is realized beforehand.
- Complex: removing the antipattern from the system requires an important analysis effort, while still not being able to ensure that users will not be impacted.</td></tr>
<tr><td>Time to recover:
- Quick: the recovery method can be applied with expected results in days or weeks.
- Medium: the recovery method needs a few months / sprints to be functional.
- Long: the effort to recover will be measured in years (including stabilization of the business uses after detecting unexpected impacts).</td></tr>
<tr><td colspan="3">Characterization: a more thorough description of the antipattern.</td></tr>
<tr><td colspan="3">Consequences: lists the most known problems and their effect on the system.</td></tr>
<tr><td colspan="3">Additional hints: indicates symptoms that are often associated with the antipattern.</td></tr>
</table>

5 Illustration with the antipattern #1:

5.1 Description

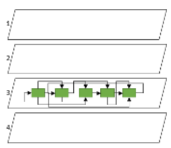
<table>
<tr><td rowspan="5">Visualization:
</td><td colspan="2">Short name: Pure technical integration</td></tr>
<tr><td rowspan="2">Description: many items in 3 depends on several items in 3</td><td>Effect on evolution: strong</td></tr>
<tr><td>Effect on usability: none</td></tr>
<tr><td rowspan="2">Typical cause: IT system organization led by technical needs instead of functional</td><td>Ease of recovery: complex</td></tr>
<tr><td>Time to recover: medium</td></tr>
<tr><td colspan="3">Characterization: software applications call each other directly or through the use of other software, without any level of indirection or contracting.</td></tr>
<tr><td colspan="3">Consequences: this causes high degree of coupling, since modifying a given application may have impact on many others, and thus many other functions or workflows depending on them. The system gets rigid and fragile to any evolution.</td></tr>
<tr><td colspan="3">Additional hints: coupling is typically caused by direct software calls to APIs, PL/SQL methods, COM functions, use of triggers. Middleware, when used inappropriately, also causes coupling, only displacing it in another software process.</td></tr>
</table>

Fig. 6. Pure technical integration anti-pattern

This antipattern “*Business evolution implemented through a completely technical integration*”, is the most important one in terms of frequency of occurrence within the audited systems, of impacts on the information system and its complexity of recovery. It is first described via its identity card then illustrated through examples before presenting its main causes, the difficulties it causes and possible solutions to address it.

5.2 Examples

A well-known example of this antipattern is when *Extract Transform Load Process* (ETL) or *Enterprise Application Integration* (EAI) is used to connect directly two databases. From far, the most-often seen example of this antipattern is, the use of an ETL to try and integrate the different functions that have to be coordinated. Of course, there is a need for a technical link somewhere but the right way to create it is definitely not to do so without any indirection level, like advised in the right part of **Fig. 4**. Direct links between two pieces of software in layer 3 (**Fig. 6**) results in dangerous technical coupling. Lot of cases have been observed by the authors at customers' sites where the ETL was directly connecting two databases without any indirection. Whereas this action is supposed to make the IT system better by exploiting the richness of interconnecting business, interconnecting only on the technical layer causes even greater problems. The link will fail for every non-compatible modification on either side of it. It is important to notice that this anti-pattern occurs in systems with poorly designed coupling.

5.3 Main causes

This antipattern may be the mother of all business/IT alignment ones we identified. Pure technical integration affects basically all studied IT systems. The problem typically comes from the fact that technicians have been trusted to implement interactions between business functions. However, how competent and willing they were, they created technical coupling, simply because they worked on their own layer of competence (layers 3 and 4).

5.4 Difficulties caused by the pattern

An ETL alone makes it for some technical coupling but as far as there is the associated amount of maintenance, the situation is acceptable. The real difficulties, which have been observed many times, happen when several ETL tasks are active in sequence (or even worse, without synchronization) while their chaining is not anymore controlled. It is the case for systems with poor IT implementation. On a particular occurrence, a departmental council synchronizing at night five different applications containing data about individuals and legal entities came to a point where duplications, conflicts and roundtrips made the whole process collapse. Complete recovery of the process could not be achieved and several months passed until the situation was cleared. This is the typical difficulty for banks for instance, of having a process that is completely embedded in the software and for which knowledge has been lost. In the end, the business rules are in COBOL and we can no longer rewrite them. Other examples have been observed where the complexity was not so high in the different information systems, but still caused problem.

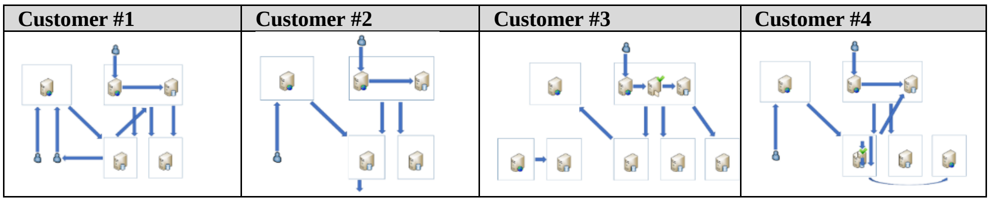


Fig. 7. simple customer's data streams

The diagram in Fig. 7 shows the GDPR-related streams for four different customers. Each of these sets of streams can be managed independently without too much maintenance effort, but the difficulty came from the fact that one given piece of software was supposed to accommodate all these situations. Superimposition of schemas is illustrated in **Fig. 8** (a).

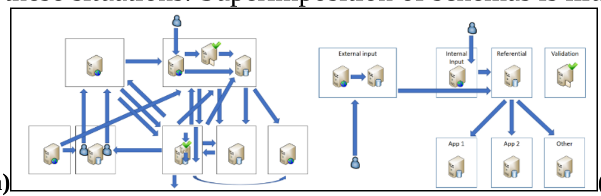
(a)  (b)

Fig. 8. (a) Real data streams for all the customers/ (b) Data streams with a referential

In such a network of data streams, it is particularly difficult to ensure robustness of the integration, non-occurrence of defaults and non-regression of features. **Fig. 8** (b) shows how the complete system that has been streamlined into well-aligned, contract-based, generic data exchanges, following a multi-year effort.

5.5 Possible solutions

The solution that has brought the best results at many times and is now routinely used by the authors is to add a level of indirection for the targeted business entity (the right part of **Fig. 4**). This generally leads to putting in place a service that acts as a referential, namely the place where the truth is in the system for a given entity. This service is both functional and software, in the spirit of a nice alignment. This approach gave the diagram (b) in **Fig. 8** of personal data streams after reorganization around a "person" referential.

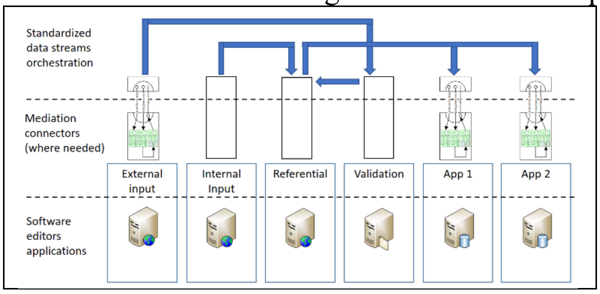


Fig. 9. Technical view of the reorganization.

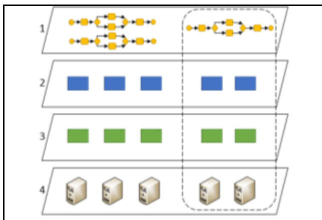
A more technical view of the solution is where the orchestration, mediation and proprietary levels have been clearly separated (**Fig. 9**). The stream of data shows a much better control and this design can be made GDPR-compliant. Thus, the right to be dereferenced can be implemented quite easily and without risk, contrary to the former architecture.

6 Three other antipatterns

In this section we summarize the 3 other most important antipatterns that occurred frequently in audited information systems and leave the 10 others for further work.

6.1 Antipattern #2: "The functional silo dedicated IT subsystem"

Description

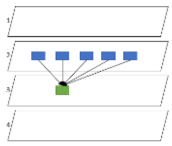
<table>
<tr><td rowspan="3"></td><td colspan="2">Dedicated silo</td></tr>
<tr><td>Description: related sets of items on layers 1, 2, 3 and 4 appear to be completely separated from the rest of the map</td><td>Effect on evolution: medium
Effect on usability: low</td></tr>
<tr><td>Typical cause: the use of a dedicated software suite has made difficult to interoperate with the rest of the Information System</td><td>Ease of recovery: easy
Time to recover: quick</td></tr>
<tr><td colspan="3">Characterization: a part of the IT system appears to be completely isolated from the rest. This may be the desired goal in some strong regulatory contexts, but this can also lead to duplicate business functions in other software.</td></tr>
<tr><td colspan="3">Consequences: the separation of software and hardware resources gives a false impression of autonomy of functions and processes. However, the mere fact that the silos belong to the same organization almost always carries a coupling, which one day will become active and cause unexpected coupling.</td></tr>
<tr><td colspan="3">Additional hints: a first sign of trouble may be that the whole appears as a silo, but the detailed map shows unassumed technical links between applications or databases. That means that the silo is still viewed as a separate system whereas, in practice, it has already started to be linked to the rest of the IT system, without these links being led – let alone known – from the business point of view.</td></tr>
</table>

Example

In a fusion or acquisition of a company by another, the capacity to quickly merge human resources, including salary, vacation, calendars, etc. is essential. The HRIS (Human Resource Information System), and more generally the HR domain, has a good maturity in Business/IT alignment, generally because the owners already tried to accommodate all functions inside a single piece of software. The antipattern happens when the HR thinks it can operate in IT system independently from the rest of the company's business domains, while employees represent entities highly used in other parts of the IT system: employees have accounts, they connect to the stock application or they are associated to customer relationships. It is mainly a lack of IT culture when HR think they are completely separated from the other parts of the IT system.

6.2 Antipattern #3: “Monolith application”

Description

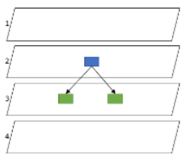
<table>
<tr><td rowspan="4">Visualization
</td><td colspan="2">Monolith</td></tr>
<tr><td rowspan="2">Description: many items in level 2 point towards a unique item in level 3</td><td>Effect on evolution: strong</td></tr>
<tr><td>Effect on usability: low</td></tr>
<tr><td>Typical cause: dependence on external software editors</td><td>Ease of recovery: easy
Time to recover: long</td></tr>
<tr><td colspan="3">Characterization: multiple business (unrelated) functions are implemented via the same software.</td></tr>
<tr><td colspan="3">Consequences: as the software gathers many functions, it has to change to follow different business requirements, triggering other changes. The system thus becomes more and more fragile.</td></tr>
<tr><td colspan="3">Additional hints: this antipattern is often accompanied with numerous technical dependencies to the concerned application. Only people with experience can make modifications to it.</td></tr>
</table>

Example

In one of the studied systems, two software applications, out of more than twenty, gather at least a third of the data streams, representing around half of the links between items in the level 3. After interview of users, we were able to connect these applications one-level up to at least ten functions in the level 2. Though it was not clearly visible at first, these applications thus constitute finally monoliths, as they assemble many different functional responsibilities. The main cause is generally because non-technical deciders consider that buying one software that covers many functions will cost less than buying a dedicated software for each function.

6.3 Antipattern #4: “Functional multiple implementations”

Description

<table>
<tr><td rowspan="4"></td><td colspan="2">Functional multiple implementations</td></tr>
<tr><td rowspan="2">Description: a single item in 2 points toward several entries in 3</td><td>Effect on evolution: strong</td></tr>
<tr><td>Effect on usability: medium</td></tr>
<tr><td>Typical cause: lack of comprehension of the functional and/or business model</td><td>Ease of recovery: average
Time to recover: medium</td></tr>
<tr><td colspan="3">Characterization: a single business function is implemented by two (or more) distinct pieces of software</td></tr>
<tr><td colspan="3">Consequences: the implementation of a given function differs depending on the context. Business evolutions are not necessarily ported in all the implementation and the behavior of the system becomes erratic.</td></tr>
<tr><td colspan="3">Additional hints: different users in different contexts observe unexpectedly different results from the same action. Business calculation like prices may differ depending on which application is consulted. Reporting has to be consolidated to find the truth in the system. Differences in functional or business behavior may also appear depending on the kind of device used to realize an action on the system.</td></tr>
</table>

Example

The business function colloquially called “reporting” used for electronic documents generation is used by several business functions. Dedicated applications tend to include it inside their own code, duplicating thus the software behind a single function. One of the studied IS had more than 40 different implementations. Most of the time, these applications are themselves monoliths, since they include this function as well as some others in addition to their main business responsibility. The main cause of this antipattern is the lack of norms and standards. Lack of a coherent map of the entire IT system or lack of communication between different owners also participate in this situation.

7 Related works

Surprisingly, most of research works on alignment between business and IT, and also Digital Transformation, are from the management field [24], [17], [25], [13], [9], [27] while digital strategy implies to embody IT challenges and changes at the strategic level of any company. To our knowledge, no work addressed formalization of antipatterns of Business-IT alignment by considering the four layers of an IT System. Most of antipattern existing works address a specific paradigm or context: few works addressed antipatterns to adopt SOA, which is the closest to the four-layers diagram we consider, like Brown&al [3] that gives a list of antipatterns in a textual format but no high-level overview of them. Other works are more tied to IT risk management [33] while some works address antipatterns in Process Modeling [19], [11], [32]. More recent works use alignment antipatterns in the context of ontologies [12], [28]. A couple of research works address alignment issues through the use of norms and pivotal formats using Domain Driven Design [6, 7], [14], [10]. In [41], authors precise that the developed conceptual model that detects business-IS strategies misalignment cannot be generalized.

8 Limitations and future work

Limitations: one main limitation of our work is that the analysis has been conducted afterward and based on observations conducted on the results of the audit of about thirty companies in their alignment concerns. We need to push forward our findings to ensure proper generalization and general implications, answer the lack of statistics and avoid drawing inference too quickly [43, 44]. In addition, we made a couple of assumptions: the work is mainly about software serving direct business requirements where modularity and separation of concerns are assumed to be done to a reasonable degree, which is not always the case. Other facets need to be addressed, e.g., the scale, the distributed nature, the roles and stakeholder diversity, the types of alignment required (functional, legal, human factors and socio-technical considerations) or security and privacy. The overlap between these facets and also the domino effect when implementing a change need to be considered.

Future work: several future research works are envisaged. From the practitioners' side, the first objective is to map out misalignment's scenarios of companies, to define a methodology to help detection of antipatterns and filling the identity card and, then, analyze and confront the results through the proposed anti-patterns. The second objective is to see if and how we can generalize our findings. Research works like Seddon & al [43], Wieringa & al [44] or R. Yin [45] represent a good starting point for this objective. Ideally, we would also like to start to populate a BITA anti-patterns basis to make it available. In addition, we would like to compare the proposed identity card with the twelve components of alignment of Luftman & al [46]. They represent the axes on which the misalignment can be felt, and it would allow to classify the antipatterns according to the impact they have on each axis, with a radar diagram for example. In the meantime, we identified a couple of academic research tracks, starting by the state of the art. We can mention, among them: (i) *for what aims BITA antipatterns can be used and how?* For instance, reverse engineering of an IT system on the 4 layers and its comparison to the antipatterns; (ii) *for what kind of systems?* We used them on existing/legacy systems but they can be used as guidelines to avoid bad design of new IT systems; (iii) *for what kind of evolution?* They can be used to address refactoring and improvement of IT systems. Moreover, it is utopic to think that analyzing an existing system will give a set of "perfect" antipatterns. It is more likely to have a set of partial antipatterns. One will need to deal with these partial results to evolve the system. In the context of complex systems, it is important to lean on good visualization means in order to detect antipatterns (like 3D representation or augmented reality). In any case, it is of importance to keep in mind the importance of ensuring continuous and qualitative BITA and by emphasizing on the links between the Business and the IT.

9 Conclusion

Although abstraction and best practices for designing applications are now widely taught and used, stepping back for complete and complex systems are not taught at school but are learnt from practice, generally after several years. They are in addition taught for IT people and not for managers. Managers as well IT employees need more formal frameworks and tools to help them to see more clearly the current and potential future technology landscapes. BITA issues fall under this category and encounter several challenges at different levels: strategy, business processes, business functions, software and the supporting infrastructure. We focused on the software side of BITA, between the business functions and there underlying software. We considered and analyzed the results of a 6-years' experience of audit and advice of about thirty public and private

French companies in their alignment concerns. We proposed the concept of *BITA antipattern*, in the spirit of software patterns and antipatterns widely adopted by companies [8], [3, 4], [23]. We also proposed an *identity card* to document antipatterns that can appear between the layers 2 and 3 of an information system. This identity card proposes a visualization of each antipattern, its attributes and descriptions with respect to effect on system evolution, effect on usability or ease of recovery. We detailed our approach on the most frequent and important antipattern we met and summarized 3 other common antipatterns. We have identified 10 additional anti-patterns left for future work. To conclude, we remind that this work is a construction-oriented research and is a retrospective rather than an in-situ exercise. It aims at proposing insights observed from industry experience and represents a first milestone for a further research objective to formalize, apply and enhance the methodology by applying it on companies, and why not of different countries, and it also aim at opening several research tracks (methodology, tools to assist the detection, misalignment management and bad dependencies between business processes and supporting software): there is definitely room for improvement in developing formalisms, approaches and tools for managers and IT deciders to help them to cope with IT changes from the strategy viewpoint.

References

1. Adomavicius, G., Bockstedt, J. C., Gupta, A., & Kauffman, R. J. (2008). Making sense of technology trends in the information technology landscape: A design science approach. Mis Quarterly, 779-809.
2. Braun, C., & Winter, R. (2007, March). Integration of IT service management into enterprise architecture. In Proceedings of the 2007 ACM symposium on Applied computing (pp. 1215-1219).
3. Brown, W. H., Malveau, R. C., McCormick, H. W., & Mowbray, T. J. (1998). AntiPatterns: refactoring software, architectures, and projects in crisis. JW & Sons, Inc.
4. Buschmann, F., Henney, K., & Schmidt, D. C. (2007). Past, present, and future trends in software patterns. IEEE software, 24(4).
5. Chorafas, D. N. (2016). EA and new generation information systems. CRC Press.
6. Evans, E. (2004). DDD: tackling complexity in the heart of software. Addison-Wesley.
7. Evans, E., & Szpoton, R. (2015). Domain-driven design. Helion.
8. Gamma, E. (1995). Design patterns: elements of reusable object-oriented software. Pearson Education India.
9. Gerow, J.E., Grover, V., Thatcher, J., Roth, P.L., 2014. Looking toward the future of IT-business strategic alignment through the past: a meta-analysis. MIS Quart. 1159–1186
10. Gouigoux, J. P.,& Tamzalit, D. (2017). From Monolith to Microservices: Lessons Learned on an Industrial Migration to a Web Oriented Architecture. In Software Architecture Workshops 2017 IEEE (pp. 62-65)
11. Gschwind, T., Koehler, J., & Wong, J. (2008, September). Applying patterns during business process modeling. In International Conference on Business Process Management (pp. 4-19). Springer, Berlin.
12. Guedes, A., Baião, F. A., & Revoredo, K. (2014). Digging Ontology Correspondence Antipatterns. WOP.
13. Hess, T., Matt, C., Benlian, A., & Wiesböck, F. (2016). Options for formulating a digital transformation strategy. MIS Quarterly Executive, 15(2).
14. Hinkelmann, K., Gerber, A., Karagiannis, D., Thoenssen, B., Van der Merwe, A., & Woitsch, R. (2016). A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. Computers in Industry, 79, 77-86.
15. Hohpe, G., & Woolf, B. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Professional.
16. Horlach, B., Drews, P., & Schirmer, I. (2016). Bimodal IT: Business-IT alignment in the age of digital transformation. Multikonferenz Wirtschaftsinformatik,1417-1428.
17. Kane, G. C., Palmer, D., Phillips, A. N., Kiron, D., Buckley, N. (2015). Strategy, not technology, drives digital transformation. MIT Sloan Management Univ.Press, 1-25.
18. Khoshafian, S. (2016). Service oriented enterprises. Auerbach Publications.
19. Koehler, J., & Vanhatalo, J. (2007). Process anti-patterns: How to avoid the common

traps of BPM. IBM WebSphere Developer Technical Journal, 10(2), 4.
20. Luftman, J. (2004). Assessing business-IT alignment maturity. Strategies for information technology governance, 4, 99.
21. Luftman, J., & Kempaiah, R. (2007). An Update on Business-IT Alignment:" A Line" Has Been Drawn. MIS Quarterly Executive, 6(3).
22. Luftman, J., Lyytinen, K., & Zvi, T. B. (2017). Enhancing the measurement of information technology (IT) business alignment and its influence on company performance. Journ. of Information Technology, 32, 26-46.
23. Martin, R. C. (2002). Agile software development: principles, patterns, and practices. Prentice Hall.
24. Matt, C., Hess, T., & Benlian, A. (2015). Digital transformation strategies. Business & Information Systems Engineering, 57(5), 339-343.
25. Nylén, D., & Holmström, J. (2015). Digital innovation strategy: A framework for diagnosing and improving digital product &service innovation.Business Horizons,57-67.
26. Peppard, J., & Ward, J. (2016). The strategic management of information systems: Building a digital strategy. John Wiley & Sons.
27. Renaud, A., Walsh, I., Kalika, M., 2016. Is SAM still alive? A bibliometric and interpretive mapping of the strategic alignment research field. J. Strat. IS 25 (2), 75–103.
28. Silva, J., Baiao, F. A., & Revoredo, K. (2016, May). Interactive Ontology Alignment using Alignment Antipatterns: A First Experiment. In (28). Brazilian Computer Society.
29. Strnadl, C. F. (2006). Aligning business and IT: The process-driven architecture model. Information systems management, 23(4), 67-77.
30. Tran, H. N., Coulette, B., & Thuy, D. T. B. (2007, July). Broadening the Use of Process Patterns for Modeling Processes. In SEKE (pp. 57-62).
31. Yeow, A., Soh, C., & Hansen, R. (2018). Aligning with new digital strategy: A dynamic capabilities approach. The Journal of Strategic Information Systems, 27(1), 43-58.
32. van Der Aalst, W. M., Ter Hofstede, A. H., Kiepuszewski, B., & Barros, A. P. (2003). Workflow patterns. Distributed and parallel databases, 14(1), 5-51.
33. Wiesche, M., Schermann, M., & Krcmar, H. (2013, January). When IT Risk Management Produces More Harm than Good: The Phenomenon of'Mock Bureaucracy'. In System Sciences (IEEE HICSS), 2013 46th (pp. 4502-4511). IEEE.
34. Winter, R., & Fischer, R. (2006, October). Essential layers, artifacts, and dependencies of enterprise architecture. EDOCW'06. 10th IEEE International (pp. 30-30). IEEE.
35. Wu, S. P. J., Straub, D. W., & Liang, T. P. (2015). How information technology governance mechanisms and strategic alignment influence organizational performance: Insights from a matched survey of business and IT managers. Mis Quarterly, 497-518.
36. https://www.computerworld.com/article/2486278/it-management/how-to-balance-maintenance-and-it-innovation.html
37. Jonathan, G. M., Rusu, L., & Perjons, E. (2020). Business-it alignment in the era of digital transformation: Quo vadis? Hawaii International Conference on System Sciences (HICSS), Hawaii, USA, (pp. 5563-5572).
38. Kahre, C., Hoffmann, D., & Ahlemann, F. (2017, January). Beyond business-IT alignment-digital business strategies as a paradigmatic shift: a review and research agenda. In Proceedings of the 50th Hawaii Intern. Conference on System Sciences.
39. Manfreda, A., & Štemberger, M. I. (2019). Establishing a partnership between top and IT managers. Information Technology & People.
40. Galliers, R. D. (2011). In celebration of diversity in information systems research. Journal of Information Technology, 26(4), 299-301.
41. El-Telbany, O., & Elragal, A. (2014). Business-information systems strategies: a focus on misalignment. Procedia Technology, 16, 250-262.
42. Budgen, D. (2003). Software design. Pearson Education.
43. Seddon, P. B., & Scheepers, R. (2012). Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. European journal of IS, 21(1), 6-21.
44. Wieringa, R., & Daneva, M. (2015). Six strategies for generalizing software engineering theories. Science of computer programming, 101, 136-152.
45. Yin, R. K. (2017). Case study research and applications: Design and methods. Sage.
46. Luftman, J., Papp, R., & Brier, T. (1999). Enablers and inhibitors of business-IT alignment. Communications of the Association for information Systems, 1(1), 11.